TECHNISCHE UNIVERSITÄT
CHEMNITZ

Superpixels and their Application for Visual Place Recognition in Changing Environments

von der Fakultät für Elektrotechnik und Informationstechnik
der Technischen Universität Chemnitz

genehmigte

Dissertation

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt von

Dipl.-Inf. Peer Neubert

geboren am 10.11.1984 in Karl-Marx-Stadt

Eingereicht am 18. August 2015

Gutachter: Prof. Dr.-Ing. Peter Protzel
Prof. Dr. Achim Lilienthal

Tag der Verleihung: 1. Dezember 2015

Peer Neubert
Superpixels and their Application for Visual Place Recognition in Changing Environments
Dissertation, Fakultät für Elektrotechnik und Informationstechnik
Technische Universität Chemnitz, Dezember 2015

Keywords
superpixel, visual place recognition in changing environments, image segmentation, segmentation benchmark, simple linear iterative clustering, seeded watershed, appearance change prediction, visual landmarks

Abstract

Superpixels are the results of an image oversegmentation. They are an established intermediate level image representation and used for various applications including object detection, 3d reconstruction and semantic segmentation. While there are various approaches to create such segmentations, there is a lack of knowledge about their properties. In particular, there are contradicting results published in the literature. This thesis identifies segmentation quality, stability, compactness and runtime to be important properties of superpixel segmentation algorithms. While for some of these properties there are established evaluation methodologies available, this is not the case for segmentation stability and compactness. Therefore, this thesis presents two novel metrics for their evaluation based on ground truth optical flow. These two metrics are used together with other novel and existing measures to create a standardized benchmark for superpixel algorithms. This benchmark is used for extensive comparison of available algorithms.

The evaluation results motivate two novel segmentation algorithms that better balance trade-offs of existing algorithms: The proposed Preemptive SLIC algorithm incorporates a local preemption criterion in the established SLIC algorithm and saves about 80 % of the runtime. The proposed Compact Watershed algorithm combines Seeded Watershed segmentation with compactness constraints to create regularly shaped, compact superpixels at the even higher speed of the plain watershed transformation.

Operating autonomous systems over the course of days, weeks or months, based on visual navigation, requires repeated recognition of places despite severe appearance changes as they are for example induced by illumination changes, day-night cycles, changing weather or seasons - a severe problem for existing methods. Therefore, the second part of this thesis presents two novel approaches that incorporate superpixel segmentations in place recognition in changing environments. The first novel approach is the learning of systematic appearance changes. Instead of matching images between, for example, summer and winter directly, an additional prediction step is proposed. Based on superpixel vocabularies, a predicted image is generated that shows, how the summer scene could look like in winter or vice versa. The presented results show that, if certain assumptions on the appearance changes and the available training data are met, existing holistic place recognition approaches can benefit from this additional prediction step.

Holistic approaches to place recognition are known to fail in presence of viewpoint changes. Therefore, this thesis presents a new place recognition system based on local landmarks and Star-Hough. Star-Hough is a novel approach to incorporate the spatial arrangement of local image features in the computation of image similarities. It is based on star graph models and Hough voting and particularly suited for local features with low spatial precision and high outlier rates as they are expected in the presence of appearance changes. The novel landmarks are a combination of local region detectors and descriptors based on convolutional neural networks. This thesis presents and evaluates several new approaches to incorporate superpixel segmentations in local region detection. While the proposed system can be used with different types of local regions, in particular the combination with regions obtained from the novel multiscale superpixel grid shows to perform superior to the state of the art methods - a promising basis for practical applications.

Contents

List of Acronyms and Terms **IX**

1 Introduction and Motivation **11**
1.1 Thesis outline, research questions and contributions 14
1.2 Publications and collaboration 16
1.3 Open source software releases 17
1.4 My personal journey to this thesis 17

I Superpixel Segmentation Algorithms and Benchmarks **19**

2 An Introduction to Superpixel Segmentations **21**
2.1 The problem of image segmentation 21
2.2 What are superpixels? 24
2.3 Creating superpixel segmentations 26
2.3.1 Fundamentals of image (over-) segmentation techniques 27
2.3.2 Algorithms 31
2.3.3 Summary of the presented superpixel segmentation algorithms . . 41

3 How to Compare Superpixel Segmentation Algorithms **45**
3.1 Superpixel segmentation benchmarks 46
3.2 Experimental setup 49
3.2.1 Figure-ground segmentation dataset 52
3.2.2 Using datasets with ground truth optical flow 52
3.2.3 Compared algorithms 56
3.3 Runtime 58
3.4 Segmentation quality 59
3.4.1 Boundary recall 60
3.4.2 Undersegmentation error 60
3.4.3 Maximum achievable segmentation accuracy 61
3.4.4 Results on segmentation quality 62
3.5 The stability of superpixel segmentations 64
3.5.1 Benchmarking robustness to affine transformations 64
3.5.2 Robustness to noise 66
3.5.3 A novel metric for stability based on ground truth optical flow . . 66
3.5.4 Results on segmentation stability 67

3.6 Compactness properties of superpixel segmentations 71
3.6.1 Compactness from geometric properties 72
3.6.2 A novel metric measuring the accordance with motion discontinuities 74
3.6.3 Results on segmentation compactness 75
3.7 Summary of the benchmark results 78

4 Two Novel Superpixel Algorithms: Preemptive SLIC and Compact Watershed 81
4.1 Trade-offs of superpixel segmentation algorithms 81
4.2 Preemptive SLIC: Making a good algorithm faster 84
4.3 Results of Preemptive SLIC 87
4.4 Compact Watershed: Getting from oversegmentations to superpixels. . 89
4.5 Results of Compact Watershed. 96
4.6 Conclusion. 100

II Superpixel for Place Recognition in Changing Environments 103

5 The Problem of Place Recognition in Changing Environments 105
5.1 Datasets comprising changing environments 108
5.1.1 The Nordland dataset - A 3000 km journey through all seasons. . 108
5.1.2 The Gardens Point dataset - Day and night with lateral shift . . . 110
5.1.3 The Alderley dataset - From sunny day to rainy night 112
5.2 Related work and own contributions to the field 112
5.3 Using precision-recall curves for evaluation of place recognition experiments 121

6 Superpixel-based Appearance Change Prediction 123
6.1 The concept of appearance change prediction for long term localization 123
6.1.1 Related work for appearance change prediction. 125
6.1.2 Overview of the main components 125
6.2 Superpixel descriptors. 126
6.2.1 A gentle introduction to descriptors 127
6.2.2 From keypoint descriptors to recognizing superpixels 129
6.2.3 What can a descriptor tell about superpixel overlap? 130
6.2.4 Experimental comparison 132
6.3 Superpixel vocabularies 135
6.3.1 A gentle introduction to visual vocabularies 135
6.3.2 Evaluating the semantics of superpixel words 137
6.4 Algorithmic approach SP-ACP: Learning to predict scene changes. . . . 140
6.4.1 Learning a vocabulary for summer and winter 141
6.4.2 Learning a dictionary to translate between vocabularies. 142
6.4.3 Predicting image appearances across seasons 142

6.5 Results on the Nordland dataset 143
6.5.1 Applying SP-ACP: Predicting images of the Nordland dataset . . 145
6.5.2 Experiments with FAB-MAP 145
6.5.3 Extending and improving BRIEF-Gist 145
6.5.4 Extending and improving SeqSLAM 148
6.6 Current limitations of the approach and future work 155

7 Landmark-based Place Recognition in Changing Environments 157
7.1 The components of place recognition based on local image features . . . 157
7.2 Local region descriptors based on Convolutional Neural Networks 159
7.2.1 What is a Convolutional Neural Network? 159
7.2.2 CNN for place recognition 160
7.2.3 Remarks on the computational efforts for CNN-based descriptors 161
7.3 The novel Star-Hough matching scheme: From local regions to image similarities 162
7.3.1 The basic problem and the challenges of changing environments . 163
7.3.2 The novel Star-Hough approach 164
7.4 Novel superpixel-based local region detectors 166
7.4.1 Using stable superpixel algorithms: Plain superpixels and segment soups 168
7.4.2 Using compact superpixel algorithms: The multi scale superpixel grid 169
7.4.3 Superpixel based weighting of scale space extrema 175
7.4.4 Object proposals 177
7.5 How to measure the repeatability of local region detectors 179
7.5.1 Size-normalized IoU 180
7.5.2 Size-sensitive IoU 180
7.5.3 Benefit of the novel metric - Evaluating the scale of SIFT-DoG . . 181
7.6 Experimental results on the repeatability of local region detectors 183
7.7 Experimental results on place recognition in changing environments . . 189
7.7.1 Influence of the proposed landmark matching scheme 190
7.7.2 Is the superpixel grid better than a fixed grid? 191
7.7.3 Is there a benefit from superpixel based reweighting of scale space extrema? 194
7.7.4 Comparison of local region detectors for place recognition in changing environments 196
7.8 Summary of achieved results and directions for future work 202

8 Conclusions 205
8.1 What has been achieved - Summary of the main contributions 205
8.2 What are open problems in terms of superpixel segmentations? 208
8.3 Achievements and open questions regarding place recognition 210
8.3.1 How do the two novel approaches compete? 210
8.3.2 Is the problem solved? 211

A Appendix **215**
A.1 Additional results Preemptive SLIC 215
A.2 Additional results Compact Watershed 218
A.3 Additional details on the Nordland dataset 220
A.4 Using Compact Watershed for SP-Grid 220
A.5 Using binary hash functions in combination with the proposed local landmarks 221

Bibliography **223**

List of Figures **241**

List of Tables **245**

List of Acronyms and Terms

ACP	Appearance change prediction, cf. chapter 6
ACR	Affine covariant region dataset, cf. [Mikolajczyk et al., 2005]
BOX	Regular grid like superpixels
BoW	Bag of words, also known as bag of visual words BovW
BRIEF-Gist	A holistic descriptor based on BRIEF - binary robust independent elementary features, cf. [Sünderhauf and Protzel, 2011]
BSDS	Berkeley segmentation dataset, cf. [Arbelaez et al., 2011]
CIE LAB	A colour space
CNN	Convolutional neural network, cf. section 7.2
CWS	Compact watershed, a novel superpixel algorithm, cf. section 4.4
DoG	Difference of Gaussians
EAMS	Edge augmented mean shift, a superpixel algorithm, cf. [Meer and Georgescu, 2001]
EdgeBoxes	An object proposal algorithm, cf. [Zitnick and Dollar, 2014]
ERS	Entropie rate segmentation, a superpixel algorithm, cf. [Liu et al., 2011]
F-score	A measure, the harmonic mean of precision and recall
FAB-MAP	A place recognition approach, cf. [Cummins and Newman, 2011]
FH	A superpixel algorithm, cf. [Felzenszwalb and Huttenlocher, 2004]
FN	False negatives in a classification experiment
FP	False positives in a classification experiment
GIST	A holistic image descriptor, cf. [Torralba et al., 2003]
GNSS	Global navigation satellite system
GPS	Global positioning system
Grid	A region detector, related to BOX segmentations, cf. section 7.7.2
HOG	Histogram of oriented gradients, cf. [Dalal and Triggs, 2005]
IoU	Intersection over union, a measure, also known as Jaccard index
KITTI	A dataset with ground truth optical flow, cf. [Geiger et al., 2012]
LMSE	Least mean square error, a robust estimation method
MASA	Maximum achievable segmentation accuracy, a measure, cf. section 3.4.3
MCG	Multiscale combinatorial grouping, an object proposal algorithm, cf. [Arbelaez et al., 2014]
MDE	Motion discontinuity error, a measure, cf. section 3.6.2
MSER	Maximum stable extremal regions, a region detector, cf. [Matas et al., 2002]
MUSE	Motion undersegmentation error, a measure, cf. section 3.5.3
NC	Normalized Cuts, a superpixel algorithm, cf. [Shi and Malik, 2000]
NC res	The same as NC, but with resized input images
oriSLIC	Simple linear iterative clustering, original implementation by Achanta et al. [2012]
PCA	Principal component analysis
pSLIC	Preemptive simple linear iterative clustering, a novel superpixel algorithm, cf. section 4.2
QS	Quickshift, a superpixel algorithm, cf. [Vedaldi and Soatto, 2008]
RANSAC	Random sample consensus, a robust estimation method
RatSLAM	A biologically inspired SLAM system, cf. [Milford et al., 2004]
RGB	A colour space
RP	Randomized Prim's, an object proposal algorithm, cf. [Manen et al., 2013]

SEEDS	Superpixels extracted via energy-driven sampling, a superpixel algorithm, cf. [Van den Bergh et al., 2012]
SeqSLAM	A topological SLAM system, cf. [Milford and Wyeth, 2012], important parameter is the length of the used sequences ds
SIFT	Scale invariant feature transform, a region detector and descriptor, cf. [Lowe, 2004]
SIFT-DoG	The DoG based scale space extrema detector of SIFT, SIFT-DoG-x is the region obtained from the inner Gaussian rescaled by factor x
SLIC	Simple linear iterative clustering, a superpixel algorithm, cf. Achanta et al. [2012]
SLAM	Simultaneous localization and mapping
SP-ACP	Superpixel based appearance change prediction, cf. chapter 6
SP-Grid	Superpixel based grid, a novel region detector, cf. section 7.4.2
Sp-Soup	Superpixel segment soup, a novel region detector, cf. section 7.4.1
Star-Hough	Star graph model and Hough voting based approach to compute image similarities, cf. section 7.3
SUN	Scene understanding database, a dataset, cf. [Xiao et al., 2010]
SURF	Speeded up robust features, a keypoint detector and descriptor, cf. [Bay et al., 2008]
TN	True negatives in a classification experiment
TP	True positives in a classification experiment
USE	Undersegmentation error, a measure, cf. section 3.4.2
VEK	A superpixel algorithm, cf. [Veksler et al., 2010]
VEK CI	The same as VEK but with a constant intensity constraint
VLFeat	A software libarary
vlSLIC	SLIC implementation from VLFeat
WIDQ	Weighted isodiametric quotient, a measure, cf. section 3.6.1
WIPQ	Weighted isoperimetric quotient, a measure, cf. section 3.6.1
WS	Seeded watershed, a superpixel algorithm, cf. [Meyer, 1992]
WTA	Winner takes all

Make things clear.
Simon Lacroix

1 Introduction and Motivation

Computer vision and mobile robotics are strongly related research fields. Vision is a powerful sensor for environmental perception as is evident from the biological examples. Mobile robotics tasks like navigation or mobile manipulation can be build upon visual information provided by 2d or 3d cameras. In the other direction, the field of computer vision may benefit by (at least) three means from the combination with mobile robotics: (1) From a so far rather theoretical point of view, placing cameras on mobile robots enhances computer vision systems from passive data processing tools to active perception systems that can explore and interact with the environment to raise interpretations and validate them actively. (2) The application of cameras on mobile intelligent agents poses a manifold of challenging research questions for the computer vision community. (3) The practical usage of computer vision systems on real robots, acting in real environments, brings computer vision systems from the laboratory to the field and reveals limitations of existing solutions. The research field of computer vision originates from artificial intelligence and mathematics back in the 1960s, two fields lesser-known for their practical orientation. Therefore, the point (3), application of computer vision systems on practical problems, can substantially enhance the field.

For instance, image segmentation is a computer vision task for which researchers have proposed theoretically profound solutions for decades and all stumble at widespread practical application. The objective of image segmentation is to find the outlines of the objects in an image. While this is a simple task for humans (at least for everyday images), it is surprisingly hard for artificial vision systems. A main reason is the plain bottom-up nature of classic figure-ground segmentation approaches: the outlines of the objects should be found without information *what are* the objects. The segmentation makes *hard* decisions that can not be reverted later.

Therefore, alternative approaches have been developed, for example, for the task of object detection. An approach to object detection, that showed to *not work* in real world situations, is to take an image, segment the foreground object and give it to a classifier. Instead, successful object detection systems skipped the segmentation step

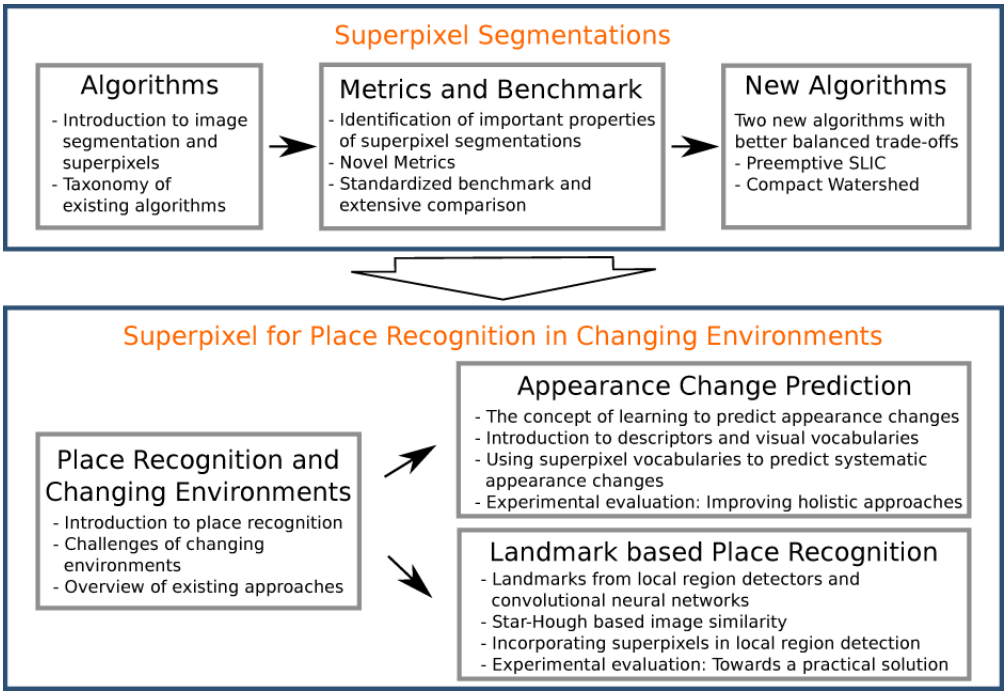


Fig. 1.1.: Relation of the parts of this work. The superpixel segmentations are the tool and place recognition in changing environments is the application. Each grey box corresponds to a chapter of this thesis.

and exhaustively searched for possible object locations in the image (the sliding window approach). More recently, this exhaustive search was in turn replaced by segmentations again - but this time it is not a single object segmentation, but a multi purpose segmentation based on an oversegmentation of the image into superpixels. Superpixels carry more information than single pixels, but are much smaller than objects. This kind of intermediate image representation showed to be beneficial for many other applications including semantic segmentation, 3d reconstruction from single views or stereo, tracking, video segmentation and semantic video annotation, traversability classification for ground robots, classification of urban streets, and many others.

While there are various algorithms available to create superpixel segmentations, there is a lack of insights on their properties and means for their evaluation - in particular, there are contradicting results published in the literature. The analysis, comparison and extension of superpixel segmentation algorithms is the topic of the first part of this thesis. The overall structure is illustrated in Fig. 1.1. Starting from the identification of important properties of these algorithms, novel metrics to measure them are proposed. They are combined with existing metrics in a standardized benchmark. This benchmark is used to compare a large set of available superpixel segmentation algorithms. The evaluation reveals important trade-offs of the properties of the available algorithms: For example, regarding the runtime and the segmentation quality or the control over geometric properties of the resulting segments. These trade-offs motivate the development of two novel superpixel segmentation algorithms that better balance the trade-offs.

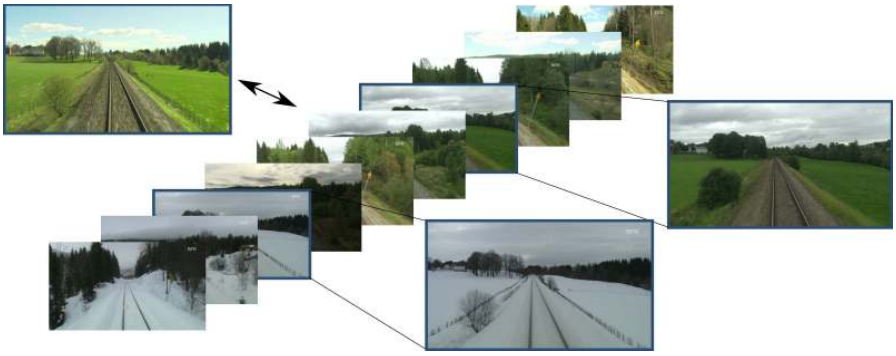


Fig. 1.2.: The problem of place recognition in changing environments. Given an image of the current appearance of a place (top left) and a set of images showing this place and others subject to severe appearance changes: Which images show the same place? (Nordland image credits: NRKbeta.no, cf. section 5.1.1.)

The topic of the second part of this thesis is the application of superpixel segmentations for a mobile robotics task: Visual place recognition in changing environments. This task is illustrated in Fig, 1.2.

Mobile robotics is an active and growing research field. For many subproblems, there are established solutions. For example, the past DARPA Grand Challenges, Urban Challenge and recent Robotics Challenge showed the progress in the field (but also revealed the need for more autonomy and higher level understanding of the world for creating and executing plans more robustly). Navigation is a fundamental capability for mobile robots. Researches have been working on this from the very beginning of this field. One of the currently most active research subjects is robust long term operation, including long term autonomy and long term navigation.

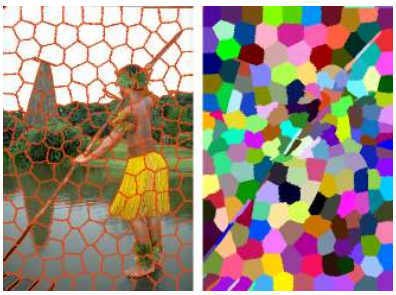
A major challenge for long term navigation are changing environments that occur when robots operate over hours, days, weeks and months. These changes can be caused by dynamic objects (e.g. moving people or cars) or by systematic changes in the environments, e.g. induced by day-night cycles, changing weather or seasons. The goal of visual place recognition algorithms is to match the current visual input with a set of images of known places. While there exists a broad range of algorithmic approaches to visual place recognition, in particular the systematic long term changes of the environment challenge all existing solutions. The second part of this thesis will investigate how superpixels can contribute to a solution and propose and evaluate two novel approaches: (1) Using visual vocabularies of superpixel words to predict systematic appearance changes. (2) Creating visual landmarks from superpixel and using them for place recognition.

The superpixel landmark based approach is integrated in a novel complete pipeline for place recognition, comprising novel approaches for local region detection, landmark description and computing image similarities from the landmarks. While the combination of these three steps (detection + description + matching) is quite common, for each of these steps, a novel approach is presented - each being particularly suited for the challenges induced by changing environments. The proposed system outperforms existing solutions and can be considered a promising solution for practical application.

1.1. Thesis outline, research questions and contributions

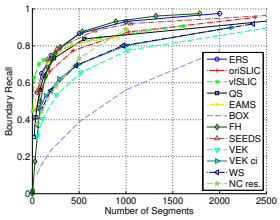
The contributions of this thesis are provided in the six chapters 2 to 7, each corresponding to a grey box in Fig. 1.1. The key research questions and contributions can be summarized as follows:

Chapter 2: Superpixel segmentations



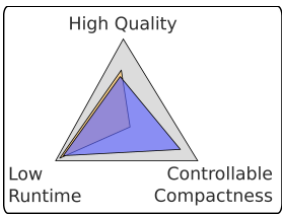
Research questions: Why is image segmentation an ill-posed problem? What are superpixel segmentations and how can they contribute to overcome this issue? Which superpixel segmentation approaches exist and how do they work?
Contributions: This chapter provides an introduction to superpixel segmentations and an overview of existing approaches.

Chapter 3: Novel metrics and benchmarks for superpixel algorithms



Research questions: What are important properties of superpixel segmentation algorithms? How can these properties be measured? In particular, what are criteria to measure stability and compactness of superpixel segmentations? What are the properties of the available algorithm implementations? Why are standardized benchmarks necessary?
Contributions: Motivated by contradicting results in the literature on the performance of superpixel algorithms, a standardized benchmark is provided. Quality, runtime, stability, and compactness are identified as important criteria for superpixels. Novel metrics to evaluate stability and compactness based on ground truth optical flow are proposed. These and other novel and existing metrics are combined in the new benchmark and used for extensive experimental evaluation of available algorithms.

Chapter 4: Novel, better balanced superpixel algorithms

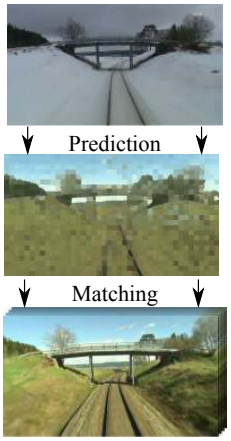


Research questions: What are trade-offs of the existing algorithms? Can these properties be better balanced? How can the runtime of the SLIC algorithm be reduced while maintaining its high quality segmentations? How can the irregularly shaped oversegmentations created by the very fast Seeded Watershed algorithm be turned into compact superpixel segmentations?
Contributions: Based on trade-offs that exhibited in the chapter 3, two novel algorithms are proposed: Preemptive SLIC incorporates a local preemption criterion in SLIC and creates high quality segmentations similar to SLIC and is about five times faster. Compact Watershed incorporates compactness constraints in a seeded watershed segmentation to create regular, compact superpixels while retaining the high speed of plain watershed segmentations.

Chapter 5: Place recognition in changing environments

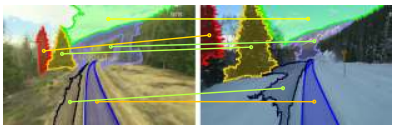
Research questions: What is place recognition in changing environments and what are the particular challenges? Which approaches exist and what are their limitations? Which benchmarks and datasets are currently available? How can superpixels contribute to a solution?
Contributions: Introduction of the problem of place recognition in changing environments and overview of related work.

Chapter 6: Appearance change prediction



Research questions: Is it possible to learn systematic image changes? How can superpixels be used to predict systematic appearance changes? Can existing place recognition algorithms benefit from such a prediction step? What are useful descriptors for superpixels? Do superpixel words provide semantic meaning?
Contributions: The novel idea of appearance change prediction and an implementation based on superpixel vocabularies. Superpixel descriptors and vocabularies are introduced and experimentally evaluated. The chosen representation is motivated by experimental results that quantify the amount of semantic information of superpixel words based on conditional entropy. Experimental results show that existing holistic place recognition approaches can benefit from this additional prediction step.

Chapter 7: Landmark based place recognition



Research questions: How can local landmarks be used to overcome the limitations of holistic approaches to place recognition? How sensitive are established descriptors and detectors towards appearance changes? Can descriptors based on convolutional neural networks be used for local features in changing environments? How can the spatial arrangement of landmarks with poor spatial precision and high outlier rates be incorporated in image matching? How can superpixels contribute to local region detection? Can the resulting novel approach improve the state of the art?
Contributions: This chapter presents the novel idea of using local region detectors together with descriptors obtained from convolutional neural networks as landmarks for place recognition in changing environments. To incorporate the landmarks' spatial arrangement despite poor spatial precision and high outlier rates due to the changing environments, the novel Star-Hough image matching approach is proposed. The superpixel segment soup, the multiscale superpixel grid, superpixel based importance reweigthing of scale space extrema and the usage of object proposals are novel superpixel-related approaches to local region detection for place recognition in changing environments. The proposed system, in particular in combination with the multiscale superpixel grid, performs superior to available methods.

In a nutshell, the four major contributions are the following:

- In terms of superpixel segmentations:
  1. Novel metrics and a standardized benchmark to evaluate superpixel segmentation algorithms.
  2. Two novel algorithms: Preemptive SLIC and Compact Watershed.
- In terms of place recognition in changing environments:
  3. The novel idea of appearance change prediction and an implementation based on superpixel vocabularies.
  4. A new local landmark based system combining novel approaches to landmark detection, landmark description and incorporation of the landmarks' spatial arrangement.

The improvements over the existing methods are extensively evaluated in the respective chapters. In particular the proposed local landmark based system of chapter 7 is supposed to be a promising contribution that improves the state of the art in place recognition based on superpixel segmentations. The final chapter 8 will summarize the achievements and discuss open questions and directions for future work.

A reading advice: The structure of the sections and subsections of this thesis was designed to emphasize the structure of the *content* not the structure of *my contributions*. Thus, the contributions are sometimes "hidden" in subsections. To simplify the identification of my contributions to the field and the achieved results, "Contribution X.Y" and "Result X.Y" statements are used throughout this thesis. If you ever get lost in the details, it may be helpful to skip to the next contribution or result statement to get a summary.

1.2. Publications and collaboration

Parts of this thesis have been published in a journal or conference proceedings. In particular these are the related publications:

Chapter 3:	[Neubert and Protzel, 2013], [Neubert and Protzel, 2012]
Chapter 4:	[Neubert and Protzel, 2014]
Chapter 5:	[Neubert et al., 2015b], [Sünderhauf et al., 2013]
Chapter 6:	[Neubert et al., 2013], [Neubert et al., 2015b], [Neubert and Protzel, 2015a], [Sünderhauf et al., 2013]
Chapter 7:	[Neubert and Protzel, 2015b]

The already published material is indicated in the text and by using footnotes. The formulation "section $X.Y$ including Fig. $X.Z$ is published in very similar form in" is used to indicate that these are citations of material from *my own* previously published papers. This citation might be slightly adapted to the overall form of this thesis (e.g. "this paper" is replaced with "this thesis"). Parts of the chapters 5 and 6 contain material that has been developed and published in collaboration with Niko Sünderhauf.

Whenever I describe work developed or published with others, I use the plural form ("we"), otherwise the first person singular form.

1.3. Open source software releases

The later presented results on existing comparisons in the areas of superpixel segmentations and place recognition will underline the importance of standardized benchmarks and the availability of ready to use implementations to allow for comparison. Implementations of the benchmarks and algorithms published in [Neubert and Protzel, 2012, 2013, 2014] were already provided together with the papers. They are available in form of open source Matlab toolboxes, free for non-commercial usage. The remaining superpixel metrics, modifications of the algorithms and the novel landmark based approach to place recognition in changing environments will be published together with this thesis.[1]

- The standardized superpixel benchmark is intended to enable researchers to compare novel superpixel algorithms to existing methods, and to enable users to select a suitable superpixel algorithm for the task at hand.
- The two novel superpixel algorithms extend the set of available, ready to use segmentation algorithms.
- The implementation of the proposed landmark based approach is intended as novel state-of-the-art baseline method for place recognition in changing environments.

1.4. My personal journey to this thesis

This section illustrates my *personal* journey to these thesis and can be skipped without ramifications for the understanding of the remainder of this thesis.

The work for this thesis has been spread over several years and included some turns and detours. In my diploma thesis, I worked on a biologically inspired approach to place recognition: the combination of saliency based proto-objects and a biologically inspired semi-metric SLAM system, RatSLAM. Proto-objects are image features inspired by bottom-up visual attention in the early human visual system. Basically they are obtained from centre-surround differences and biologically motivated normalization procedures. A main issue of these visual features is their lack of a reasonable spatial image support - thus, I thought they could benefit from a combination with image segmentations. A couple of months later, I found myself segmenting images and comparing segmentation algorithms - after a short affair with figure-ground segmentations I already found my way to the superpixel segmentations. I was very happy about being able to make some contributions to this field.

However, working in a group of mobile robotics enthusiasts, who treat computer vision mainly as a tool and not as a research field, I was gently guided back to the problem of mobile robot navigation. While this is a mature field, the particular problem of place

[1] `https://www.tu-chemnitz.de/etit/proaut/mitarbeiter/peerNeubert.html`

recognition in changing environments (e.g. summer-winter) showed to be an appealing challenge. In particular, since superpixels turned out to be a useful tool to implement a novel idea to approach this problem: learning to predict systematic appearance changes across seasons and other environmental changes. Together with Niko Sünderhauf, I was able to demonstrate promising improvements on place recognition performance of existing methods in combination with this novel prediction approach.

However, while the learning of systematic appearance changes for long term localization is a theoretically appealing idea, it poses strong requirements on the training data and can be considered a rather academic approach. At this point in time, the topic of my thesis was more or less fixed: superpixels and their application for place recognition in changing environments - I had a task (place recognition) and a tool (superpixels), any further contribution to my thesis should be related to both.

Based on the experiences in both fields, various ideas appeared, how the task might benefit from the tool and how the various challenges of changing environments could be addressed. Many of the approaches contributed only to my personal experience, not to a solution to the problem. However, the remaining ideas resulted in a novel system that builds upon superpixels, outperforms the available methods for place recognition in changing environments and can be considered a promising basis for a practical solution to this problem.

PART

I

SUPERPIXEL SEGMENTATION ALGORITHMS AND BENCHMARKS

The first part of this thesis comprises:

- An introduction to superpixel segmentations and existing approaches to generate them in chapter 2
- A standardized benchmark and experimental comparison based on novel and existing metrics in chapter 3
- Two novel superpixel algorithms in chapter 4

There's more to the picture than meets the eyes.

Neil Young, from the song "My My, Hey Hey", 1979

2 An Introduction to Superpixel Segmentations

2.1. The problem of image segmentation

Images are omnipresent in our lives: in form of pictures on the walls, images obtained from analogue or digital cameras or as the visual perception of the world through our eyes. To understand why image segmentation is hard for artificial vision systems, one has to understand that there is a fundamental difference between an image as the computer (or algorithm) sees it and the visual perception of humans. I want to demonstrate this by tasks that can easily be solved by one of them and are hard for the other.

The checkerboard illusion by Edward H. Adelson shown in Figure 2.1 is a well known and notwithstanding fascinating example how the human visual system can be fooled. For most humans it is hard to recognize (or even accept) that the two marked fields share the same intensity. It is a catchy demonstration for the overall functioning of the human visual system: We do not consciously construct the world from the particular grey level perceived at the retinal rod cells, instead our mind interprets the whole scene to reason about its individual components. Misleading this unconscious interpretation is the basis for many visual illusions. In contrast, artificial computer vision systems can solve some of these illusions easily. For example the checkerboard illusion vanishes for a typical bottom-up image processing chain that works on raw pixel values and evaluates illumination differences immediately.

On the other hand, telling the story of an everyday image, i.e. recognizing the contained objects and interpreting their relations and possibly their actions, is trivial for seven year old children but overcharges the most sophisticated computer vision pipelines available today. In general, bottom-up interpretation of images fails at the latest when it comes to image segmentation. Segmentation is the task of subdividing an image into

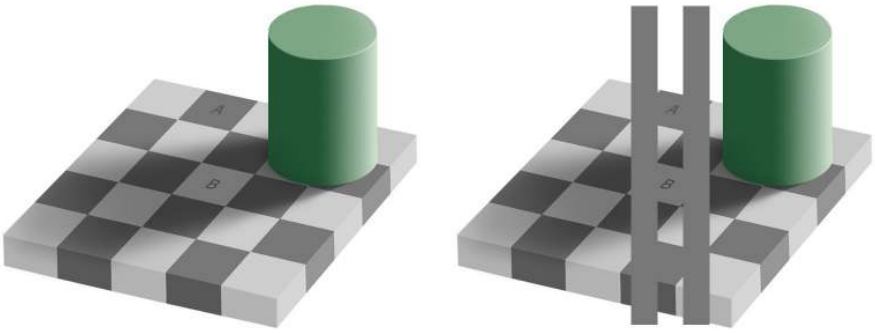


Fig. 2.1.: A simple task for a computer and hard for humans. The checkerboard illusion by Edward H. Adelson: Are the colours of the two marked squares in the left image the same? Despite their obviously different appearance, they share the same colour. The additional bars in the right image resolve the discrepancy. (Left image licensed under Creative Commons (CC-Zero) `http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html` [Last accessed: 2015-08-14])

its constituent regions or objects [Gonzalez and Woods, 2002, p. 567]. It is a handy example for a task that is surprisingly hard for computer vision systems: "Segmentation of nontrivial images is one of the most difficult tasks in image processing" [Gonzalez and Woods, 2002, p. 567]. This becomes clearer if we think of increasingly complex images as they are shown in Figure 2.2. Humans can recognize the individual grains, the flippers at the feet of the diver (although one is partially occluded) and even the objects on the table. Moreover, we can provide a reasonable contour for each - or in other words, segment the object from the background. However, the way we humans see the world is not to obtain raw sensory input from our eyes and then consciously interpret them. It's rather that a magical, ready to use interpretation of the raw information becomes aware. The interpretation is strongly influenced by conscious and unconscious, high and low level mechanisms like semantic knowledge or saccades (unconscious eye movements). However, in bottom-up image segmentation, segmentation and recognition are artificially separated and an ordering is induced between both. The bottom-up segmentation system has to decide about the boundary of an object before knowing the object.

In its general form, image segmentation is an ill-posed problem. "Defining a criterion for grouping pixels clearly depends on the goal of the segmentation. Consequently, a unique general method cannot perform adequately for all applications." [Stolkin, 2007, p. 431]. An intuitive example is the so called figure-ground segmentation, which is defined as separating the foreground object(s) from the background. Given the cluttered table in Figure 2.2 it becomes clear, that we need higher level information to decide whether the foreground is the table *with* all the stuff on top, or the stuff itself or even individual parts of the stuff.

A major problem of bottom-up figure-ground segmentation algorithms in computer vision pipelines are *early hard* decisions that can not be reverted later. E.g. an approach to object detection that will *not work* is to take an image, segment the foreground object and give it to a classifier. Thus, the dominant approach to object detection in images

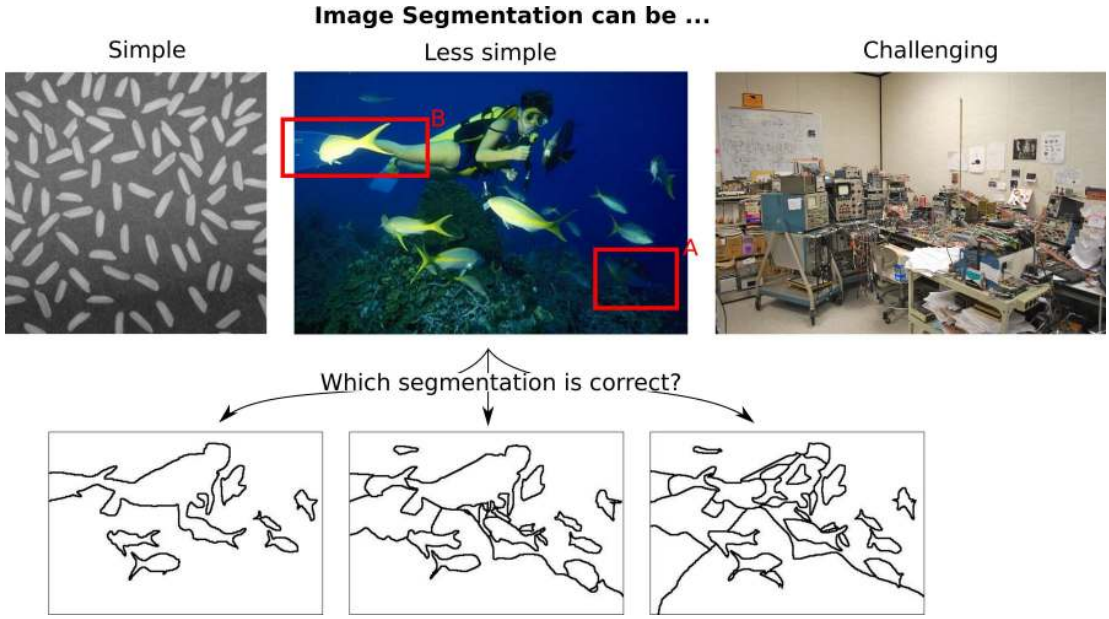


Fig. 2.2.: Beyond the Matlab rice grain counting example (top-left), segmentation can be challenging. For example due to object boundaries that are not induced by visual cues (e.g. in the red box A) and objects that are split into parts (B). It is an ill-posed problem. None of the segmentation boundary images in the bottom row can be considered the *only* true solution. Also, none of them can be considered a true solution *altogether* since none of them captures all object boundaries. Scenes like that shown in the right image underline the challenges for real world usage of image segmentation algorithms.

has been the sliding window approach for many years. Instead of classifying the single one foreground segment, the image is exhaustively searched by moving a classification window (the part of the image that is given to the classifier) step by step over the entire image. The sliding window approach to object detection has recently been superseded by a new class of detection algorithms based on object proposals - which can in turn be based on segmentations again. But this time, it is not a single foreground-background segmentation but a so called multi purpose image segmentation, an oversegmentation of the image that is given to higher level processing steps. This kind of segmentation is also known as superpixel segmentation and has proven valuable for a broad range of computer vision applications as will be discussed in the following section 2.2.

Before we proceed, I want to give a more formal definition of the basic entities involved in image segmentation: In digital image processing, an image is the result of discrete sampling of reflected light. More formally:

Definition 2.1 (Image) *An image I of height h and width w is a vector function $I\colon W \times H \to D^c$ over the discrete set of pixels $P_{i,j}$ with $i \in H = \{1, ..., h\}$, $j \in W = \{1, ..., w\}$, number of channels c and codomain D for each channel.*

Typical choices for D are real numbers between 0 and 1: $D = \{d \in \mathbb{R} : 0 \leq d \leq 1\}$ or the range of 8 bit integer values $D = \{d \in \mathbb{N}_0 : 0 \leq d \leq 255\}$. While there may be different codomains amongst channels they are often converted to a common codomain for the computational implementation. A typical configuration are three channels, one for each of the colours red, green and blue - yielding an RGB colour image.

The result of an image segmentation is a partition of the image into disjoint sets of pixels:

Definition 2.2 (Image Segmentation) *Given an image I with the set P of pixels, a set of disjoint subsets $S = \{S_1, ..., S_m\}$ of P such that $P = \cup_{r=1}^{m} S_r$ is called an image segmentation.*

This definition comprises all possible image segmentations. Additional criteria like connectedness and homogeneous appearance should be incorporated to yield useful segments for the application at hand. Such criteria are later introduced together with image segmentation techniques.

An image segmentation can e.g. be expressed by a mapping $L(i, j)$ that assigns the label of the assigned segment to each pixel $P(i, j)$. In a foreground-background segmentation, there are exactly two labels. Multi-object segmentation of k objects (including a background object with label 0) yields $L(i, j) \in \{0, 1, ..., k\}$. The following section presents and discusses segmentations with $max_{i,j}(L) >> k$ for k objects in the image. Such segmentations are called superpixel segmentations.

2.2. What are superpixels?

In one sentence, superpixels are an oversegmentation of an image - or seen the other way around a perceptual grouping of pixels. Instead of finding the few (e.g one to five) foreground segments that correspond to objects, superpixel segmentation algorithms split the image into typically 25 to 2500 segments. The objective of this oversegmentation is a partitioning of the image such that no superpixel is split by an object boundary, while objects may be divided into multiple superpixels. This way, the object outlines can be recovered from the superpixel boundaries at later processing stages. Such segmentations are sometimes also coined *multi purpose image segmentations*. The difference between a "classical" object-background segmentation and a superpixel segmentation becomes obvious by the example segmentations in Fig. 2.3. Formally, a superpixel segmentation can be defined as follows:

Definition 2.3 (Superpixel Segmentation) *A superpixel segmentation is a segmentation according to definition 2.2 with a total number of segments that is much larger than the number of objects in the image. Given an image of size (w, h) with $n = w \cdot h$ pixels that contains k objects, then for the number s of superpixels typically yields: $n \gg s \gg k$.*

Fig. 2.3.: Example results of different oversegmentation algorithms on the same input (from left to right: NC [Shi and Malik, 2000], EAMS [Comaniciu and Meer, 2002], FH [Felzenszwalb and Huttenlocher, 2004]). Each segmentation is visualized by the boundary image drawn in red colour on the input image and by the label image, where all pixels that belong to the same segment share the same colour.

For example, the image in Fig. 2.3 has $481 \cdot 321 = 154,401$ pixels and maybe about 10-20 objects. It has been segmented into about 200 superpixels from which the object boundaries could be recovered. Superpixels are an intermediate image representation carrying more information than individual pixels while avoiding premature hard decisions about object boundaries. These decisions are left for higher computational layers.

Before its current meaning, the term superpixel was used for slightly different concepts. Initially it has been used for any group of neighboured pixels, particularly including figure-ground segmentations. In the 90s, superpixels were commonly understood as the 3x3 or 5x5 neighbourhoods of a pixel. Today, the term is commonly used for any type of oversegmentation of an image.

The example superpixel segmentations obtained by different algorithms in Fig. 2.3. shows that the results can be quite different. In particular, there are algorithms that produce *compact* segments which are regularly distributed and of similar size (e.g. the left segmentation in Fig. 2.3). For other algorithms, the individual segments can vary strongly in size and shape (middle and right segmentations in Fig. 2.3). In the literature, the term superpixel is sometimes only used for the first type of regular segmentations, while the latter irregular segmentations are coined oversegmentations. This is due to the more pixel-like appearance and distribution of the compact segmentations. In this thesis, the terms superpixel segmentation and oversegmentation will be used interchangeable for both classes of multi purpose segmentations. However, there will be an extensive evaluation of compactness properties of superpixels to gain a more detailed understanding of regularity properties of oversegmentations in section 3.6.

The first superpixel algorithm in its current understanding was the *Normalized Cuts* algorithm presented by Shi and Malik [1997]. The authors extended their approach in subsequent work and established the term superpixel for this kind of segmentation in [Ren and Malik, 2003]. A more detailed description of this algorithm can be found in section 2.3.2.

Which problems can benefit from superpixel segmentations? They are an intermediate image representation. As the term *multi purpose image segmentations* implies, they are

used for various applications. Malisiewicz and Efros [2007] analysed the benefit of such segmentations in the context of object recognition. They draw two important conclusions: 1. Object recognition benefits from the right spatial support. 2. It is feasible to provide this spatial support based on oversegmentations. This work led the way for the inclusion of oversegmentation based object proposal algorithms in some of the currently best performing object detection pipelines. For example the winning and second place object detection entries of the ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC13) [Russakovsky et al., 2015] both build on object proposals [van de Sande et al., 2014; Wang et al., 2013], as well as the 2014 wining entry Region-based Convolutional Network (R-CNN) [Girshick et al., 2014]. The key idea of object proposal algorithms is to replace the exhaustive search performed by sliding window object detection approaches. Instead of giving all possible bounding boxes of relevant location, size and aspect ratio to the classifier, object proposal algorithms generate a set of promising candidates that are given to the classifier. Many state of the art object proposal algorithms are build upon oversegmentations, e.g. Selective Search [Uijlings et al., 2013] and Randomized Prim's [Manen et al., 2013] build upon Felzenszwalb-Huttenlocher segmentation [Felzenszwalb and Huttenlocher, 2004] while Category-Independent Object Proposals [Endres and Hoiem, 2014] and Multiscale Combinatorial Grouping [Arbelaez et al., 2014] use superpixels based on gPb [Arbelaez et al., 2011]. This nicely illustrates how superpixel can be used to overcome the problems induced by the ill-posed problem formulation of segmenting objects from images.

Other applications of superpixels include semantic segmentation [Miĉuŝik and Koŝecká, 2009], 3d reconstruction from single views [Hoiem et al., 2005] or stereo [Miĉuŝik and Koŝecká, 2010], tracking [Wang et al., 2011], video segmentation [Vazquez-Reina et al., 2010] and semantic video annotation [Kae et al., 2014], traversability classification for ground robots [Kim et al., 2007], classification of urban streets [Upcroft et al., 2014], and many more. Similar to superpixels, there exists a group of algorithms called supervoxels. These are extensions of superpixels from the two dimensional spatial image domain to the three dimensional spatial-temporal video domain, e.g. [Veksler et al., 2010], or the three dimensional euclidean space, e.g. [Papon et al., 2013]. Although the range of ready to use supervoxel algorithms is rather limited, their application should be considered whenever superpixels are applied to such three dimensional data.

As a teaser: The second part of this thesis will deal with a particular application of superpixel segmentation in the context of mobile robotics: How can superpixel segmentations be used for place recognition in changing environments. However, this first part will provide us the superpixel-related tools to approach this question.

2.3. Creating superpixel segmentations

There are almost as many different superpixel segmentation algorithms as there are possible applications. Depending on the intended usage, a different superpixel algorithm may be the right choice. Figure 2.3 already gave an idea of the broad spectrum of possible outcomes that such algorithms can provide. To find the right algorithm for the

task at hand, knowledge about available algorithms and their theoretical and practical properties is crucial. The rest of this first part of the thesis is dedicated to the presentation of existing and new superpixel algorithms, the identification of crucial properties and an extensive evaluation of these properties using existing and newly proposed evaluation metrics. This section will first introduce some basic techniques and definitions in section 2.3.1 to pave the way for the understanding of the selected superpixel algorithms presented in section 2.3.2.

2.3.1. Fundamentals of image (over-) segmentation techniques

Finding the components or parts of an image is a fundamental problem in image processing and computer vision. Image segmentation has a long history, tracing back to the fields of artificial intelligence (e.g. image regions for scene analysis by Brice and Fennema [1970]) and mathematics (e.g. segmentation through functional approximation by Pavlidis [1972]). According to its long history, a variety of approaches to image segmentation exits. Many of them were applied to both figure-ground- and oversegmentations.

In general, they are based on two region properties: similarity and discontinuity [Gonzalez and Woods, 2002, pg. 568]. A coherent image region is induced by either its inner similarity or the discontinuities at the borders to other regions (or a combination of both). There are various image cues that can be exploited to compute similarity and discontinuity. Typical choices are grey level intensity, colour, texture cues and the spatial image location. In particular, for colour and texture cues there exist a wide range of possible alternatives.

For example texture cues may be obtained from statistical, geometrical, model based or signal processing methods [Wu, 2003]. Statistical methods derive a set of statistics from the grey level distribution of local neighbourhoods, e.g in form of a grey level cooccurance matrix (also called GLCM or Haralick features) [Haralick et al., 1973]. Geometrical methods try to find basic primitives or parts, that constitute the texture, e.g. [Rosenfeld and Lipkin, 1970]. Model based methods aim at capturing the process that generated the texture, e.g. in form of Markov Random Fields [Dubes and Jain, 1989]. Signal processing methods analyse texture using spatial filters or through filtering in the frequency domain, e.g. Law's Filter [Laws, 1980] or Gabor filters [Bovik et al., 1990]. Increasingly complex methods yielded progress for texture analysis: "[...] considerable progress was made in both classification and synthesis by modelling textures using first the mean, then the mean and variance and finally by the full joint PDF of locally computed filter responses" [Varma and Zisserman, 2003]. Although simpler, e.g. texton-based approaches, also showed competitive performance [Varma and Zisserman, 2003], computation of texture cues is a fairly time consuming task. Superpixel segmentation algorithms that are used as a preprocessing step often relinquish the effort of incorporating texture.

For colour features, there is a similar diversity. Beyond the often used RGB colour space there exists a wide range of alternative colour spaces with different properties. The RGB colour space is a result of the design of the imaging sensors used in most cameras. They provide three types of sensor cells, one for red, blue and green. This is close to the biological example in our eyes, where we have S,M and L type cells for perception

of light with different wavelength. Nevertheless, it is hard for humans to find the RGB-tuple that corresponds to the colour of a shown patch by trial-and-error. One reason is that for example the intensity (or brightness/luminance) of a colour is influenced by all three channels. There are colour spaces that better decouple independently perceived appearance properties of the colour. For example, the HSV colour space is composed by hue (H), the dominant wavelength, saturation (S), a representation of the purity of the colour and the value (V), the intensity or brightness. The human visual system performs a slightly different decoupling: Very early in the visual path, the excitation of the above mentioned S,M and L cells is encoded in a channel for the intensity, and two others for the red-green and the blue-yellow contrast. The CIE LAB colour space is a technical implementation of this process. It provides the interesting property that the Euclidean distance in the LAB colour representation is close to the difference perceived by humans (at least for fairly similar colours). This makes the CIE LAB colour space particularly interesting for segmentation algorithms that try to resemble human figure ground segmentations. There exist many colour features beyond these, e.g. see [Gevers and Smeulders, 1996] for an overview.

However, although the available implementations of segmentation algorithms use a particular set of image cues, for most of them, this choice can be separated from the algorithmic approach and the approach can be used with other image cues. An example will be presented in section 4.2, where the influence of using the RGB or LAB colour space for a particular approach is evaluated. While the selection of the right set of features is important, the focus of this thesis is on the segmentation approaches and not on the image cues used.

There are some general concepts and algorithms that are used in several of the described algorithms in this section and over the course of this thesis. I want to introduce these concepts in the following. Some of the terms have varying meanings in the literature and lack a common exact definition, e.g. the term *image graph*. The here presented definitions are consistent through this thesis, but may be different from those in the referenced literature.

2.3.1.1. Image segmentation and graph theory

Image segmentation can be formulated as a graph partitioning problem. Several state of the art segmentation and oversegmentation algorithms are based on this analogy. A graph can be defined as follows [Diestel, 2000]:

Definition 2.4 (Graph) *In graph theory a graph $G = (V, E)$ is the combination of a finite set of vertices V and a set of edges $E \subseteq [V]^2$ that connect vertices. Each edge is a 2-element subset of V, the edge connecting vertices i, j is denoted $e_{\{i,j\}}$. In case of a weighted graph $G = (V, E, W)$, a weight $w_{\{i,j\}} \in W \subset \mathbb{R}$ is associated to each edge.*

To relate graph theory to image processing, the image graph is a special graph that associates vertices to pixels and weighted edges to relations between pixels.

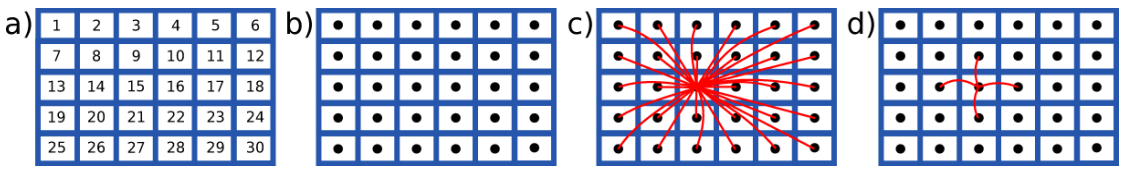
Definition 2.5 (Image Graph) *An image graph is a weighted graph $G = (V, E, W)$ associated to an image I where each node $v_i \in V$ represents an image pixel and each edge $e_{\{i,j\}} \in E$ connects a pair of pixels i, j. The edge weights $w_{\{i,j\}} \in W$ are associated to similarities (or dissimilarities) between the pixels corresponding to this edge.*

The edge weights can be obtained from simple intensity or colour differences, or involve higher order measures like texture. In case of oriented measures, like image gradients, the edges may also be directed and constitute a directed graph. An often used form of an image graph is a grid graph.

Definition 2.6 (Grid Graph) *A grid graph is an image graph with an edge $e_{i,j} \in E$ for each pair of neighboured pixels i, j. Typical neighbourhoods are the Von Neumann neighbourhood (4-connected pixels) or the Moore neighbourhood (8-connected pixels).*

This significantly reduces the number of edges from $|V|^2$ for a fully connected image graph to about $4 \cdot |V|$ or $8 \cdot |V|$ depending on the chosen neighbourhood. The difference can be seen in the following example.

Example 2.1 (Image Graphs) *The 30 pixels of the tiny (5×6) image, illustrated by blue boxes in a), can be represented by a set of vertices in b). Connecting each pixel to all other pixels by an edge yields a fully connected image graph. The edge set for an example pixel is shown in c). The grid graph reduces the set of edges to the spatial neighbourhood of a pixel. The resulting set of edges for a 4-connected neighbourhood and the same example pixel can be seen in d).*



For an image that is associated to an image graph, each graph partition yields a segmentation of the image.

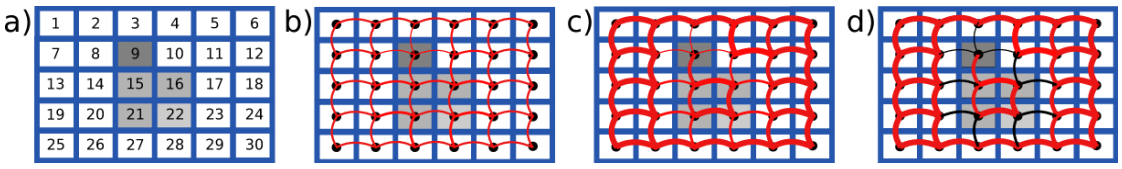
Definition 2.7 (Graph Partition) *Given a graph $G = (V, E)$, a set of disjoint subsets $P = \{P_1, ..., P_k\}$ of the set of vertices V such that $V = \cup_{i=1}^{k} P_i$ is called a graph partition.*

This definition is very similar to the previous definition of image segmentation but replaces pixels with vertices. This facilitates the application of a broad range of graph theoretical approaches to compute graph partitions, most importantly those based on graph cuts. Conceptually, a graph cut is the set of edges that connects two parts of a graph.

Definition 2.8 (Graph Cut) *Given a weighted graph $G = (V, E, W)$ and a bipartite graph partition $P = \{P_1, P_2\}$, the set of edges $C = \{c_{\{i,j\}} \in E : ((v_i \in P_1) \wedge (v_j \in P_2)) \vee ((v_i \in P_2) \wedge (v_j \in P_1))\}$ is a graph cut. The weight of the graph cut is the sum of the weights of its edges.*

Given an image graph with edge weights derived from pixel similarities, a partition with a small weight graph cut splits the set of vertices and correspondingly pixels such that there are few similarities between connected pixels located in different subsets of the partition.

Example 2.2 (Image Segmentation and Graph Cuts) *The tiny image in a) can be seen as a darker foreground object (pixels 9, 15, 16, 21 and 22) in front of a white background. b) shows the edges of the corresponding grid graph with a 4-connected neighbourhood. The weight of each edge can be computed from grey level similarity. High edge weights correspond to high pixel similarity and are illustrated by thick edges in c). The set of thin black edges in d) constitute a small weight cut yielding a partition of the image graph that separates the foreground from the background.*



There are well investigated algorithms to compute a partition with (approximately) minimal weight graph cut, e.g. based on the MinCut-MaxFlow theorem [Ford and Fulkerson, 56]. Recursive computation of such cuts can be used to iteratively refine the partition. The outlined connections between graph theory and image segmentations are exploited in several segmentation approaches [Shi and Malik, 2000; Felzenszwalb and Huttenlocher, 2004; Veksler et al., 2010]. One of the challenges when computing image segmentations based on graph cuts is to avoid trivial image graph partitions that separate single pixels or very small groups. How this problem can be solved is further discussed when the Normalized Cuts [Shi and Malik, 2000] algorithm is presented in section 2.3.2.1.

2.3.1.2. Image segmentation and data clustering

Grouping pixels to image segments can also be approached using techniques for unsupervised data clustering. A well known algorithm for this task is k-means. There are several oversegmentation algorithms based on k-means (e.g. Achanta et al. [2012]) or related algorithms like mean-shift (e.g. Comaniciu and Meer [2002]) and medoid-shift (e.g. Vedaldi and Soatto [2008]). To prepare the presentation of these algorithms, I want to shortly review the k-means algorithm.

k-means is a vector quantization method that simultaneously finds k clusters in a set of n observations and assigns the observations to the clusters. Finding the optimal solution of this problem is NP-hard. The here presented approximation procedure published by Lloyd [1982] goes back to an idea of Steinhaus [1956]. It uses an iterative algorithm to find a local minimum of the following objective function: Given a set of n observations $\{x_1, x_2, ..., x_n\}$ and the number of clusters k, and being $P = \{P_1, P_2, ..., P_k\}$ a partition of the set of observations, find:

$$\underset{P}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in P_i} ||x - \mu_i||^2$$

where μ_i is the mean of all observations assigned to cluster P_i. In simple words, k-means searches a clustering that minimizes the within-cluster sum of squared distances. Starting from a random initialization of the k cluster centres with samples from the observations, two steps are repeatedly executed:

1. **Assignment** Assign each observation to the nearest cluster centre.

$$P_i^{(t)} = \{x_s : ||x_s - m_i^{(t)}||^2 \leq ||x_s - m_j^{(t)}||^2 \forall j, 1 \leq j \leq k\}$$

2. **Update** Compute the new cluster centres from the assigned observations.

$$m_i^{(t)} = \frac{1}{P_i^{(t)}} \sum_{x_j \in P_i^{(t)}} x_j$$

A main drawback of k-means is the necessity to choose the number of clusters k. Moreover the results strongly depend on the choice of the initial clusters. For practical applications, often multiple runs with different initializations and possibly different numbers of clusters are performed. The solution with the best value of the above objective function is taken.

Section 2.3.2.7 presents the simple and elegant method by [Achanta et al., 2012] to create superpixels based on k-means. In their approach, the parameter k can be used to influence the number of created superpixels. However, the methods by Comaniciu and Meer [2002] and Vedaldi and Soatto [2008] presented in sections 2.3.2.3 and 2.3.2.4 build on similar methods that can implicitly estimate the number of clusters. Moreover, section 4.2 will present a novel segmentation method that is also based on k-means and creates high quality segmentations at low runtime.

2.3.2. Algorithms

The following pages present a selection of superpixel segmentation algorithms from the literature. The underlying algorithmic approach of each algorithm is outlined and accompanied by example segmentations that illustrate the segmentation characteristics. A comparison of the technical properties of these algorithms is also given in table 2.1,

followed by more segmentation examples in Fig. 2.13. Prerequisite for an algorithm to be included in this comparison is a publicly available implementation. Further, this comparison focuses on algorithms that provide sufficiently low runtime to be used as a preprocessing step. The only exception is the Normalized Cuts algorithms that is included due to its historical relevance.

Beyond the overview of their basic properties in table 2.1, these algorithms are extensively evaluated regarding runtime in section 3.3, segmentation quality in section 3.4, their stability in the presence of several types of distractions in section 3.5 and their compactness properties in section 3.6. Readers that already know these algorithms or are just interested in the comparison can directly skip to one of the aforementioned sections.

2.3.2.1. Normalized Cuts (NC) - The very first superpixel algorithm

Properties NC	
Technique	Graph cuts
Cue(s)	Intensity, colour, texture, size
Compact?	Yes
Runtime	Minutes to hours

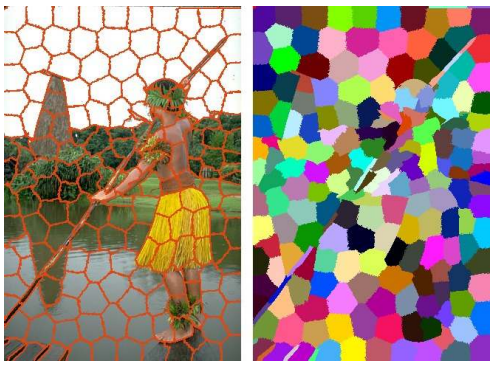


Fig. 2.4.: Example NC segmentation

Shi and Malik [1997] proposed a novel approach for solving the perceptual grouping problem in computer vision: the normalized cuts algorithm. A more detailed description can also be found in Shi and Malik [2000]. Rather than focusing on local image features, they incorporate a global optimization criterion. They propose to solve a general eigenvalue problem on the image graph to optimize the total dissimilarity between image segments as well as the total similarity within each segment. In subsequent years Malik et al. [2001] extended the algorithm with the intervening contour framework, texton based texture cues and proposed to use NC to produce an oversegmentation of the image as a preprocessing step for the final segmentation. Ren and Malik [2003] investigated the use of Gestalt principles for combining the result of the NC oversegmentation to large segments and finally established the term superpixel for such oversegmentations. This can be regarded as the dawn of the era of superpixel segmentation algorithms. The difference between the '97 and the later approaches is the number of produced segments: in 1997 the Normalized Cuts algorithm was used to find the few (e.g. 4-7) major segments in the image, while later on the algorithm was used as a preprocessing step to split the image into much more superpixels that could subsequently be grouped to larger segments based on higher level cues. In [Malik et al., 2001] the number of superpixels was 10-100 and further increased to 200 in [Ren and Malik, 2003].

For the Normalized Cuts algorithm, the image is represented as a weighted undirected graph $G = (V, E)$. In terms of graph theory an image segmentation can be seen as a graph partitioning (see Definition 2.7). The weights of all edges that connect vertices that belong to two different sets sum up to the cut of these two sets. Thus, the edges that belong to the graph cut between two parts of an image graph form the boundary between the associated image segments. There exist efficient algorithms to find minimal cuts in image graphs (e.g. based on the MinFlow - MaxCut theorem). Wu and Leahy [1993] noticed in an earlier approach of graph cut based image segmentation that the minimum cut criteria favours cutting small segments. This is not surprising since larger segments contain more edges in their cut and thus have higher cut values. To avoid this unnatural bias, the Normalized Cut computes the cost of a partition of V into subsets A and B as a fraction of the total edge connections to all the nodes in the graph:

$$NC(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \tag{2.1}$$

where

$$assoc(A, V) = \sum_{a \in A, v \in V} w(a, v) \tag{2.2}$$

is the sum of weights from the subset of nodes A to all nodes in the graph. This definition penalizes small sets of vertices since their cut value "almost certainly" (see Shi and Malik [1997]) becomes a high fraction of their total sum of connection weights. One key advantage of this approach is that a good approximation to the optimal partition can be computed efficiently by solving a generalized eigenvalue system.[1] Finding the true optimal partition is an NP-hard problem. For details, please refer to Shi and Malik [2000].

For computation of superpixels based on Normalized Cuts Ren and Malik [2003] use a locally connected graph over the set of image pixels with weights based on the contour and texture cues from Malik et al. [2001]. Locally connected means, there are edges to all nodes within a certain distance. To produce higher numbers of segments using NC, they propose to compute the second to the twelfth smallest eigenvalues, weight the eigenvectors according to their eigenvalues, transform each pixel in this eigenvector space and finally use k-means vector quantization to find clusters in this transformed space. These clusters are the resulting superpixels. An open source implementation of the original Normalized Cuts algorithm for Matlab and C++ is available[2]. Major

[1] The general idea is the following: Given the weight matrix W composed by all weights $w_{i,j}$ between nodes i and j, and D the diagonal matrix containing the sum of all weights of each node: $D_{i,i} = \sum_j w_{i,j}$. Shi and Malik [1997] showed that the Normalized Cut problem can be solved by computing:

$$y = \arg\min NC = \arg\min_y \frac{y^T(D - W)y}{y^T D y} \tag{2.3}$$

where $y = \{a, b\}^N$ is a binary indicator specifying for each pixel to which partition it belongs. Since the above expression is a Rayleigh quotient, one can solve this expression by relaxing y to take on real values (instead of discrete labels) and solving a generalized eigenvalue problem.

[2] `http://www.timotheecour.com/software/ncut/ncut.html` [Last accessed: 2015-08-14]

drawbacks are the high runtime of several minutes for larger images and the fact that the errors introduced by the approximation are not well understood [Felzenszwalb and Huttenlocher, 2004].

2.3.2.2. Felzenszwalb-Huttenlocher segmentation (FH)

Properties FH	
Technique	Graph merging
Cue(s)	Colour
Compact?	No
Runtime	About one hundred ms

Fig. 2.5.: Example FH segmentation

Felzenszwalb and Huttenlocher [1998] propose another graph-based segmentation approach that later became a well-known superpixel algorithm, more details can be found in [Felzenszwalb and Huttenlocher, 2004]. They define a predicate for measuring the evidence of a boundary between two regions and present an implementation in a greedy algorithm that also satisfies global properties. Its goal is to preserve details in low-variability image regions and ignore details in high-variability image regions. Starting from a weighted 8-connected image graph with pixel dissimilarity values in the edge weights (i.e. Euclidean distances in RGB colour space) two measures are defined:

1. The internal difference $ID(C)$ of a component C is the largest weight in the minimum spanning tree (MST) of the component. The MST is the cheapest (in terms of the edge weights) structure to connect the whole component. Its largest weight in the MST appears at the part of the component that is the most loosely coupled to the component.
2. The external difference $ED(C_1, C_2)$ between two components C_1 and C_2 is the minimum weight of an edge connecting the two components (using median boundary weight may be more robust but makes the problem NP-hard [Felzenszwalb and Huttenlocher, 2004]).

To decide whether there is evidence for a boundary between two components the external difference is compared to the internal differences.

$$D(C_1, C_2) = \begin{cases} true & \text{if } ED(C_1, C_2) > \min(ID(C_1) + \tau(C_1), ID(C_2) + \tau(C_2)) \\ false & \text{otherwise} \end{cases} \tag{2.4}$$

The external difference between the components has to be larger than the internal dif-

ference of each component C plus an additional term $\tau(C)$.

$$\tau(C) = \frac{k}{|C|} \tag{2.5}$$

This term requires smaller segments to have stronger evidence for a boundary. k is a parameter of the system and sets a scale for the segmentation (larger k causes larger segments). This parameter influences the size and number of segments, but it is neither possible to enforce a minimum segment size nor a certain number of segments.

The function τ can be any non-negative function. E.g. the authors suggest to use the ratio of perimeter to area to suppress long and thin segments. However, with the current formulation, the algorithm would enforce just one of each neighboured segment pair to be of the desired shape. An open source C++ implementation by the authors is available[3]. In this implementation there exists a postprocessing step to enforce a minimum segment size by merging too small segments with the neighboured components to which they are connected by the smallest weight.

2.3.2.3. Edge Augmented Mean Shift (EAMS)

Properties EAMS	
Technique	Mean shift and edge detection
Cue(s)	Intensity, colour, location
Compact?	Possible
Runtime	A few seconds

Fig. 2.6.: Example EAMS segmentation

Edge Augmented Mean Shift [Christoudias et al., 2002] builds on two modules: mean shift segmentation and a confidence based edge detector. Mean shift is a mode seeking algorithm originally proposed by Fukunaga and Hostetler [2006]. Given a set of discrete data samples and an initial estimate of the modes of the density function underlying these samples, mean shift iteratively computes a kernel-based weighted combination of nearby samples to re-estimate the location in the feature space. A major difference to the k-means algorithm is that mean shift does not need to know the number of cluster centres in advance. Comaniciu and Meer [2002] describe its application to image segmentation. They propose a two steps approach:

1. First the image is represented in the five dimensional Luv-colour and spatial feature space and filtered using mean shift. In this step, each pixel converges to a mode.

[3] `http://www.cs.brown.edu/~pff/segment/` [Last accessed: 2015-08-14]

2. In a second step, the basins of attraction of these modes are recursively fused until convergence (controlled by spatial and colour resolution parameters). Additionally too small regions may be eliminated.

The used confidence based edge detector is a generalization of the traditional Canny edge detector and described in [Meer and Georgescu, 2001]. The edge detector is integrated in the mean shift filtering step. An implementation is available as part of the EDISON system[4] [Christoudias et al., 2002]

2.3.2.4. Quickshift (QS)

Properties QS	
Technique	Medoid shift
Cue(s)	Intensity, colour, location
Compact?	Yes
Runtime	A few seconds

Fig. 2.7.: Example QS segmentation

The above mean shift algorithm implements the mode seeking approach essentially by gradient ascent [Cheng, 1995]. This requires additional structure of the underlying data space to define a gradient (e.g. Hilbert space or smooth manifold structure [Vedaldi and Soatto, 2008]). Sheikh et al. [2007] proposed medoid shift to overcome this limitation. Medoid shift also starts with a kernel density estimate of the probability distribution of the data points but constrains the trajectories to pass through the data points. However, the runtime of general medoid shift is considerably higher than that of mean shift. For N data points with dimensionality d, the complexity of general medoid shift is $O(dN^2 + N^{2.38})$ (unfortunately with high constant) and that of mean shift $O(dN^2T)$ with T being the number of iterations which is normally much smaller than N. However, Vedaldi and Soatto [2008] show that the computational complexity of medoid shift in *Euclidean* space is only $O(dN^2)$, which is faster than mean shift. They further point out that medoid shift is not able to consistently identify all modes of the density. This issue can be addressed e.g. by reiterating medoid shift [Sheikh et al., 2007] or by using medoid shift as preprocessing step to simplify the data for mean shift. Vedaldi and Soatto [2008] also propose a further approach called Quickshift, which turned out to be a useful superpixel segmentation algorithm: While mean shift moves the data points in the underlying space, medoid shift restricts these paths to data points. In terms of graph theory the resulting set of paths of the points can be seen as a forest. Quickshift extends this idea by directly constructing a tree of paths by moving each data point to

[4] `http://www.wisdom.weizmann.ac.il/~bagon/matlab.html` [Last accessed: 2015-08-14]

the nearest neighbour for which there is an increment of the density. The final clusters are obtained by splitting the branches of the tree based on a threshold λ. To obtain superpixels from an image, Vedaldi and Soatto [2008] use Quickshift to cluster the image pixels in the combined Luv color and spatial position space (as in [Comaniciu and Meer, 2002]). An implementation is available as part of the VLFeat library[5].

2.3.2.5. Watershed (WS)

Properties WS	
Technique	Seeded region growing
Cue(s)	Intensity, colour
Compact?	No
Runtime	Milliseconds

Fig. 2.8.: Example WS segmentation

The idea of watershed segmentation originates from [Digabel and Lantuéjoul, 1978]. The intuitive idea comes from geography: when a landscape is flooded by falling waterdrops, dependent on the amount of water there are basins filled with water and dividing ridges between them. These ridges are the watersheds. Since watershed segmentation is a well known algorithm, there are various algorithmic implementations and adaptations, see [Roerdink and Meijster, 2000] for an overview.

An open source implementation is available as part of OpenCV[6]. OpenCV implements a seeded watershed segmentation (also called marker controlled watershed). The seeds are externally provided to the algorithm. The choice of the markers has a large influence on the resulting segmentation. For example, [Meyer, 1992] use local gradient minima. The seeds grow iteratively pixel by pixel until they reach a border to the segment around another seed. These borders form the watersheds. The next seed to expand by one pixel is chosen based on the priority obtained from a distance function. In the OpenCV implementation, this distance function is based on the intensity or colour value of the pixels. Additionally to this basic seeded watershed starting from local gradient minima, section 4.4 will present a novel adaptation, coined Compact Watershed, that initializes the seeds to form a superpixel lattice and incorporates spatial constraints in the construction of the superpixels. This section will also give more details on the algorithmic approach.

[5] `http://www.vlfeat.org` [Last accessed: 2015-08-14]
[6] `http://www.opencv.org` [Last accessed: 2015-08-14]

2.3.2.6. Entropy Rate Superpixel Segmentation (ERS)

Properties ERS	
Technique	Graph merging
Cue(s)	Intensity, colour, size
Compact?	Yes
Runtime	About a second

Fig. 2.9.: Example ERS segmentation

Liu et al. [2011] propose a new objective function for superpixel segmentation: the combination of the entropy of a random walk on an image graph representation and a balancing term on the size of the segments. In their formulation, a high entropy favours homogeneous segments. The *entropy rate* is the asymptotic measure of the remaining uncertainty of the random process of adding edges to form clusters. They show that their objective function is submodular (the discrete analogue to convexity in continuous domains) and exploit this property in a greedy algorithm that approximates the optimal solution by a bound of $\frac{1}{2}$. Their algorithm starts with an empty set of edges and sequentially adds the edge that yields the largest gain of their objective function. They iterate until the right number of clusters is reached. For speed-up, they ignore edges that would introduce cycles, which reduces the solution space to the set of forests. This approach computes a hierarchy of segments. More details can also be found in their journal paper [Liu et al.]. An implementation is available on Github[7].

2.3.2.7. Simple Linear Iterative Clustering (SLIC)

Properties SLIC	
Technique	Local k-Means
Cue(s)	Intensity, colour, location
Compact?	Yes
Runtime	A few hundred ms

Fig. 2.10.: Example SLIC segmentation

Simple Linear Iterative Clustering (SLIC) [Achanta et al., 2012] is based on the concept of a local k-means clustering. The cluster centres are initialized on a uniform grid in the

[7] `https://github.com/mingyuliutw/ers` [Last accessed: 2015-08-14]

image plane. *Local k-means* denotes that each cluster considers only pixels in its spatial neighbourhood. The metric during clustering is a weighted combination of LAB colour distance and spatial distance in the image plane. The weight of the spatial component (the "compactness" parameter) influences the regularity of the resulting segments shape and size.

There are two implementations available: one by the authors[8] and another as part of VLFeat[9]. The implementation of the authors is slightly different from the description in their paper: they use a modified distance function and a fixed number of iterations. Thera are several adaptions of SLIC. E.g. Schick et al. propose to evaluate only the boundary pixels during the k-means update. However, this makes the algorithm more sensitive to image gradients. Section 4.2 presents a novel modified version of SLIC, called Preemptive SLIC, that yields a significant speed-up by introducing a local termination criterion for each segment. This section also gives more details on the algorithmic approach of SLIC.

2.3.2.8. Superpixels Extracted via Energy-Driven Sampling (SEEDS)

Properties SEEDS	
Technique	Moving boundary pixels
Cue(s)	Intensity, colour, smooth boundaries
Compact?	Not by default, but there is a compact variant available
Runtime	About one hundred ms

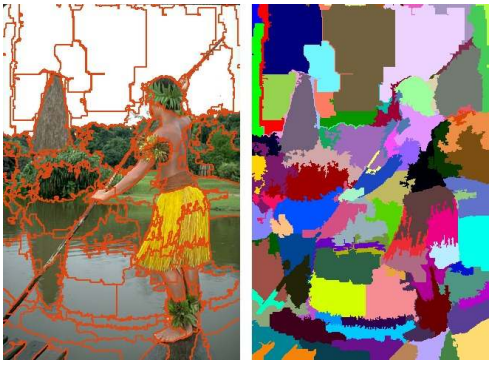


Fig. 2.11.: Example SEEDS segmentation

Van den Bergh et al. [2012] describe a different approach on generating superpixels. Starting from a grid of square superpixels, they iteratively reassign boundary pixels (or blocks of pixels) from one segment to a neighboured segment. This way they start from a complete segmentation and subsequently adapt it to the image content. They use an objective function that enforces homogeneity of the segment colour distribution (in LAB colour space) and smooth boundary shapes. Boundary smoothness is measured by building histograms of superpixel labels in small local patches (e.g. 3×3). Inside superpixels there will appear only a single superpixel label while at the borders there will be different labels. They use the sum of squared histogram entries as measure for the smoothness of the patch. Both terms in this objective function are designed for very fast optimization by a greedy strategy. They can be evaluated and updated very fast for changing segment boundaries. In each step, the algorithm proposes a set of single pixels

[8] oriSLIC: `http://ivrg.epfl.ch/supplementary_material/RK_SLICSuperpixels/index.html` [Last accessed: 2015-08-14]

[9] vlSLIC: `http://www.vlfeat.org/` [Last accessed: 2015-08-14]

or blocks of pixels at segment boundaries to be exchanged between the two neighboured segments. Invalid proposals (e.g. when a segment is split) are ignored. If the new segmentation provides a gain of the objective function, it is retained, otherwise it is rejected. In [Van den Bergh et al., 2013] two other boundary terms are proposed: First a compactness term (average distance of pixels to the centre of gravity of the superpixel). Second an edge prior that computes LAB colour edges and snaps segment boundaries to these edges. They evaluate the boundary terms and their combinations and conclude that all boundary terms negatively influence boundary recall and should be omitted if no compactness property is necessary. An implementation is available directly from the authors website[10] or as part of OpenCV[11] (Version $\geq$ 3.0).

2.3.2.9. Veksler superpixels

Properties VEK	
Technique	Energy minimization by graph cuts
Cue(s)	Intensity, size
Compact?	Yes
Runtime	A few seconds

Fig. 2.12.: Example VEK segmentation

Veksler et al. [2010] propose another graph cut based approach for superpixel tessellation that focuses on regular partition. They formulate the segmentation problem as an energy minimization problem that explicitly encourages regular superpixels.

Given a set of pixels P and a set of segment labels L, they search a labelling f that assigns each pixel $p = (x_p, y_p)^T$ the label f_p. Their energy function builds on their prior work on graph cut optimization [Boykov et al., 2001] and is the weighted sum of a data and a smoothness term:

$$E(f) = \sum_{p \in P} D_p(f_p) + \lambda \sum_{\{p,q\} \in N} w_{pq} \cdot V_{pq}(f_p, f_q) \tag{2.6}$$

Unary constraints $D_p(f_p)$ express how likely is the label F_p for pixel p. Binary constraints $V_{pq}(f_p, f_q)$ describe how likely labels f_p and f_q are for neighbouring pixels p and q. Typically, the binary constraint supports the common labels for similar pixels and inhibits common labels for pixels with e.g. a high gradient. They use an 8-connected neighbourhood and Potts model $V_{pq}(f_p, f_q) = \min(1, |f_p - f_q|^1)$. The weights w_{pq} describe the similarity between pixels. They are inversely proportional to image intensity

[10] `http://www.mvdblive.org/seeds/` [Last accessed: 2015-08-14]
[11] `http://www.opencv.org` [Last accessed: 2015-08-14]

gradients, however, one could also incorporate other cues like colour or texture gradients. For the data term, they start with a grid like initialization of the segment midpoints $m(l) = (x, y)^T$. They set the data term such that nearby pixels may have the same segment label, while for pixels with distances larger than $maxPatchSize$ there exists no common label with finite energy. The open source implementation[12] by the authors provides two variants for the unary term:

1. Compact Superpixels (coined VEK in our experiments) with data term:

$$D_p(l) = \begin{cases} 1 & \text{if} \quad ||p - m(l)||_1 \leq \text{maxPatchSize} \\ \infty & \text{otherwise} \end{cases} \tag{2.7}$$

   Since this term is invariant for all valid labellings (those without ∞ terms), the weighting parameter λ in Equation 2.6 has no influence. Thus, the control over boundary smoothness is implicit in the binary terms and can (and need) not be adjusted by an additional parameter.

2. Constant Intensity Superpixels (coined VEK CI) with data term:

$$D_p(l) = \begin{cases} |I_p - I_{m(l)}| & \text{if} \quad ||p - m(l)||_1 \leq \text{maxPatchSize} \\ \infty & \text{otherwise} \end{cases} \tag{2.8}$$

   Here, the unary term depends on the similarity of the pixel to the midpoint of the segment. This requires an additional constraint in the label of the midpoints, see [Veksler et al., 2010] for details.

In both cases, the number of resulting segments is controlled by the $maxPatchSize$ parameter. While optimizing this energy is NP-hard, the graph cut minimization method described in [Boykov et al., 2001] guarantees a factor of 2 approximation. Compared to NC, this graph based technique is much faster and produces visually similar results. Moreover, there is a straight forward extension to image sequences (or 3d data) to create supervoxels [Veksler et al., 2010].

2.3.3. Summary of the presented superpixel segmentation algorithms

Before we dive deeper into a detailed evaluation of these algorithms, I want to summarize the presented algorithms. Table 2.1 provides an overview of the theoretical properties of the algorithms. Fig. 2.13 shows example segmentations with varying numbers of segments. Section 3.2.3 will present the available implementations of these approaches including parameter settings that are used in the upcoming experiments.

[12] `http://www.csd.uwo.ca/faculty/olga/code.html` [Last accessed: 2015-08-14]

	Base Technique	Runtime	Intensity	Colour	Texture	Location	Size	Compact	Grid Initialization	Enforce Connectivity	Programming Language[a]	Licence	Publication
NC	Graph Cuts	Minutes to hours	✓	✓	✓	✓	✓	✓	-	-[b]	C	GPL	Shi and Malik [1997]
FH	Graph merging	About hundred ms	✓	✓	-	-	-	-	-	✓	C	GPL	Felzenszwalb and Huttenlocher [2004]
EAMS	Mean-shift	A few seconds	✓	✓	-	✓	-	✓	-	✓	C, M	-	Comaniciu and Meer [2002]
QS	Medoid-shift	A few seconds	✓	✓	-	✓	-	✓	-	(✓)[c]	C, M	BSD	Vedaldi and Soatto [2008]
WS	Seeded region growing	Some milliseconds	✓	✓	-	-	-	-	-	✓	C, P	BSD	Meyer [1992]
ERS	Random walks	About a second	✓	✓	-	-	✓	✓	-	✓	C, M	[d]	Liu et al. [2011]
SLIC	k-means	A few hundred ms	✓	✓	-	✓	-	✓	✓	(✓)	C, M	GPL	Achanta et al. [2012]
SEEDS	Move boundary pixels	About hundred ms	✓	✓	-	-	-	(✓)	✓	✓	C, M	GPL	Van den Bergh et al. [2012]
VEK	Graph Cuts	A few seconds	✓	-	-	-	-	✓	✓	(✓)	C++	[e]	Veksler et al. [2010]
CWS	Seeded region growing	Some milliseconds	✓	✓	-	✓	-	✓	✓	✓	C, M	GPL	Neubert and Protzel [2014]
pSLIC	k-means	Tens of ms	✓	✓	-	✓	-	✓	✓	✓	C, M	GPL	Neubert and Protzel [2014]

Table 2.1.: Overview of the compared algorithms. The two bottom lines show the novel algorithms CWS and pSLIC that are presented in chapter 4.

[a] M Matlab, P Python, C C/C++
[b] There are scattered single pixels in the used implementation.
[c] (✓) indicates that in rare cases or depending on the implementation segments may be not connected.
[d] Free for non-commercial usage.
[e] Free for research purposes only.

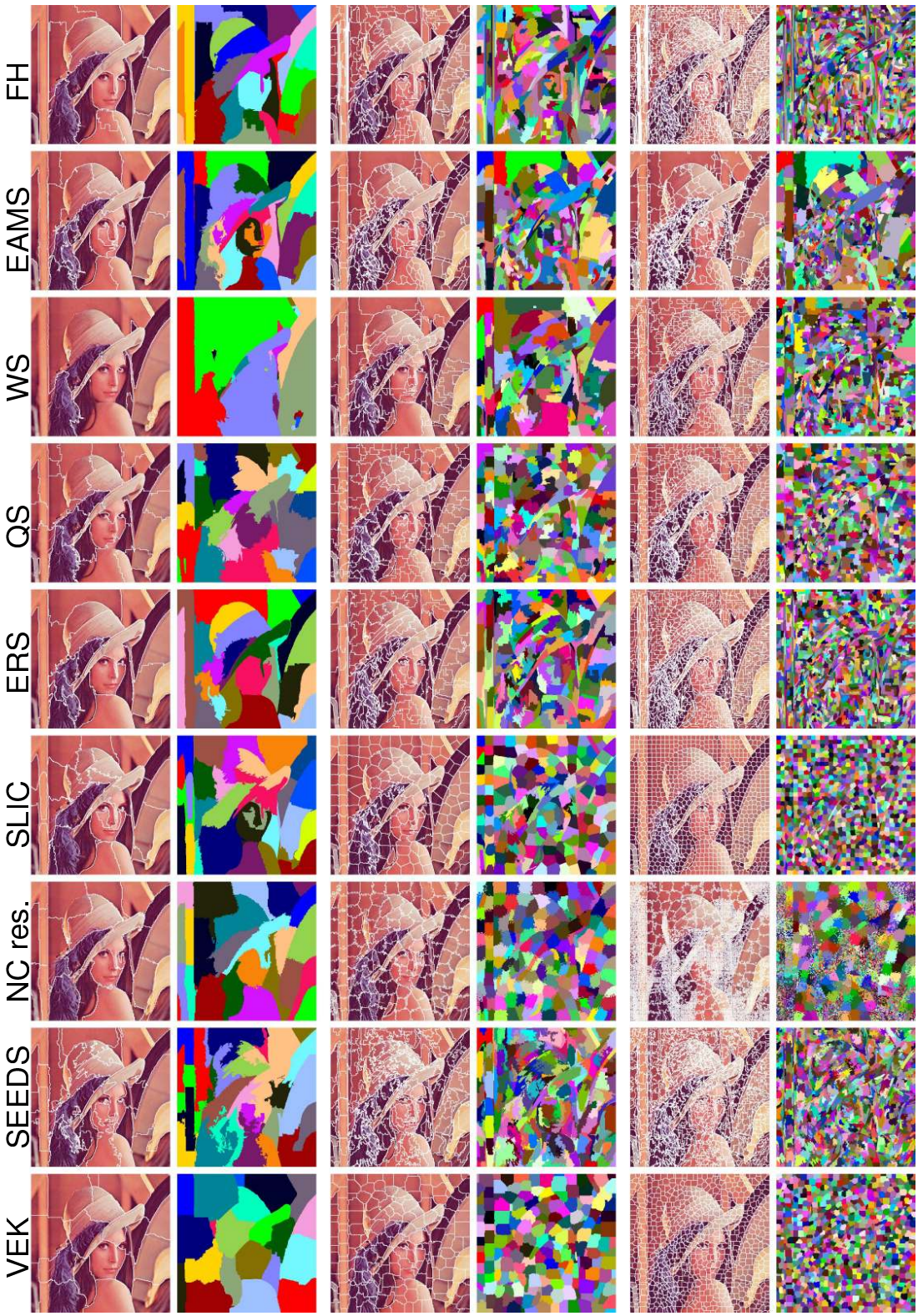


Fig. 2.13.: Example segmentations with about 25, 200 and 1000 superpixels.

To measure is to know.
If you can not measure it, you can not improve it.
Lord Kelvin

3 How to Compare Superpixel Segmentation Algorithms

The previous section revealed the broad range of available superpixel segmentation algorithms. Each of them with individual properties, advantages and disadvantages. Depending on the application at hand, a different superpixel algorithm may be the preferred choice. To decide which algorithm might be the most suitable, two types of knowledge are crucial: knowledge about the requirements of the application and about the properties of the superpixel algorithms. While there exist many superpixel segmentation algorithms, there is a lack of a common base to compare their properties and to facilitate the right choice. In this part of my thesis I will work on establishing a common base by four means:

1. Identifying and discussing important requirements on these algorithms. Namely
   - Runtime
   - Segmentation quality
   - Robustness
   - Compactness

   This is done in Section 3.1 together with a review of related work and common practise for comparing superpixel algorithms - this will emphasize the need for a standardized benchmark.
2. The four sections 3.3 to 3.6 will propose a combination of established and novel metrics to evaluate each of the above requirements on superpixel algorithms.
3. These metrics are combined to a standardized benchmark, including implementations of the metrics, datasets and usage guidelines (cf. section 1.3 on open source software releases).

4. Finally, the algorithms of the previous chapter 2 are compared using this benchmark. Since there are several metrics presented for the evaluation of the criteria, the experimental results for each criterion are presented directly after the discussion of the respective metrics in the sections 3.3 to 3.6.

More generally, this section is about the comparison of superpixel segmentation algorithms on two levels of abstraction, approaching the two questions: How good are the available superpixel segmentation algorithms? And on a more abstract level: What are criteria for a good superpixel segmentation and what are metrics to measure them? The answers to these questions will guide the direction towards the two novel segmentation algorithms presented in the following chapter 4.

3.1. Superpixel segmentation benchmarks

There has been a lot of research on superpixels since the term has been established in Ren and Malik [2003]. In particular, the previous sections showed the diversity of superpixel segmentation algorithms that have been proposed. However, limited work has been done on comparing these algorithms and on evaluating their properties. The state of the art method to compare and benchmark superpixel segmentation algorithms is to evaluate their capability to recover human figure-ground segmentations. The Berkeley Segmentation Dataset and Benchmark (BSDS500) [Arbelaez et al., 2011] is a commonly used comparison framework for segmentation algorithms including superpixel segmentations [Achanta et al., 2012; Van den Bergh et al., 2012; Arbelaez et al., 2014]. It consists of 500 manually segmented images where humans were asked to outline object boundaries in the images to obtain ground truth segmentations. Although there exist multiple manual segmentations for each image, the ground truth data depends on the semantic interpretations of objects and their boundaries by the human annotators. Other resources for manually annotated segmentations are e.g. the PASCAL VOC challenge [Everingham et al., 2010] and the MSRC [Shotton et al., 2009] datasets. To overcome the dependency on manual ground truth segmentations, Moore et al. [2008] propose to use Explained Variation. This error metric describes the proportion of image variation that is explained if all pixelvalues within a superpixel were replaced with the superpixel mean colour. Although they established a human independent metric, Explained Variation has the drawback of penalizing segments with consistent texture but large pixel variance. Koniusz and Mikolajczyk [2009] measure the robustness of superpixel segmentations indirectly by the repeatability of features extracted from the segments. Their evaluation depends on the additional processing step of feature extraction and is based on a small dataset consisting of 48 images. Xu and Corso [2012] evaluate several supervoxel methods based on video data. A supervoxel is the video equivalent to a superpixel and covers a subset of the spatio-temporal lattice composed by the concatenated video frames such that all supervoxels of a video comprise the full lattice and are pairwise disjointed. They evaluate how well supervoxels cover human video segmentations based on combined spatio-temporal error metrics and Explained Variation [Moore et al., 2008].

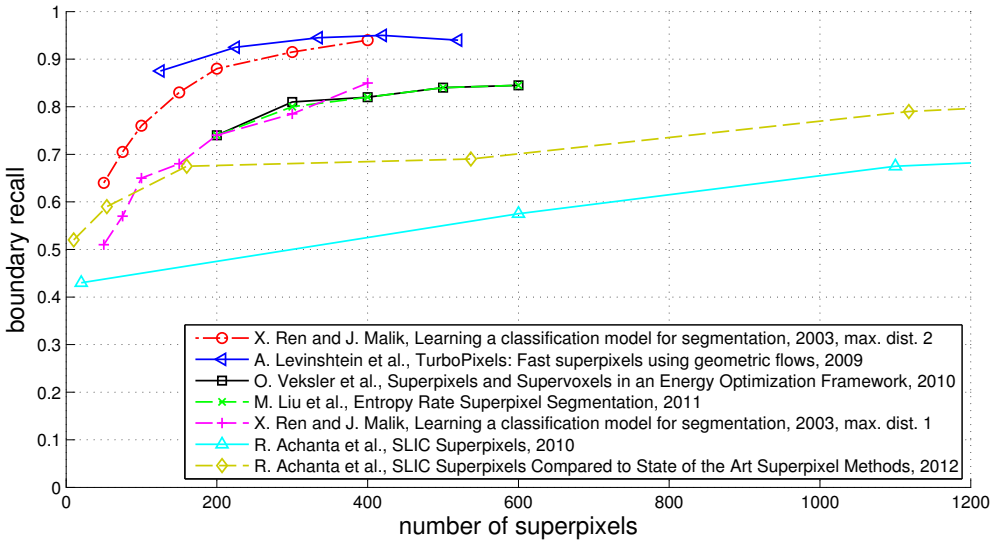


Fig. 3.1.: Why we need a standardized benchmark: This is a collection of plotted curves from different papers showing the performance of the Normalized Cuts superpixel segmentation algorithm. In each paper one of these curves was used to compare a new superpixel algorithm to this established segmentation algorithm. Thus, each curve was intended to show the result of the same algorithm using the *same* implementation, the *same* data set and the *same* error metric. The apparent differences in the curves emphasize the need for a standardized benchmark.

Further information on the evaluation of video segmentations can be found in [Galasso et al., 2013]. Other resources for comparison of oversegmentation algorithms with respect to a particular application are [Strassburg et al., 2015] for image parsing and [Ge et al., 2006] in the context of image saliency.

I want to emphasize two reasons why more work on superpixel benchmarks is needed:

1. The existing comparison procedures do not yield comparable results.
2. There exist more requirements than what can be measured by comparison with figure-ground segmentations, i.e. regarding the segmentation stability or compactness

Let me illustrate the first reason: Newly presented algorithms are often compared to a couple of established algorithms for which publicly available implementations exist. Due to its broad publicity and its free implementation, superpixel segmentation based on Normalized Cuts [Ren and Malik, 2003] (cf. section 2.3.2.1) is one of the commonly used algorithms for comparison. Together with the Berkeley Segmentation Data Set [Arbelaez et al., 2011] and the two error metrics "boundary recall" and "undersegmentation error", this is a repetitive comparison framework in the literature. However, there are small variations in the usage of the dataset or implementation of the error metric that cause major differences in the benchmark results. This is illustrated in Fig. 3.1. In

this plot each curve originates from a paper where the authors report to use the freely available implementation of the Normalized Cuts superpixel algorithm, together with the Berkeley Segmentation Data Set and the boundary recall error metric. One would expect all papers to report the same curve. Nevertheless, the results are apparently different. A closer look reveals small differences in the usage of the dataset (e.g. Is the full dataset used or just a part? How is the multiple ground truth handled?) and the implementations of the error metrics (e.g. How are boundary recall and undersegmentation error implemented? What is the threshold on boundary recall?). The differences between the shown curves in Fig. 3.1 underline the demand for a standardized benchmark that this part of the thesis addresses.

Result 3.1 *There is a need for a standardized superpixel segmentation benchmark.*

The remainder of this chapter 3 presents own contributions to comparison of superpixel segmentation algorithms based on the four aforementioned criteria: runtime, segmentation quality, stability and compactness. It is based on the previous publications in [Neubert and Protzel, 2012] and [Neubert and Protzel, 2013]. The first paper approaches the described issue regarding standards for benchmarking superpixel segmentation algorithms. It presents a standardized benchmark based on figure-ground segmentations including an open source implementation of all parts: from accessing data, over implementation of all metrics, up to creation of the figures. The **segmentation quality evaluation** in section 3.4 presents an enhanced version of this benchmark including an extended set of metrics and results on more algorithms.

In [Neubert and Protzel, 2012], we also made a first step towards identification and evaluation of superpixel properties beyond the accordance with human made manual segmentations, the second reason from the above listing why more work on superpixel benchmarks is needed. The application of superpixel algorithms on multiple images showing the same scene or object raise the question about the stability of the extracted segments: Are the same (or at least similar) segments extracted across different views or different imaging conditions? While superpixel borders at considerable image gradients may constantly be detected, oversegmentation algorithms tend to create lots of spurious segment borders that vary strongly under image changes. Even slight changes of the image, e.g. a small camera motion or changes in lighting, can cause substantial changes of the produced segmentation. For some applications this might be irrelevant, while others could benefit from a superpixel algorithm with more stable segmentations.

In [Neubert and Protzel, 2012], we evaluate the stability in terms of the algorithm's robustness towards affine image transformations. The transformations are manually chosen and synthetically applied on the test images. However, affine image transformations do not cover all relevant image transformations, e.g. they exclude changing lighting conditions, occlusions and image noise. To overcome the limitations of manually transformed images, we propose to use ground truth optical flow data to evaluate the performance on video data in [Neubert and Protzel, 2013]. Moreover, the usage of ground truth optical flow allows us to overcome the dependence on human ground

truth segmentations biased towards semantic interpretation. The **segmentation stability evaluation** in section 3.5 builds upon this prior work and presents an extended set of metrics to measure the repeatability of superpixel segmentations in the presence of artificial and real world image changes including viewpoint changes, moving objects, illumination changes, noise and motion blur.

The previously shown example segmentations (e.g. in Fig. 2.13) give an idea of the different possible characteristics of superpixel algorithms regarding size, shape and spatial distribution of the segments. Section 3.6 discusses the importance of these characteristics and measures them based on a **segmentation compactness evaluation**.

Contribution 3.1 *This section identified segmentation runtime, quality, stability and compactness to be relevant properties of superpixel segmentation algorithms.*

For each of the mentioned evaluations of runtime, quality, stability and compactness, first the fundamentals in form of the used metrics are provided, followed by experimental results for the algorithms presented in section 2.3.2. Table 3.1 gives an overview of the evaluation metrics and the conducted experiments. To pave the way for these evaluations, the following section describes the experimental setup including a presentation of the datasets and details on the used implementations and parameters of the segmentation algorithms.

3.2. Experimental setup

There are important differences between comparing theoretical properties of algorithms (e.g. asymptotic runtime) or numerical evaluations based on test images. For the latter, the performance of the algorithm strongly depends on practical aspects like parameter choice, used image cues, involved preprocessing steps, implementation details and the used test data. For example, an algorithm may perform superior to another just due to a better parameter choice for the regarded test data or an additional preprocessing step that is not part of the theoretical algorithm, but of the available implementation. Section 3.2.3 gives details on the used implementations and parameters of the evaluated superpixel algorithms, e.g. there are algorithms for which multiple implementations are evaluated.

The results of a numerical evaluation also depend on the used datasets. The overview in table 3.1 includes the most important requirements on the datasets for each metric. The evaluation of segmentation quality is based on images with known object-ground segmentation by human annotators. Other metrics, for the evaluation of segmentation robustness and compactness, require image sequences with known ground truth optical flow. The following two sections discuss these requirements and present the datasets used in the subsequent experiments. This completes the experimental setup up to the metrics which are presented in the corresponding sections, following the description of the datasets and implementations.

Criteria	Metric	Required data	Used dataset	Section
Runtime		Arbitrary images	BSDS500 test	3.3
Quality	Boundary recall	Manually annotated figure-ground segmentations	BSDS500 test	3.4.1
	Undersegmentation Error	Manually annotated figure-ground segmentations	BSDS500 test	3.4.2
	MASA	Manually annotated figure-ground segmentations	BSDS500 test	3.4.3
Stability	Stability to affine image transformations measured by Precision-Recall	Arbitrary images, transforms and transformed images	BSDS500 test	3.5.1
	Stability to image noise measured by Precision-Recall	Arbitrary images, noise models	BSDS500 test	3.5.2
	Motion Undersegmentation Error	Image sequences with ground truth optical flow	Sintel, KITTI	3.5.3
Compactness	Geometric properties: Standard deviation of size; isodiametric and isoperimetric quotients	Arbitrary images	BSDS500 test	3.6.1
	Motion Discontinuity Error	Image sequences with ground truth optical flow	Sintel, KITTI	3.6.2

Table 3.1.: Overview of conducted superpixel algorithm evaluations

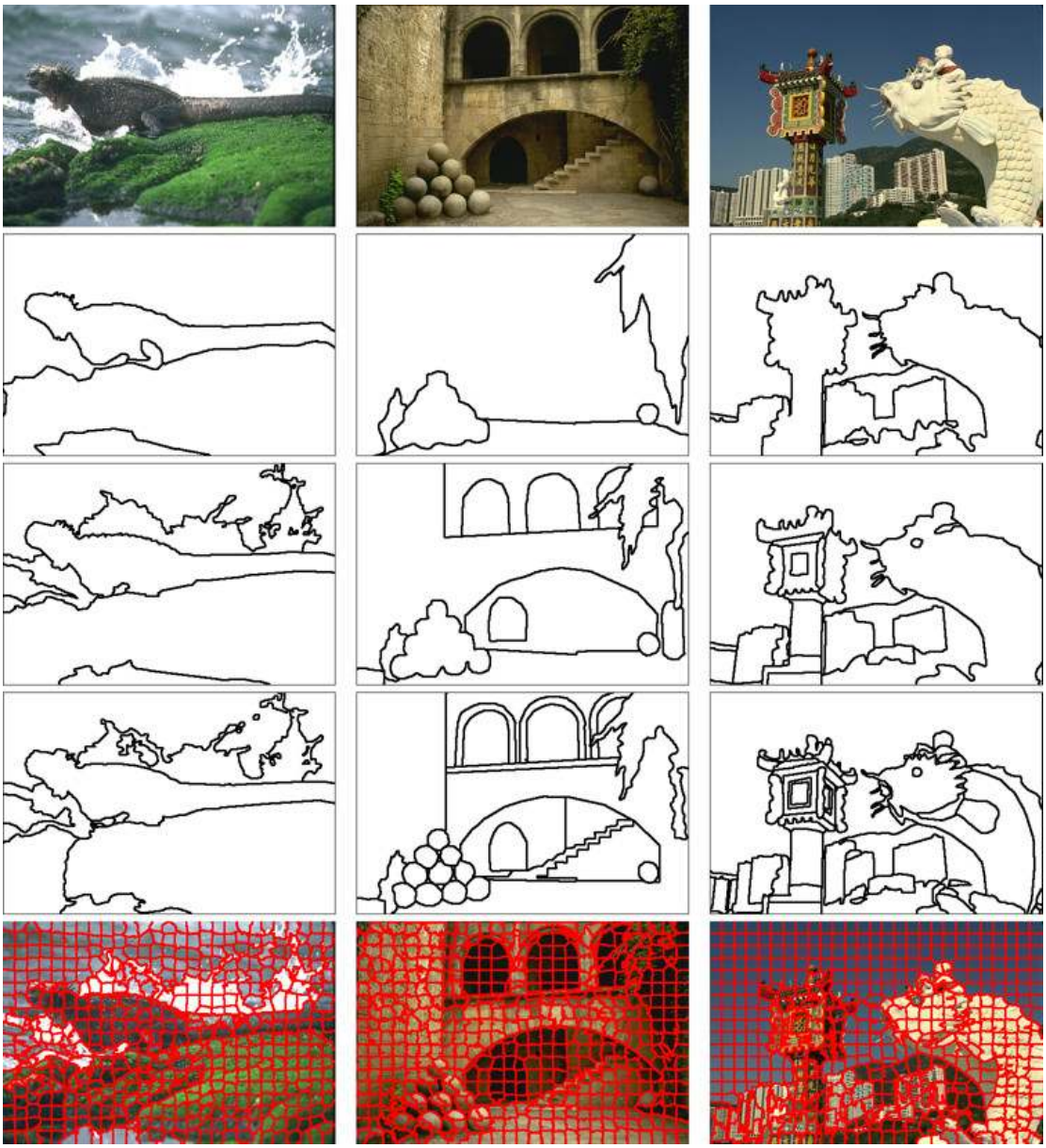


Fig. 3.2.: The first row shows example images from the BSDS500 dataset [Arbelaez et al., 2011]. Below each image, there are three manual segmentations, each by a different human annotator. These ground truth segmentations can have very different levels of detail and complexity. Nevertheless, none of them can be considered more true than the other - a consequence of the ill-posed segmentation task. The bottom row shows an example superpixel segmentation using the proposed PreemtiveSLIC algorithm.

3.2.1. Figure-ground segmentation dataset

Evaluating superpixel segmentations based on figure-ground segmentations requires particularly suited error metrics and obviously knowledge about ground truth figure-ground segmentations. Since its presentation in 2001, the Berkeley Segmentation Dataset (BSDS) [Martin et al., 2001] is the standard database for evaluation of figure-ground segmentation algorithms and provides such ground truth segmentations. The original dataset consists of 300 colour images with ground truth object boundaries by human annotators. The 481x321 RGB images are part of the Corel image database and show everyday scenes. In 2011 the dataset has been extended to 500 images [Arbelaez et al., 2011]. The two datasets are distinguished as BSDS300 and BSDS500. Other resources for ground truth segmentations are object recognition databases with pixel wise labels like the MSRC database [Shotton et al., 2006], PASCAL VOC [Everingham et al., 2014] or LabelMe [Russell et al., 2008] datasets. MSRC contains 591 natural images with objects belonging to 21 classes. The training and validation datasets of the segmentation datasets of the 2008 PASCAL segmentation challenge provide 1023 images with pixel wise labels for 20 object categories. While other large scale recognition datasets like ImageNet [Deng et al., 2009] do not include object outlines, LabelMe provides several thousand images with rough polygonal outlines of objects.

Since bottom-up image segmentation is an ill-posed problem, often, there is no unique solution for outlining objects in images. Moreover when humans outline object boundaries, they almost inevitably incorporate semantic knowledge about the objects which is inaccessible for bottom-up segmentation algorithms. The BSDS takes this into account by providing multiple ground truth segmentations for all images, each by a different human annotator.

Due to BSDS's multiple ground truth annotations and its accuracy, the superpixel segmentation comparison in this work is based on images from BSDS500. Example images and ground truth segmentations can be seen in Fig. 3.2. The dataset is divided in three parts:

1. **Train** There are 200 images for learning purposes.
2. **Val** The cross-validation dataset contains 100 images for adjusting the parameters of an algorithm.
3. **Test** The 200 test set images are for the final comparison of different algorithms.

3.2.2. Using datasets with ground truth optical flow

Based on the above described datasets, the capability of a segmentation algorithm to resemble human-made figure-ground segmentations can be evaluated. Besides runtime, this is an often used criteria, e.g. in [Achanta et al., 2012; Van den Bergh et al., 2012; Arbelaez et al., 2014]. While this is a crucial capability, section 3.1 discussed the segmentation stability as another important property. The evaluation of an algorithm's stability can be based on datasets with known ground-truth optical flow [Neubert and

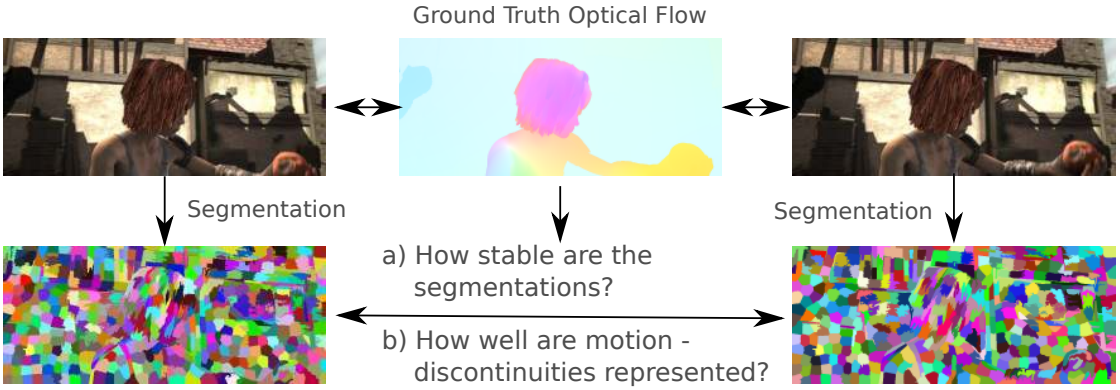


Fig. 3.3.: In [Neubert and Protzel, 2013] we propose to use ground truth optical flow fields to compare superpixel segmentation algorithms. On top left and right are two Sintel images with slight motion visualized by the optical flow field between them. Motion direction is coded by hue, saturations codes the motion magnitude, see [Baker et al., 2011] for details. Beneath the images there are example superpixel segmentations (using *ERS*). While some object contours are visible, there are a lot of additional segment boundaries. The answers to the two posed questions are part of the here presented comparison.

Protzel, 2013]. Given an image pair I_1 and I_2 from an image sequence, the optical flow is the vector field describing the motion of each image point between I_1 and I_2. Beyond evaluation of the stability, the knowledge of the pixel motion can also be used to evaluate whether the additional segment boundaries that do not belong to object boundaries (as they are outlined in manual figure-ground segmentations) carry useful information, e.g. about motion discontinuities. Figure 3.3 illustrates the two criteria that will be evaluated using ground-truth optical flow.

Stability-Criteria Does the segmentation algorithm find the same regions or object boundaries independent of changes in the image?

Discontinuity-Criteria How well are motion discontinuities in the image sequence represented by the algorithm's segment boundaries? E.g. the motion gradient between a moving foreground object and the background or in the interior of a non-rigid object.

The repeatability of segmentation boundaries, the first criteria, is the subject of the presented stability evaluation in section 3.5 and the second question is related to the compactness properties evaluated in section 3.6.

Contribution 3.2 *This section presented the idea of using datasets with ground truth optical flow to evaluate superpixel segmentations.*

Fig. 3.4.: The top row shows two example images of the KITTI dataset [Geiger et al., 2012]. The flow field (bottom-left) illustrates the forward motion of the car mounted camera. Pixels with invalid optical flow are black. The bottom-right image shows the result of applying the flow field on the first image: Large parts of the view of the second image are recovered with pixels of the first image. However the many missing parts emphasize the advantages of synthetic datasets with much denser ground truth flow fields.

This part of the thesis builds upon the ground truth optical flow datasets already described in [Neubert and Protzel, 2013][1]. There exist several datasets for evaluation of optical flow algorithms. Our benchmark is based on the KITTI [Geiger et al., 2012] and the Sintel [Butler et al., 2012] datasets. While the Middlebury dataset [Baker et al., 2011] is an established optical flow dataset, the amount of data with public ground truth is limited to eight sequences, each with up to eight frames. Beyond Middlebury, several large scale real world datasets for evaluation of optical flow algorithms have been published: KITTI [Geiger et al., 2012], HCI [Meister et al.] and a collection on the Image Sequence Analysis Test Site (EISATS)[2]. Due to the amount of data and the provided ground truth, we decided to use the KITTI dataset for our evaluation. Image source is a stereo camera mounted on a driving car, thus there is severe camera motion in subsequent images. We only consider the left images of the stereo pairs. Ground truth optical flow is available for the training subset of the original KITTI dataset, resulting in 194 gray level image pairs with a size of 1226×370 pixels for our benchmark. Each image pair shows an individual street scene. The ground truth has been generated using a 3D laser scanner. Due to limited sensor range, occlusions and other restrictions in the ground truth computation, there is flow information for about 25 % of the image pixels (averaged over all image pairs). For example, there is no flow information for pixels showing the sky. Note that although there is camera data available for longer sequences (20 frames per sequence), no ground truth optical flow data is available for this extended dataset.

[1] The remaining part of this section 3.2.2 including Fig. 3.3 and Fig. 3.4 is published in very similar form in [Neubert and Protzel, 2013]

[2] `http://tinyurl.com/EISATS-flow` [Last accessed: 2015-08-14]

Fig. 3.5.: Each row shows an example pair of consecutive frames from the Sintel dataset [Butler et al., 2012] together with the ground truth optical flow field. For visualization of flow fields, the direction of the motion vector is coded by hue and its length by saturation, see Baker et al. [2011] for details. The Sintel flow fields are dense and accurate even for small motions.

Besides the real world datasets, computer rendered images are a great data source due to their near perfect ground truth information. The MPI Sintel Dataset [Butler et al., 2012] is based on the open source animated short film Sintel produced by Ton Roosendaal and the Blender Foundation. It provides naturalistic video sequences and is designed to encourage research on long-range motion, motion blur, multi-frame analysis and non-rigid motion. Moreover, the motion statistics of the dataset showed to be realistic [Butler et al., 2012]. MPI Sintel constitutes the second dataset in our benchmark. It consists of 23 scenes, each with 20 to 50 colour images with a size of 1024×436 pixels. These longer scenes allow us to combine flow fields to sequences over multiple frames. Moreover, there is much denser ground truth flow data. There is even ground truth motion for pixels that are occluded in one of the two scenes. This is possible since the ground truth optical flow is directly extracted from the data used for rendering. From the different levels of rendered details, the most realistic *final* rendering is used.

To exploit the ground truth optical flow, we transform the segmentations, by applying the optical flow on the segment labels stored in the label image. The result is illustrated in Fig. 3.4. To transform label images, inverse mapping with nearest neighbour interpolation is used. Given an image pair I_1 and I_2, there are several cases of pixels with non valid information between the two images: (1) Pixels that are in I_1 and disappear in I_2, e.g. they are occluded or have moved outside the image. (2) Pixels that are in I_2 but not in I_1, i.e. these are pixels that appear behind a moving object or intrude on the observed scene. (3) Pixels for that no valid ground truth motion could be generated due to further influences. For real world datasets, there are obvious reasons for non-perfect ground truth (like limitations of the sensors used to generate the ground

truth), but even in synthetic datasets there are few additional pixels marked as invalid in the ground truth flow field. These pixels have to be handled separately when the flow field is applied on an image. In the here presented evaluation, pixels for which there are no ground truth flow information are ignored in computation of the metrics. The high amount of invalid pixels in the real world datasets restricts the transformation of a sparse image, e.g. boundary images. Therefore, metrics based on transformed boundary images should be avoided when using such ground truth optical flow fields.

3.2.3. Compared algorithms

Section 2.3.2 introduced a selection of existing approaches to create superpixel segmentations. The numerical evaluation presented in this chapter 3 uses publicly available implementations of these algorithms. For each of the introduced algorithms there is at least one implementation available, for some there are either multiple variants by the respective authors or different implementations from different sources. We use the implementations listed in table 3.2.

However, a numerical benchmark, evaluating algorithms by applying them to images, strongly depends on the used implementation and, sometimes even more importantly, the used parameters. While parameters of an algorithm provide the user the chance to influence the results, they also pose the problem of finding a suitable parameter value. For the full evaluation of an algorithm, one could exhaustively test all possible parameter configurations. However, this is not feasible in practise for the here presented comparison. To provide a fair comparison, we used the default parameters for each implementation. If there are no default parameters available or not suitable for the used data, we select those parameter values that provided the best segmentation quality on the BSDS *validation* data in a small set of initial experiments.

In the benchmark, segmentations with varying numbers of superpixels are evaluated. Some segmentation algorithms provide a parameter to set the number of superpixels directly (while not all practically enforce this number). For the others the *single* parameter with the largest influence on the resulting number of superpixel is varied to create the different segmentations. This approach should facilitate a fair comparison with a feasible effort for parameter selection. The resulting parameter settings are listed in table 3.3.

For different algorithms, each having the number of segments as an direct parameter, the setting of this input number may differ since not all algorithms strictly enforce this number of segments, some require (slightly) larger values.

I want to emphasize that the usage of such a selected set of parameters poses a limitation to the generality of the results on the performance of an algorithmic approach. The benchmark is comparing a *certain implementation* listed in table 3.2 with the particular *parameter setup* given in table 3.3. However, for practical applications of such an algorithm, one has to make the same decisions on selecting a particular implementation and set of parameters. Moreover, implementations of the metrics presented in the following sections, are freely available (cf. section 1.3) and can be used to evaluate other implementations and further parameter settings.

Algorithm, Publication ID	Used implementation
Normalized Cuts, [Ren and Malik, 2003]	
NC res	Due to its runtime, the input for NC is resized to 160 pixels on the longer side before segmentation. `http://www.timotheecour.com/software/ncut/ncut.html`
Felzenszwalb-Huttenlocher Segmentation, [Felzenszwalb and Huttenlocher, 1998]	
FH	`http://www.cs.brown.edu/~pff/segment/`
Edge Augmented Mean Shift, [Meer and Georgescu, 2001; Comaniciu and Meer, 2002]	
EAMS	`http://www.wisdom.weizmann.ac.il/~bagon/matlab.html`
Quickshift, [Vedaldi and Soatto, 2008]	
QS	`http://www.vlfeat.org/`
Marker-Controlled Watershed Segmentation, [Meyer, 1992]	
WS	We use the OpenCV implementation with markers at gradient minima. The image is rescaled to obtain the desired number of local gradient minima. The runtime for distribution of the markers is omitted. `http://opencv.willowgarage.com/wiki/`
Entropy Rate Superpixel Segmentation, [Liu et al., 2011]	
ERS	`https://github.com/mingyuliutw/ers`
Simple Linear Iterative Clustering, [Achanta et al., 2012]	
oriSLIC	`http://ivrg.epfl.ch/supplementary_material/RK_SLICSuperpixels/index.html`
vlSLIC	`http://www.vlfeat.org/`
Superpixels Extracted via Energy-Driven Sampling , [Van den Bergh et al., 2012]	
SEEDS	`http://www.mvdblive.org/seeds/`
Veksler Superpixel, [Veksler et al., 2010]	
VEK	Compact variant `http://www.csd.uwo.ca/faculty/olga/code.html`
VEK CI	Constant intensity variant, available from the same website
Uniform grid	
BOX	For baseline comparison, BOX simply divides the image into a regular grid with the given number of segments.

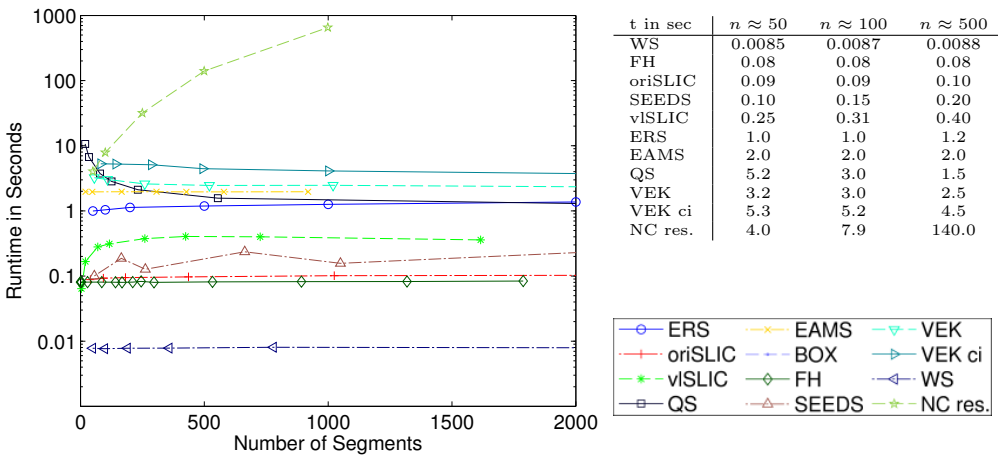
Table 3.2.: Listing of the compared implementations. All websites last accessed: 2015-08-14.

ID	Fixed parameters	Varied parameters
NC res	Resized to 160 pixels	$n \in \{50, 100, 250, 500, 1000\}$
FH	sigma=1 minSize=20	$k \in \{10, 25, 50, 100, 200, 250, 300, 400, 500, 1e3, 5e3, 25e4, 5e5, 1e6\}$
EAMS	hs=8, hr=8	$minSize \in \{10, 25, 50, 100, 250, 1000, 2500\}$
QS	ratio=0.5 kernelSize=2	$maxDist \in \{5, 10, 15, 20, 25, 37, 50\}$
WS	-	$n \in \{50, 100, 250, 500, 1000, 2500\}$
ERS	-	$n \in \{50, 100, 250, 500, 1000, 2000\}$
oriSLIC	compactness=10	$n \in \{50, 100, 250, 500, 1000, 2000\}$
vlSLIC	regularizer=1000 minRegionSize=25	$regionSize \in \{10, 15, 20, 25, 37, 50, 100, 250\}$
SEEDS	-	$n \in \{50, 100, 250, 500, 1000, 1500, 5000\}$
VEK	-	$n \in \{50, 100, 250, 500, 1000, 2500\}, patchSize = 0.8 \cdot \sqrt{(h \cdot w)/n}$
VEK CI	lamda=50	$n \in \{50, 100, 250, 500, 1000, 2500\}, patchSize = 0.8 \cdot \sqrt{(h \cdot w)/n}$
BOX	-	$n \in \{50, 100, 250, 500, 1000, 2000\}$

Table 3.3.: Parameters that are different from defaults.

3.3. Runtime

The presentation of the compared superpixel algorithms in section 2.3.2 already delineated the large differences of the computational efforts necessary to create the segmentations. For the presented algorithms, it ranges from a few milliseconds to several minutes for a single image. In particular, for the usage of superpixels as a preprocessing step in real world applications, the runtime of the algorithm can be an important selection criterion. To evaluate the computational effort of an algorithm, we measure the average runtime for segmenting an image of the 200 BSDS500 test images of size 481x321 pixels on an Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz with 16 GB RAM.



t in sec	$n \approx 50$	$n \approx 100$	$n \approx 500$	$n \approx 1000$
WS	0.0085	0.0087	0.0088	0.0092
FH	0.08	0.08	0.08	0.08
oriSLIC	0.09	0.09	0.10	0.10
SEEDS	0.10	0.15	0.20	0.16
vlSLIC	0.25	0.31	0.40	0.40
ERS	1.0	1.0	1.2	1.3
EAMS	2.0	2.0	2.0	2.0
QS	5.2	3.0	1.5	1.4
VEK	3.2	3.0	2.5	2.5
VEK ci	5.3	5.2	4.5	4.1
NC res.	4.0	7.9	140.0	653.9

Fig. 3.6.: *(left)* Runtime of segmentation algorithms (log-scale). BOX has zero runtime and is thus not visible. *(right)* Runtime in seconds for different number of superpixels.

Fig. 3.6 shows the results of the runtime measurements. The logarithmic scale for the time axis induces the large differences - the runtime values cover 5 orders of magnitudes. **WS** is by far the fastest algorithm and creates a segmentation of 1000 superpixels in about 9 ms, which is more than 70,000 times faster than the slowest algorithm **NC res**. However, one has to keep in mind that the runtime of WS excludes the time for creation of the seeds. This is due to two reasons: 1. I made no special efforts to speed up the detection of the local gradient minima. 2. The seeds may be known in advance, e.g. due to the application (e.g. during tracking) or due to a fixed initialization as it will be used later in section 4.4. The local minima detection used in WS takes about 10 to 40 ms per image depending on the number of seeds and the image size.

FH, **SEEDS**, **oriSLIC** and **vlSLIC** run in fractions of a second. The runtime difference between the two SLIC implementations is supposed to be caused by the simplifications made in **oriSLIC** compared to the algorithmic descriptor in the paper [Achanta et al., 2012]. **vlSLIC** follows this description more closely (cf. section 2.3.2.7).

The runtime of most algorithms depends on the number of superpixels. Algorithms that iteratively subdivide image regions tend to have increasing runtime (e.g. **NC res**), while algorithms that merge image regions can have decreasing runtime (e.g. **QS**). However, the involved pre- and postprocessing step in the used implementations (e.g. handling not connected segments) can create computational efforts that superpose and hide the asymptotic runtime behaviour of the core segmentation approach itself. The general tendency is that more superpixels create more computational efforts.

In [Neubert and Protzel, 2012], we made similar runtime measures for some of the algorithms using an Intel Core 2 Quad Q9400 CPU @ 2.66 GHz with 4 GB RAM. The increased computational power compared to this earlier evaluation caused on average a speedup of about factor two.

Result 3.2 *The runtimes of the compared superpixel segmentation algorithms cover 5 orders of magnitude. Watershed segmentation is the fastest algorithm. It is at least ten times faster than any other algorithm from the compared approaches.*

3.4. Segmentation quality

There is a multitude of requirements on superpixel segmentation algorithms. Most importantly, we want superpixels to correspond to object boundaries - we want them to not overflow object boundaries and to correspond to objects and their parts as much as possible. Intuitively, this can be measured by comparing the superpixel segmentations to ground truth figure-ground segmentations. Therefore, there is a set of commonly used evaluation criteria. Most frequently these are boundary recall, undersegmentation error and maximum achievable segmentation accuracy. Boundary recall measures how well object boundaries are represented by superpixel boundaries. This is an established measurement to evaluate segmentation algorithms as well as figure-ground segmentations. However, using boundary recall alone favours segments with long boundaries. E.g. think of a segment with an anfractuous boundary that assigns each image pixel as boundary,

this single segment would achieve perfect boundary recall. Thus, in figure-ground segmentation, boundary recall is used together with boundary precision to take care of the length of boundaries. However, since for superpixel segmentation a low precision is inherent, undersegmentation error is more appropriate. Undersegmentation error evaluates the area of superpixels (in contrast to the boundary) and penalizes those that flow over figure-ground segment areas. Figure 3.7 illustrates this process. These two metrics are supplemented by the maximum achievable segmentation accuracy measure that evaluates the *best* possible combination of superpixels to reconstruct a ground-truth object. Since there exist various implementations of these error metrics, details and equations are explained in sections 3.4.1 to 3.4.3. [3]

3.4.1. Boundary recall

Boundary recall is an established measure for figure-ground segmentations, e.g. used in [Martin et al., 2004]. I use our definition from [Neubert and Protzel, 2012]: Boundary recall is the fraction of ground truth edges that fall within a certain distance d of at least one superpixel boundary. We use $d = 2$. Given a ground truth boundary image G and the algorithms boundary image B, the computation of boundary recall is straight forward:

1. **True Positives (TP)** Number of boundary pixels in G for which a boundary pixel in B exists within a range of d.
2. **False Negatives (FN)** Number of boundary pixels in G for which there is no boundary pixel in B within a range of d.
3. **Boundary Recall** $R = \frac{TP}{TP+FN}$

Multiple ground truth boundary images are combined using the logical OR operation. Thus, the resulting boundary map answers for each pixel the question: Is there a ground truth segmentation with a boundary at this pixel?

3.4.2. Undersegmentation error

Again, I follow our definition from [Neubert and Protzel, 2012]: Undersegmentation error compares segment areas to measure to what extent superpixels flood over the ground truth segment borders. A ground truth segment divides a superpixel P into an *in* and an *out* part. This is illustrated in Fig. 3.7. There exist various implementations of undersegmentation error metrics. [Levinshtein et al., 2009] summarize for each segment S over the out-parts of all superpixels P that overlap the segment:

$$UndersegmentationError_{TP} = \sum_{S \in GT} \frac{\sum_{P:P \cap S \neq \emptyset} |P_{out}|}{|S|} \tag{3.1}$$

[3] Parts of sections 3.4.1, 3.4.2 and 3.4.4 including Fig. 3.7 are published in very similar form in [Neubert and Protzel, 2012].

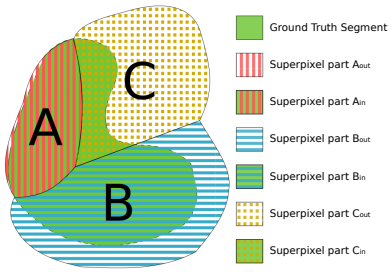


Fig. 3.7.: Illustration of undersegmentation error. A ground truth segment (green) is covered by three superpixels (A,B,C), that can flood over the ground truth segment border.

For the example in Fig. 3.7, this is: $\frac{|A_{out}|+|B_{out}|+|C_{out}|}{|S|}$. However, there is a serious penalty for large superpixels that have only a small overlap with the ground truth segment. Thus, Achanta et al. [2012] use a similar model, but only superpixels with an overlap with the segment of at least 5% of the superpixel size are regarded. To overcome this free parameter, we propose a new formulation of the oversegmentation error in [Neubert and Protzel, 2012]. There, we define the remaining error as the smaller error introduced by either appending the *out*-part to the segment or by omitting the *in*-part of the superpixel. Being N the total number of pixels, this results in:

$$UndersegmentationError = \frac{1}{N}\left[\sum_{S \in GT}\left(\sum_{P:P \cap S \neq \emptyset} min(P_{in}, P_{out})\right)\right] \tag{3.2}$$

The inner sum is the error introduced by this specific combination of ground truth segment and superpixel. For the example in Fig. 3.7 this is: $\frac{|A_{out}|+|B_{out}|+|C_{in}|}{|S|}$.

Contribution 3.3 *This section presented a parameter free formulation of the undersegmentation error.*

3.4.3. Maximum achievable segmentation accuracy

A common application for superpixels is to combine them to objects in higher level processing steps. In typical one-directional processing pipelines, the hard segmentation decisions are irreversible. Therefore, it is an interesting question to ask: How good is the *best* possible reconstruction of image objects using a given set of superpixels as units? The maximum achievable segmentation accuracy is such an upper bound measure [Nowozin and Lampert, 2010]. It measures a similar property as undersegmentation error but since both are repeatedly used metrics, I include both in this evaluation. Following Liu et al. [2011], each superpixel is assigned to the ground truth segment with the largest overlap and then the fraction of correctly labelled pixels is counted. Given a ground truth segmentation labelling G and a superpixel segmentation S, maximum achievable

segmentation accuracy is computed as follows:

$$MASA = \frac{\sum_k \max_i |S_k \cap G_i|}{\sum_i |G_i|} \tag{3.3}$$

3.4.4. Results on segmentation quality

The evaluation and results shown in Fig. 3.8 and 3.9 extend the evaluation presented in our previous work [Neubert and Protzel, 2012]. The performance of the regular grid **BOX** segmentation poses a baseline for each performance measure. In all metrics, the benefits of superpixels become obvious.

The properties of the resulting superpixel segmentations strongly depend on the algorithm. **ERS** produces similar sized superpixels, that carefully follow image gradients. Increasing the number of segments results in a refinement of the prior segmentation (i.e. only a few superpixel borders get lost, but some superpixels are divided). Further, **ERS** allows one to catch an exact number of segments.

Superpixel borders of **FH** strictly follow image gradients. The resulting segments vary strongly in shape and size. Sometimes, long, thin segments occur along strong image gradients. This causes problems with very small segment numbers (e.g. 30) on the test image size (481x321). However, **FH** yields the best boundary recall for higher number of segments (≥ 500) but shows high undersegmentation error and low maximum achievable segmentation accuracy. This is supposed to be caused by the graph merging criterion that relies on the single maximum weight in the minimum spanning tree of a component. If this criterion fails to detect a boundary between two image parts, the resulting superpixel can excessively overflow objects boundaries. An interesting question is the influence of this effect on the stability of these segmentations evaluated in section 3.5.

EAMS segments also vary in size and shape, but are more regular than **FH** segments. They yield high boundary recall and low undersegmentation error for low numbers of

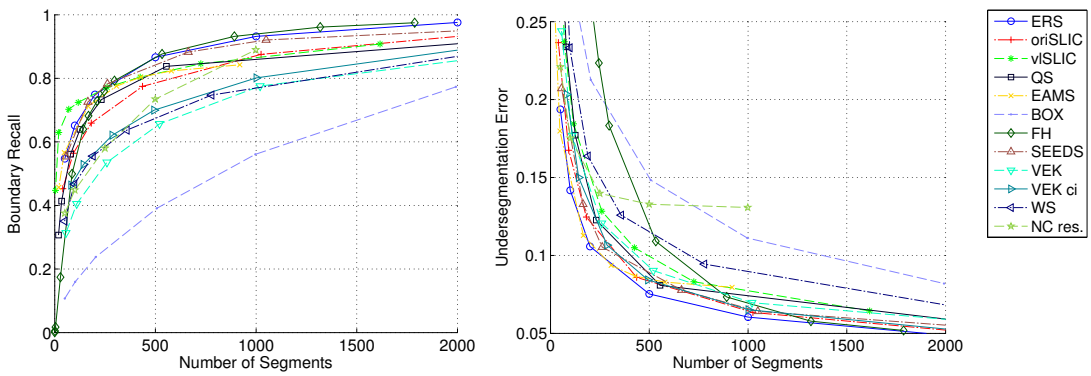


Fig. 3.8.: Results of segmentation algorithms on the proposed benchmark. For boundary recall higher is better, for oversegmentation error, lower is better.

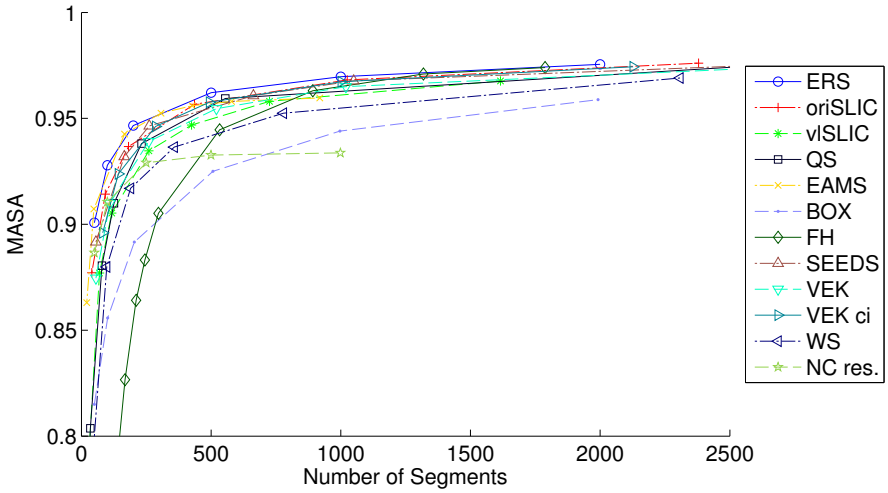


Fig. 3.9.: Results on maximum achievable segmentation accuracy. Higher is better.

superpixels. Supplemented by the high maximum achievable segmentation accuracy, **EAMS** is a good choice in particular for low segment numbers. Further, increasing the number of segments provides a good refinement of the prior segmentation.

NC res does not give such refinement, but the superpixels are regularly sized, shaped and distributed. The comparability of the evaluation of Normalized Cuts is limited since we resize the image before we compute the segmentation due to computational reasons. The negative effects of the rescaling become more severe with an increasing number of superpixels.

QS produces more irregular shapes and there occur small artifact segments. The regularity of segment borders of **oriSLIC** and **vlSLIC** strongly depend on the compactness parameter. Tuning this parameter is crucial. Moreover both implementations handle the compactness parameter differently. Values that create comparable smooth results for large numbers of segments produce much smoother borders at **oriSLIC** for small segment numbers. Nevertheless, the compactness parameter enables adjusting the regularity of the superpixels at the cost of loosing some object borders.

WS is by far the fastest algorithm, at the cost of inferior performance in the quality measures. It can neither compete at boundary recall nor undersegmentation error with the best performing approaches. In particular, the undersegmentation error and the maximum achievable segment accuracy show that the used Watershed implementation provides a less accurate reconstruction of the ground truth segments even for high numbers of segments.

Result 3.3 *Boundary recall and undersegmentation error show different properties of algorithms. The ordering of the performance of algorithms on maximum achievable segmentation accuracy is similar to that on undersegmentation error since they measure similar undersegmentation properties.*

Result 3.4 *Superpixel algorithms provide better segmentation quality than the uniform grid segmentation (BOX). Most superpixel algorithms are optimized for these quality measures. ERS, SLIC, EAMS, QS and SEEDS provide good trade-offs between boundary recall and undersegmentation. FH is optimized for boundary recall. Using NC with resized images is not recommended.*

3.5. The stability of superpixel segmentations

Section 3.1 discussed the importance of the segmentation quality. While runtime and quality are established criteria with ready to use measures, evaluating the stability of segmentations requires novel approaches. In [Neubert and Protzel, 2012] we were the first to present an evaluation measure for the robustness of superpixel segmentation, in particular towards affine image transformation. It bases on the comparison of segmentations of images and these images subject to affine transformation like rotation or scaling. This enables us to evaluate how sensitively an algorithm reacts to a well defined change in the image. In this thesis, I additionally evaluate to what extent the segmentations are affected by different levels of noise in the input images.

To overcome the limitations of the artificial image modifications, we proposed to use datasets with ground-truth optical flow in [Neubert and Protzel, 2013]. These datasets provide more complex image changes and the necessary information to evaluate the segmentation stability: If we know the motion of each pixel during an image sequence, we can use this information to compare the accordance of segmentations across the sequence. The ground truth optical flow information can also be used to identify object boundaries without the need for manual annotations, this will be topic of section 3.6. This section explains the developed metrics used to evaluate the stability criteria and presents comparison results for the above set of superpixel algorithms.[4]

3.5.1. Benchmarking robustness to affine transformations

This section presents an approach to evaluate the following question: To what extent does the modification of the input image affect the resulting segmentation? Here, the modifications are affine image transformations: rotation, scaling, shearing and shift. The goal is to identify superpixel segmentation algorithms, whose resulting changes of the segmentation correspond to the changes of the input image. For example, if the input is shifted by Δx pixels to the left, we want the superpixel boundaries to be shifted by Δx pixels to the left as well.

To evaluate the robustness of an algorithm A against an affine image transformation T, we compute two superpixel boundary images using this algorithm and compare them. The first boundary image $A(I)$ is computed on the origin image directly. For the second

[4]Parts of section 3.5.1 including Fig. 3.10 are published in very similar form in [Neubert and Protzel, 2012]. Parts of section 3.5.3 including Fig. 3.11 are published in very similar form in [Neubert and Protzel, 2013].

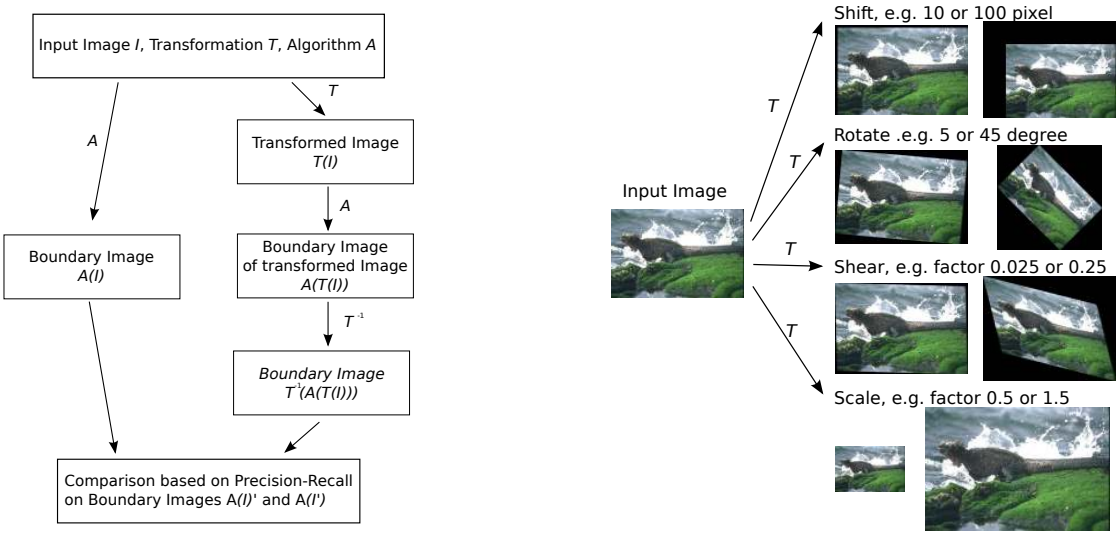


Fig. 3.10.: Image transformations. *(left)* To evaluate the robustness to an image transformation T, we compute two boundary images, $A(I)$ and $T^{-1}(A(T(I)))$, and compare. For more details, see text. *(right)* Illustration of example transformations.

boundary image, the image is first transformed by T to $T(I)$ using inverse mapping, followed by the computation of the boundary image $A(T(I))$. Back-transformation yields $T^{-1}(A(T(I)))$ which is compared to the boundary map $A(I)$ obtained from the segmentation of the origin image. The transformation scheme is illustrated in Fig. 3.10 together with some example transformations. The evaluation metric for comparing the two boundary maps is precision-recall represented by the F-score. Computation of TP, FN and Recall is done according to their computation for boundary recall in section 3.4.1, FP and Precision are computed as follows (d is set to 2 again):

1. **False Positives (FP)** Number of boundary pixels in B for which there is no boundary pixel in G within a range of d
2. **Boundary Precision** $P = \frac{TP}{TP+FP}$
3. **F-score** $F = 2 \cdot \frac{P \cdot R}{P+R}$

Due to their non rectangular shape, a black border can appear on some transformed images $T(I)$. To minimize their effects on the segmentation, all origin images are padded with a black border before the transformations. We evaluate the performance using the images of the BSDS500 *test* dataset.

Contribution 3.4 *This section presented an approach to evaluate the robustness of superpixel segmentations towards affine image transformations. A very similar formulation will be used for evaluation of robustness towards image noise.*

3.5.2. Robustness to noise

To evaluate the robustness of segmentation against image noise, the images of the BSDS500 *test* dataset are spoiled with artificial noise of different intensities and the influence on the resulting segmentations is measured. Two types of noise are evaluated. First, a Gaussian noise term with standard deviation s is added independently to each pixel:

$$n \sim \mathcal{N}(0, s^2) \tag{3.4}$$

$$I'_{i,j} = I_{i,j} + n \sim \mathcal{N}(I_{i,j}, s^2) \tag{3.5}$$

This corresponds to noise induced during image acquisition, e.g. sensor noise caused by poor illumination.

Second, Salt & Pepper noise is induced. Therefore, a certain rate of image pixels is switched to either black or white. Such noise typically occurs due to transmission errors. These two noise models are somehow contrary: the Gaussian noise affects all pixels to a small amount, the Salt & Pepper noise changes a few pixels very dramatically. The evaluation methodology is similar to that for the robustness against affine image transformation: the original image and the image spoiled with noise are segmented and compared based on maximum F score on the boundary images. Since there is no change of pixel coordinates (there is zero optical flow) in the noise experiments, the back transformation T^{-1} can be omitted.

3.5.3. A novel metric for stability based on ground truth optical flow

The evaluation based on affine image transformations and image noise both use synthetic image modifications. This allows one to evaluate the influence of a particular and well defined influence on the segmentation stability. However, in practical applications, the influences will appear combined and together with further effects like non affine image transformations (e.g. due to viewpoint changes, occlusions and dynamic objects), motion blur, changing illumination and others. To evaluate the stability of superpixel segmentations under such combined influences, we propose to use datasets that provide such changes together with ground-truth optical flow in [Neubert and Protzel, 2013].

From the datasets with optical flow information presented in section 3.2.2, the synthetic Sintel dataset provides dense flow fields. The real world KITTI dataset does not provide flow information for all pixels (cf. Fig. 3.4). This prevents the transformation of sparse images like boundary maps to evaluate their stability as it is done in case of affine image transformations and noise. Therefore, we proposed the following region based metric in [Neubert and Protzel, 2013].

The key idea is to segment two images showing the same scene before and after some changes (e.g. dynamic objects, camera motion, illumination changes), then use ground truth optical flow data to transform the segmentation of the first image into the view of the second image to make them comparable, and evaluate how good one segmentation

can be reconstructed from the other. We propose to use the following procedure to measure the stability of a segmentation between two images I_1, I_2:

1. Segment both images, resulting in label images L_1, L_2. In a label image, all pixels belonging to the same segment have the same pixel value (see Fig. 3.3 for examples).
2. Apply the given ground truth optical flow on the first segmentation L_1 to transform it into the second image, resulting in L_1^F. In other words, we transform the labels of segmentation L_1 to the pixels of image I_2 (similar to the transformation in Fig. 3.4).
3. Use the undersegmentation error to evaluate how well segmentation L_1^F can be reconstructed by segments of segmentation L_2 and vice versa. In particular, we do not expect to have the exact same label at a pixel in L_2 and L_1^F, but if two pixels are in the same segment in L_2, we want them to be in the same segment in L_1^F, too.

Computation of undersegmentation error has been described in Sec. 3.4.2. To compare two segmentations L_1^F and L_2, and being N the total number of pixels, we define the motion undersegmentation error (MUSE) to be computed as follows:

$$MUSE = \frac{1}{N}\left[\sum_{a \in L_1^F}\left(\sum_{b \in L_2 : a \cap b \neq \emptyset} min(b_{in}, b_{out})\right)\right] \tag{3.6}$$

Each segment a of segmentation L_1^F is reconstructed with segments b of L_2 that overlap with a. MUSE accumulates the error that is introduced by b when reconstructing a either when b is included in the reconstruction or not. If b is included, then the introduced error is the number of pixels of b that are outside a (defined as b_{out}). Otherwise, if we do not include b, there is a gap in the reconstruction of a and the error is the number of pixels in this gap (b_{in}, the number of pixels that are in $a \cap b$). Pixels without valid flow information are ignored. Since this is not a symmetric metric (the error diverges depending on whether comparing L_1^F to L_2 or vice versa), we compute the average of both cases. Fig. 3.11 visualizes for an example image pair, how often each pixel is counted in equation 3.6. The evaluation results of the set of compared algorithms can be seen in Fig. 3.14.

Contribution 3.5 *The motion undersegmentation error is a novel metric to evaluate the stability of superpixel segmentation algorithms based on ground truth optical flow.*

3.5.4. Results on segmentation stability

Figure 3.12 shows results of all algorithms for the affine transformations shifting, rotation, scaling and shearing. In these experiments the algorithms' parameters are chosen to result in about 250 segments per image.

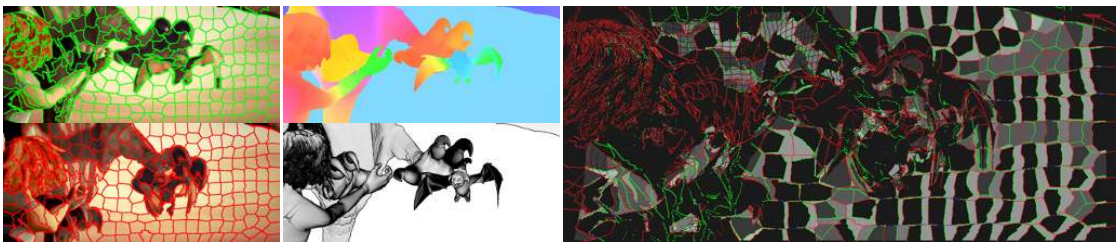


Fig. 3.11.: The two left images show segmentations of subsequent images of a Sintel sequence with a high amount of background motion (using *vlSlic*). The flow field is visualized in the upper middle with its gradient image beneath. Notice the high (dark) motion gradients inside objects. The image on the right is a visualization of the motion undersegmentation error (MUSE) (including segment boundaries): the gray level shows how often the pixel is counted in equation 3.6. The grid effect on homogeneous image regions is typical for algorithms with strong compactness constraints and can hardly be completely avoided. However, algorithms diverge significantly in the amount of border variation in homogeneous areas.

FH and **EAMS** are invariant to image shifts, **QS** is at least robust to shifts. The other algorithms perform inferior, but still reasonable. **oriSLIC** and **VEK** show the most severe problems. It is noticeable that with an increasing amount of shift, most algorithms' performance quickly drops until it reaches a rather stable level for the remaining larger shifts. This is coherent with the fact that the set of superpixel boundary pixels contains salient image edges but also additional boundaries, e.g. induced by compactness constraints. The set of salient edge pixels is to a large part recovered in the modified images, while the additional boundaries are not. Moreover, a fraction of the segment boundaries will always be counted due to chance. The performance of *BOX* gives an idea of this lower bound. However, chance will increase with more irregular and thus longer boundaries. The peaks of **vlSLIC** appear at shifts that correspond to a multiple of the step width of the superpixel initialization.

A similar effect can be seen for rotation where the performance of most algorithms is somehow periodic - rotations of 90 or 180 degrees are easier than arbitrary rotations. While we used only integer shifts, the affine image transformations rotation, scale and shear also include small interpolation effects that occur during inverse mapping. The ordering of the performance of the compared algorithms is similar for all types of transformations, except for scaling where some algorithms showed to be more sensitive (e.g. **WS**) and other less (e.g. **oriSLIC**). Except for scaling, the large difference of **BOX** to even the most sensitive superpixel algorithms illustrates their potential benefit.

Result 3.5 *Algorithms with compactness constraints and/or grid initialization are more sensitive to affine image transformations except for scaling.*

The evaluation of the robustness towards image noise is shown in Fig. 3.13. The overall performance of superpixel segmentations is as expected, increasing the amount of noise decreases the boundary repeatability. In contrast to its previously shown robustness,

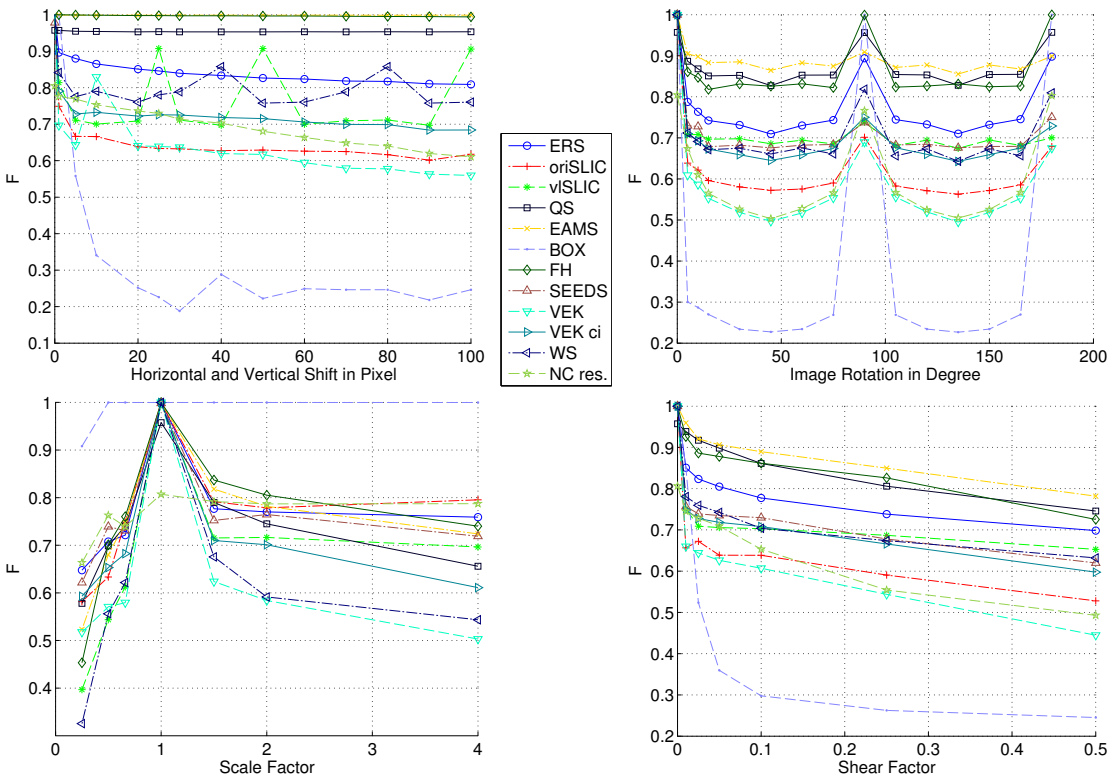


Fig. 3.12.: Robustness of algorithms towards shifting, rotation, scaling and shearing. Higher is better. Note: For shift, the EAMS curve is hidden behind the FH curve.

FH reveals a sensitivity for Salt & Pepper noise. There, the algorithm performs worse than most other algorithms in this comparison. This is supposed to be due to the graph based segment merging that relies on a single weight of a minimum spanning tree. If this weight is changed significantly, the segmentation can become quite different. Changing single parts of the image significantly is exactly the effect of Salt & Pepper noise. There is a parameter for the amount of image smoothing in the used implementation. However, as discussed in section 3.2.3, changing additional parameters of algorithms is beyond the scope (respectively capabilities) of this here presented comparison. While enforcing the smoothing could help for this issue, it might have a negative influence on other criteria, e.g. the segmentation quality. Moreover, for the practically more important Gaussian noise, **FH** performs stable again.

oriSLIC is stable against Gaussian noise with small standard deviation, but sensitive to larger deviations. Again, the two SLIC implementations differ and **vlSLIC** performs superior. **EAMS** shows to be stable again, except for high amounts of Gaussian noise. **VEC**, **VEC ci** and **WS** are susceptible to one or both types of noise.

From this evaluation of individual, well defined image modifications, let us proceed with a more realistic scenario. Fig. 3.14 shows the sensitivity to complex combinations of various influences including camera motion, dynamic objects, illumination changes, noise, motion blur and others using the proposed motion undersegmentation error (MUSE). We independently evaluate this criteria on the two optical flow datasets,

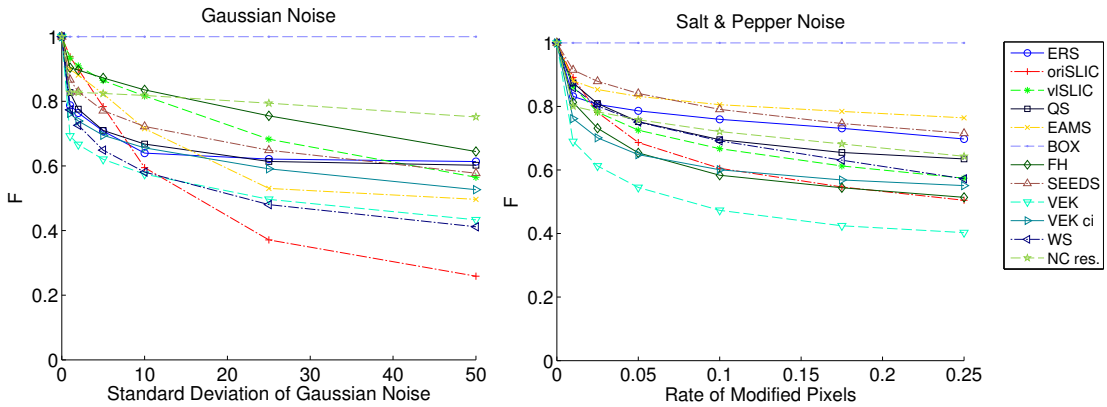


Fig. 3.13.: Robustness to Gaussian and Salt & Pepper noise. Higher is better.

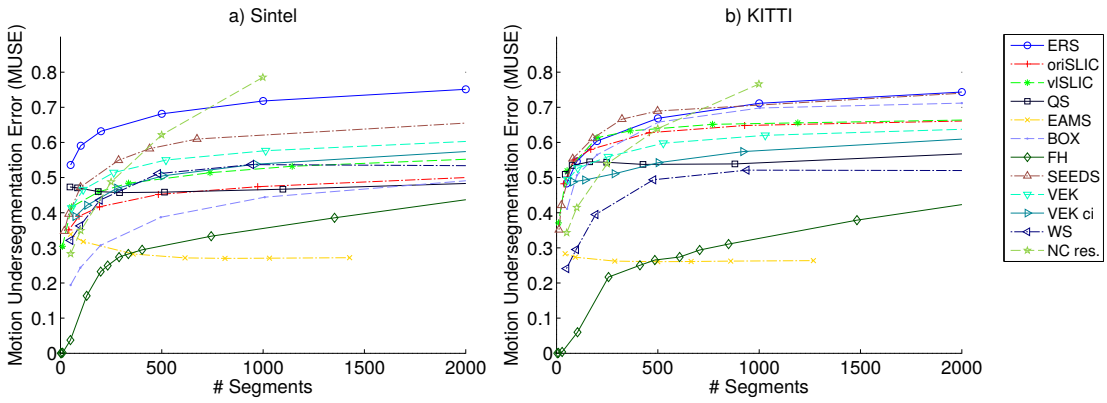


Fig. 3.14.: Results of the MUSE metric on several superpixel segmentation algorithms for both datasets. Lower is better.

Sintel and KITTI, and average over all images of each dataset. To evaluate the algorithms performance on varying numbers of segments, we run all algorithms with different parameter sets according to the description in section 3.2.3. In terms of the motion undersegmentation error, we can not expect an oversegmentation algorithm to solely create stable segments, since they also split homogeneous image areas without image gradient support. The effects can be seen at the grid-like areas on the visualization of the MUSE metric in Fig. 3.11. For the ground truth optical flow datasets, the pixel motions and other image changes are quite complex and we can not isolate individual influences. However, we can use the **BOX** segmentation as indicator for the difficulty of the dataset. **BOX** showed to be sensitive for affine image transformations. Its reasonable performance on Sintel is due to the fact that in the film sequences, there are moving foreground objects and little overall motion in the image (cf. Fig.3.5). Moreover, in many scenes, the background is rather homogeneous and the segment boundaries created there are mainly due to regulation of the segment's size or distribution. Algo-

rithms that strongly connect to image gradients and are not limited by regulation of the segment's size or distribution (like **FH** or **EAMS**) can capture the boundaries of the moving objects and create large stable segments in the homogeneous background resulting in low MUSE values in Fig.3.14.

The worse performance of **BOX** at the KITTI dataset also demonstrates the higher amount of systematic camera motion in KITTI images since the camera is mounted on a driving car. This results in higher overall motion in the image (cf. Fig.3.4) and a few additionally moving foreground objects.

The performance on the ground truth optical flow datasets is in accordance with the previous evaluations on robustness against affine image transformations and noise and additionally reveals further insights. Most importantly, it is an evaluation on combined influences as they happen in real world applications. Each curve on the MUSE metric evaluates different numbers of segments. It is apparent that MUSE values increase with a growing number of segments. This is contrary to the characteristic of the undersegmentation error when comparing a superpixel segmentation to a figure-ground segmentation (cf. section 3.4.2). The intuitive reason is that when comparing two superpixel segmentations, an increasing number of segments in the segment set used for reconstruction is connected to an increased number of segments in the set that is reconstructed. Thus, we do have smaller building blocks, but also want to build more filigree elements. The mean shift algorithms **EAMS** and **QS** can compensate this increase. In particular for smaller numbers of segments, the motion undersegmentation errors of **FH** are considerably lower than for the other algorithms.

Result 3.6 *FH and EAMS provide the the most stable segmentations. For small numbers of segments, FH superpixels are considerably more stable than those of all compared algorithms.*

This evaluation also reveals a sensitivity of **ERS** towards these combined complex image changes that did not appear for affine image transformations and noise. **ERS** incorporates a compactness constraint on the size of the segments. However, while the segment size is stable, the segment borders flicker in homogeneous image areas subject to illumination changes and thus cause high motion undersegmentation error.

The Sintel dataset provides sequences of 20 to 50 images. The overall amount of optical flow increases when comparing images with larger distances in the sequence. For the compared algorithms, this results in higher MUSE values while preserving the characteristics and ordering of the resulting curves. An example result can be seen in Fig. 3.18.

3.6. Compactness properties of superpixel segmentations

The so far shown example segmentations revealed a broad variety of characteristics of superpixel segmentations. Some algorithms produce uniformly distributed and shaped segments of similar size and for others the geometric properties of the segments vary

strongly. Compactness measures aim to quantify the location of a superpixel segmentation on this spectrum:

> *A segmentation has a high compactness, if its segments are uniformly distributed and have uniform size and shape.*

Some algorithms provide compactness constraints that allow the user to adjust the amount of regularization on segment size and/or shape (e.g. SLIC and ERS).

A compact segmentation algorithm is intended to vary from the perfectly regular shape (e.g. a circle or a square) only if there is considerable support from the image data. In terms of strongly varying boundaries, non-compact segmentation algorithms are related to overfitting in machine learning [Schick et al.]. An overfitted learning algorithm lost generalisation capability since too much data was used to describe the essential information. In case of irregularly shaped superpixels, too many boundary pixels are used to describe the essential shape of the object.

While compact and non-compact segmentations are visually quite different, it is not straight forward to develop a measure that captures the potential benefit of compactness for practical applications. Compactness can help to prevent segmentations to excessively overflow object boundaries. In accordance, compact segmentation algorithms tend to provide smaller undersegmentation error. On the other hand, compactness constraints create additional boundaries that are not solely caused by visual cues. Dependent on the application, compactness may be helpful or not. For example, as basis for object detection, compactness of a superpixel segmentation may not be required, while a superpixel based stereo or 3d reconstruction algorithm may benefit from a prevention of arbitrary large segments. Moreover, the majority of compact superpixel segmentations provide a regular neighbourhood structure, e.g. the usual 4- and 8-neighbourhoods. Subsequent processing steps may benefit from such a regular neighbourhood. Additionally, this simplifies the usage of algorithms that are designed to work on a pixel level, where the regular neighbourhood is obvious. For example, the SP-Grid algorithm, that will be presented in the second part of this thesis in section 7.4.2, requires compact segmentation algorithms with a regular neighbourhood - and will show to outperform available methods on place recognition in changing environments.

I will approach the evaluation of compactness in two steps. The first being a collection of measures of geometrical properties in section 3.6.1. Second, I will again use ground truth optical flow to evaluate the benefit of compactness more closely to real applications in section 3.6.2.

3.6.1. Compactness from geometric properties

As a straight forward measure, the amount of variation in superpixel segment size can be expressed as average standard deviation of the segment sizes in the test images. The results are shown in Fig. 3.15.

Schick et al. propose to use the isoperimetric quotient to measure the compactness of segmentations. The isoperimetric quotient is the relation between the area of a given

shape and a circle of the same perimeter as this shape. Given the area A_s and the perimeter L_s (the number of boundary pixels) of a shape, the circle with radius $r = \frac{L_1}{2\pi}$ has the same perimeter as the shape but a different area A_{epc}. The isoperimetric quotient IPQ_s is computed as follows:

$$IPQ_s = \frac{A_s}{A_{epc}} = \frac{4\pi A_s}{L_s^2} \tag{3.7}$$

Given a set of superpixels $s \in S$ for an image I, the weighted isoperimetric quotient $WIPQ$ of this segmentation is the average isoperimetric quotient weighted by the number of pixels in each segment:

$$WIPQ = \sum_{s \in S} IPQ_s \cdot \frac{|s|}{|I|} \tag{3.8}$$

This compactness measure incorporates two effects: eccentricity and boundary smoothness. Elongated superpixels that deviate from close to circular shapes and irregularly or wiggly boundaries cause low values. Regularly shaped segments (shapes close to a circle) with smooth boundaries cause short perimeters and thus high compactness values.

However, to some extent a wiggly boundary can be smoothed by simple post processing. To evaluate the regularity of the segment shapes without the influence of a wiggly boundary, I adapt the compactness measure of Schick et al. by evaluating the deviation from circular shape using the isodiametric quotient (in contrast to iso*peri*metric). This is the ratio of the area A_s of the segment and the area A_{edc} of a circle with the same diameter. The diameter d of a segment is defined as the maximum euclidean distance between two points on its boundary.

$$IDQ_s = \frac{A_s}{A_{edc}} = \frac{A_s}{\frac{1}{4}d^2\pi} \tag{3.9}$$

Analogue to Eq. 3.8, the weighted isodiametric quotient $WIDQ$ of this segmentation is the average isodiametric quotient weighted by the number of pixels in each segment:

$$WIDQ = \sum_{s \in S} IDQ_s \cdot \frac{|s|}{|I|} \tag{3.10}$$

In section 3.6.3, the compactness properties of segmentations are evaluated using the three presented measures: the standard deviation of the segment size, the smoothness of the boundary by the weighted isoperimetric quotient $WIPQ$ and the deviation of the shape from a circle by the weighted isodiametric quotient $WIDQ$.

Contribution 3.6 *This section proposed to use the standard deviation of the segment sizes and the weighted isodiametric quotient to evaluate geometric properties of superpixel segmentations.*

3.6.2. A novel metric measuring the accordance with motion discontinuities

The previous section presented three measures to evaluate the geometric properties of superpixels that yield the compactness of the segmentation. Beyond concrete properties like variation in size or shape, the potential benefit of compactness is rather abstract. Moreover, some properties are not that easy to quantify, e.g. what is a good distribution of segments in the image? One advantage of compactness is the limitation of negative effects due to missed object boundaries when reconstructing objects from superpixels as it is measured by the undersegmentation error (cf. section 3.4.2).

However, there might be other important boundaries in the image that are not marked in human figure ground segmentations, e.g. think of the individual components of objects or the individual moving parts of deformable objects. Separating these parts in different superpixels might be crucial for subsequent processing steps like 3d reconstruction. Separating image parts with little or even no evidence from image gradients is an important property of compact segmentations.

In [Neubert and Protzel, 2013], we propose to use datasets with ground truth optical flow to evaluate how good motion discontinuities are represented by the algorithms' segment boundaries. Motion discontinuities indicate differently moving image parts, e.g. induced by a moving foreground object or a motion inside a non rigid image area. While the boundary of the former is likely to be marked in a human made manual ground truth segmentation, for the latter there are presumably no boundaries in a human made segmentation. However, dependent on the application, both may be relevant.

In the gradient image of an optical flow field, both types of boundaries are present.[5] Fig. 3.11 (bottom-mid) shows the gradient magnitude of the optical flow field between two subsequent images of the Sintel dataset. One can clearly see how high motion gradients (shown in dark colour) appear at boundaries of moving objects, supplemented by smoother gradients inside objects. Dependent on the application, it is important to have a segment boundary near areas with high motion gradients. The intuitive argument is that the high motion gradient indicates objects or object parts that can move separately and thus should probably be handled individually in the application. This formulation includes objects as well as object parts, independently from a semantic interpretation by humans. Following this argumentation, there should be segment borders near high motion gradients to potentially handle differently moving parts individually. Since not all of these motion gradients are also indicated by visual image gradients, the ability to nevertheless capture them is a practical example for the benefit of compactness of superpixel segmentations.

In [Neubert and Protzel, 2013] we propose the motion discontinuity error (MDE) to evaluate how well motion discontinuities are captured. While the capability of a segmentation algorithm to generate the boundaries of moving objects is intuitively covered by a figure-ground segmentation evaluation, this is not the case for smooth gradients inside objects. Moreover it is unclear, at which of the smooth gradients there should be a segment border. To avoid arbitrarily chosen thresholds to separate important high

[5] The remaining part this section 3.6.2 is published in very similar form in [Neubert and Protzel, 2013].

gradients from ignored low gradients, we propose to use the following error measure: Given F, a ground truth optical flow field from an image I to another image, B the boundary image of a segmentation of image I, and $D(B)$ the distance transform of B containing for each pixel the distance to the nearest segment boundary, we define the motion discontinuity error as follows:

$$MDE = \frac{1}{\sum_i \sum_j \|\nabla F(i,j)\|_2} \sum_i \sum_j \|\nabla F(i,j)\|_2 \cdot D(B(i,j)) \tag{3.11}$$

In one sentence, this is the Frobenius inner product of the optical flow gradient magnitude and the distance transform of the boundary image of the segmentation, divided by the sum of all gradients. A more intuitive formulation is to accumulate over all image pixels a penalty, which is the product of the magnitude of motion discontinuity at this pixel and its distance to the next segment border. Finally, the measure is normalized by the total amount of motion in the image.

We want to penalize if there is a motion discontinuity (a high gradient magnitude in the flow field) but no near segment border that can potentially separate the two differently moving image parts. However, this is a slightly optimistic measure, since we do not verify whether the nearest segment boundary really separates the two differently moving parts. Similar to measures like boundary recall, MDE favors segmentations with many boundary pixels (e.g. caused by strongly irregular segment borders) and should be used together with complementing measurements (e.g. undersegmentation error).

Contribution 3.7 *The motion discontinuity error is a novel measure to evaluate compactness properties of superpixel segmentations based on ground truth optical flow.*

3.6.3. Results on segmentation compactness

Fig. 3.15 shows the average standard deviations of the superpixel sizes in an image with a certain number of segments. For each algorithm, the average segment size is the same (the number of pixels in the image divided by the number of segments) but the amount of variance of the individual superpixel from this mean value is quite different. For the grid segmentation **BOX** there is almost no deviation, while the segment sizes of **FH** and EAMS vary a lot. **WS** creates segments of considerably different sizes as well, in particular for small numbers of segments.

The evaluation of the shapes based on the weighted isodiametric quotient, resembles the visual impression of the algorithm's properties. Algorithms like **SLIC** and **VEK** create regularly sized, compact segments. The shape of **FH** segments closely follows the image gradients without a shape prior resulting in arbitrarily shaped segments and low compactness values. Although the **EAMS** segments vary strongly in size, their shape is more compact than that of of **FH**. For all compactness measures, **NC res** suffers from the image resizing and the fact that the pixels assigned to a segment are not forced to be connected in the **NC** implementation as can be seen by the example segmentations in Fig. 2.13.

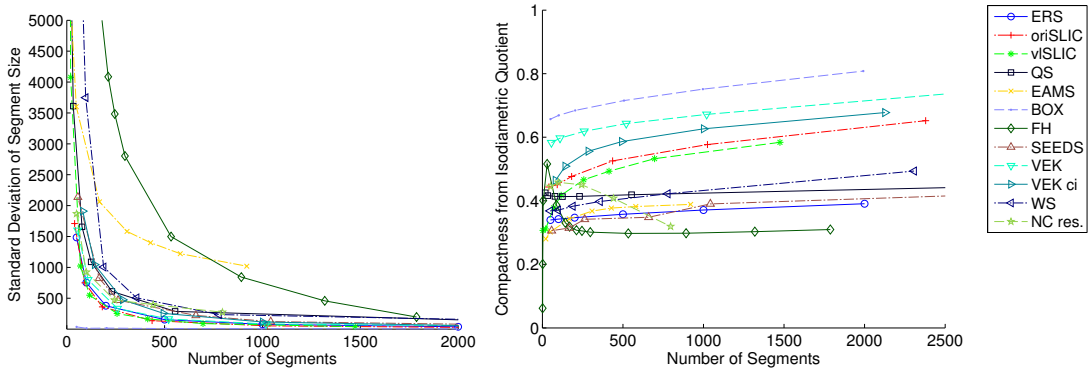


Fig. 3.15.: Standard deviation of segment size and compactness from weighted isodiametric quotient (regular shape).

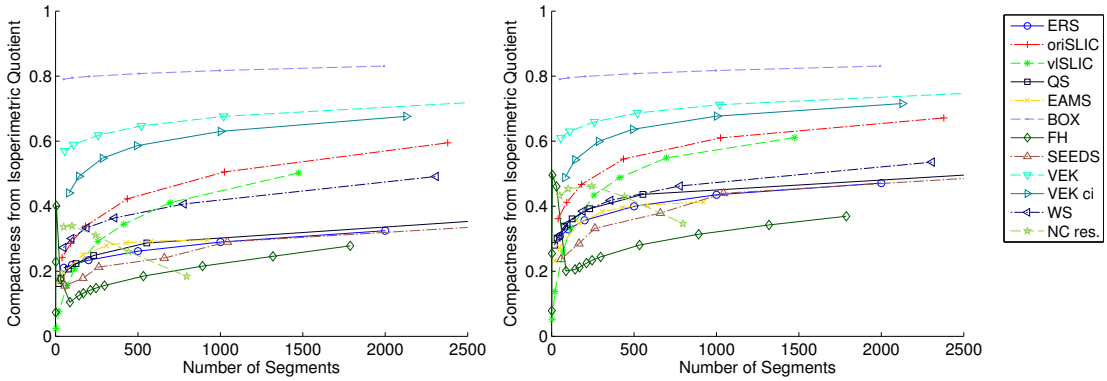


Fig. 3.16.: Weighted isoperimetric quotient (boundary smoothness). Left: original boundaries. Right: smoothed segment boundaries (morphological closing with radius 5).

The weighted isoperimetric quotient combines the effects of eccentricity and irregular, wiggly boundaries. Fig. 3.16 shows on the left side the weighted isoperimetric quotient as described above. The right figure shows the effect of additional smoothing of the segment boundaries. Smoothing is done by a morphological closing operation on the label image with a structure element with radius five. This smooths irregular segment boundaries at a small scale while preserving the overall eccentricity. For some algorithms, the isoperimetric quotient is sensitive to this smoothing since this measure mixes the influences of eccentricity and irregularity of the boundaries. The smoothed boundaries closer correspond to the results of the isodiametric quotient. The improvement from the original to the smoothed boundaries indicates the amount of small scale flicker in the segmentations. For example, the compactness of **ERS** and **QS** benefit more from the rescaling than the compactness of **WS**. This indicates that the high perimeter values of the former are to a higher rate due to small scale irregularities that are reduced by the smoothing. The low compactness of **WS** is mainly caused by the overall eccentricity.

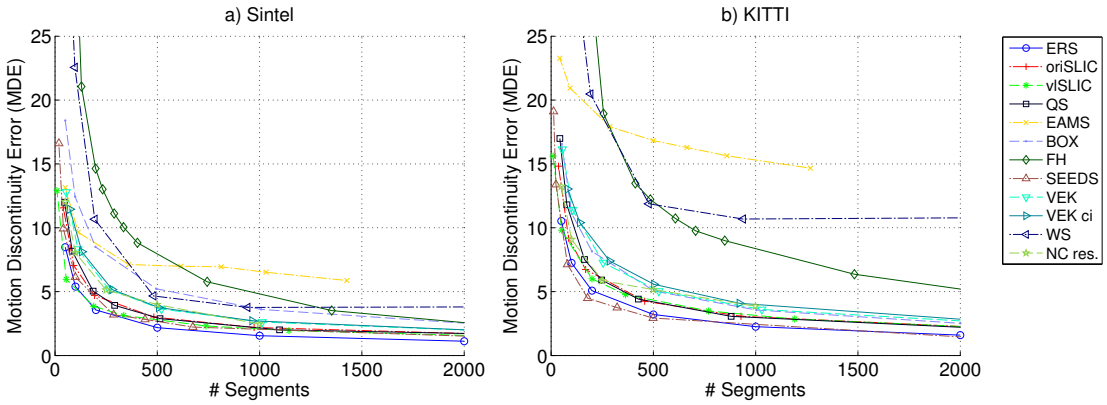


Fig. 3.17.: Results of the MDE metric on several superpixel segmentation algorithms for both datasets. Lower is better.

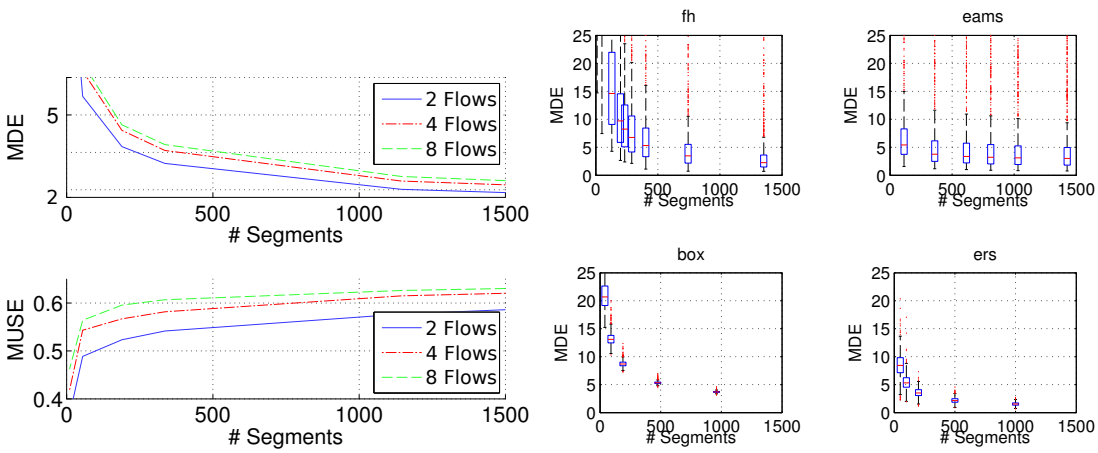


Fig. 3.18.: *(left)* Motion discontinuity error (MDE) and motion undersegmentation error (MUSE) of *vlSLIC* for concatenated motion fields of varying lengths. *(right)* Boxplots of motion discontinuity error on the Sintel dataset for some algorithms (whiskers at 1.5×IQR).

Result 3.7 *The results on standard deviation of segment size and the weighted isodiametric quotient are in accordance with the qualitative visual impression of superpixel segmentation compactness.*

Result 3.8 *The SLIC and VEK variants are examples for compact segmentation algorithms. FH, EAMS and WS show strongly varying sizes. SEEDS and ERS segments are well distributes but show strongly varying boundaries.*

At **motion discontinuity error**, superpixel algorithms with additional segment boundaries caused by compactness constraints perform superior since they decrease the average values of the distance transform of the boundary image. On the other hand, algorithms that exclusively rely on image gradients (like **FH** and **FH**) miss many motion discontinuities. The two SLIC implementations that already performed well on recov-

ering object-ground segmentations in section 3.4, also show good results on recovering motion discontinuities. The additional information about the statistics on MDE given by the boxplots in Fig. 3.18 show that the high mean error of **eams** is mainly due to constantly high outlier rate. The median error is much lower. Other algorithms show statistics similar to **ERS**: median error is smaller than the mean and the number of outliers decreases with larger numbers of segments.

To evaluate the dependency of the algorithm's performance on the amount of motion, Fig. 3.18 also shows exemplary results of one algorithm (vlSLIC) on concatenated flow fields (combined flow fields over multiple frames). All algorithms show similar behaviour, MDE uniformly increases with growing sequence lengths.

Result 3.9 *Compact superpixel segmentations better capture motion discontinuities. From the different compactness properties, a uniform distribution of the superpixel boundaries in the image (e.g. grid-like) is more important for MDE than regular shape or smooth boundary. For this metric EAMS, FH and WS perform inferior.*

3.7. Summary of the benchmark results

This section identified segmentation quality, runtime, stability and compactness as important properties of superpixel segmentation algorithms. While for runtime and segmentation quality there are straight forward and established measures, I presented novel evaluation methodologies for measuring stability and compactness. Particularly, I proposed metrics that exploit datasets with known ground truth optical flow to evaluate practical relevant complex combinations of influences. The implementations of these benchmarks are publicly available to foster the evaluation of new approaches (cf. section 1.3).

Based on these metrics, a set of existing algorithms has been compared. It turned out that most algorithms are optimized for quality. There, boundary recall and undersegmentation error measure different aspects of segmentation quality, one attends to the boundary of objects, the other to the area. Some algorithms are significantly better at one of them, e.g. FH ist optimized for boundary recall and may produce large errors for the reconstruction of the area of objects from the resulting segments. Algorithms like SLIC, EAMS, SEEDS, QS or ERS are well balanced on both quality aspects. However FH and EAMS turned out to be the most stable algorithms while WS is by far the fastest. Compactness of algorithms has been measured in terms of geometrical properties. Some algorithms like SLIC or ERS provide parameters to influence the compactness. However, evaluation of these additional parameters is beyond the scope of this benchmark.

More compact segmentation turned out to be less stable. The compactness constraints are a purely geometrically driven influence on the segment shape independent from visual image cues. In particular for affine image transformations like shift or rotation, the prior on the geometry and distribution of segments is supposed to negatively influence boundary recall. However, for non geometrical influences like image noise, the compactness can in contrast support stable segmentations. Further, the evaluation based on

the proposed motion discontinuity error showed that compactness can help to limit the negative influence of image boundaries that are not indicated by image gradients, e.g. inside non rigid objects.

The here presented comparison is somehow limited in terms of general insights on the segmentation approaches since it compares particular implementations and a particular set of parameters for each of them. However, it provides information on practical considerations that have to be made when selecting a particular algorithm for a task at hand. Moreover, it can be used to benchmark new algorithms - the topic of the following section.

Contribution 3.8 *An open source implementation of the benchmarks is available, cf. section 1.3 on open source software releases.*

In a nutshell: How to select the best algorithm for the task at hand?

The crucial question is: What are relevant criteria for the task? If very high *runtime* is important, then Seeded Watershed may be the only choice. If *compactness* is required as well, then the Compact Watershed algorithm, that will be presented in the following chapter, is preferable. In terms of compact superpixel with high *quality* and reasonable stability, the SLIC variants are a good choice. The following chapter will introduce PreemtiveSLIC, a faster version of SLIC that saves about 80 % of the runtime.

If compactness is not important, but boundary recall is essential, then FH is an established superpixel method that showed to be fast and stable in the benchmark. FH tends to provide small and very large segments - the latter may strongly overflow object boundaries. If this is a problem, QS and EAMS are other reasonable choices for well balanced *stable* algorithms.

Further, the taxonomy in table 2.1 may be helpful - and, of course, the multitude of curves shown in this chapter.

Divide and conquer.
Anonymous

4

Two Novel Superpixel Algorithms: Preemptive SLIC and Compact Watershed

The comparison of existing superpixel algorithms in the previous chapter revealed some trade-offs in the available algorithms. In this section, I will propose two novel segmentation algorithms that better balance these trade-offs. Preemptive SLIC is a five times faster approximation to SLIC. Compact Watershed incorporates compactness constraints in the very fast seeded watershed segmentation algorithm. In a first step, I will discuss the trade-offs, followed by algorithmic descriptions of the two novel algorithms together with extensive evaluation based on the benchmarks introduced in the previous section. Earlier versions of these novel algorithms and parts of the presented evaluation are also published in [Neubert and Protzel, 2014].

4.1. Trade-offs of superpixel segmentation algorithms

Chapter 2 presented the broad range of existing superpixel segmentation algorithms. Segmentation quality, runtime, stability and compactness have been identified as important criteria to evaluate superpixel algorithms in chapter 3. A main insight from this comparison is that each algorithm is a trade-off between these criteria. In theory, there may exist a perfect superpixel segmentation algorithm, that runs sufficiently fast and provides the perfect segments for the subsequent processing steps. However, in practice, we have to look for the existing algorithm that provides the best balanced trade-off for the task at hand.

The comparisons from the previous section can be used to identify significant trade-offs in the existing algorithms. The most intuitive is the trade-off between segmentation quality and runtime. Moreover, for some metrics, non compact oversegmentation algorithms perform better than algorithms with compactness constraints, e.g. boundary

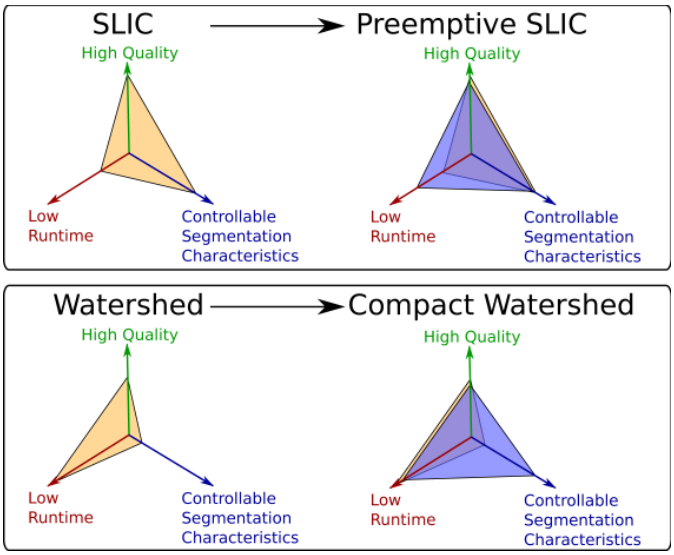


Fig. 4.1.: This sections proposes two new adaptations of existing algorithms that change the balance in trade-offs of superpixel segmentation algorithms. The three coloured axes visualize a set of typical trade-offs: segmentation quality measures, runtime and appropriate segmentation characteristics for subsequent processing steps (e.g. similar size). The application of the segmentation results may give a weighting to the different axes. But in general, a triangle with larger area indicates a better segmentation algorithm. The proposed Preemptive SLIC algorithm is a much faster version of SLIC [Achanta et al., 2012], running at frame-rates > 30 Hz on a standard desktop CPU. This comes at the cost of slightly worse segmentation quality. The proposed Compact Watershed algorithm is an extension of a seeded watershed segmentation, that conserves the high speed and gives the user control over the compactness of the segments. This figure is also published in [Neubert and Protzel, 2014].

recall. At boundary recall, neither regularly shaped segments nor a grid like distribution are helpful. For other metrics, e.g. undersegmentation error, constraints on the size and shape of the segments help to prevent segments from growing unbounded at homogeneous image regions. Such constraints also influence the stability of segmentations. The proposed novel algorithms are based on two interesting examples for trade-offs:

- SLIC shows good quality but at a rather low speed compared to Watershed
- Watershed shows worse quality and compactness properties, but performs at a very high speed

This section presents approaches to do both: to speed up the good-but-slow algorithm SLIC and to improve the worse-but-fast algorithm Watershed.

We were not the first laying hands on these well-know algorithms. For example, there exists a frame-rate GPU version of SLIC [Ren and Reid, 2011] and for sure there exist various implementations and distance metrics for watershed segmentation, see [Roerdink

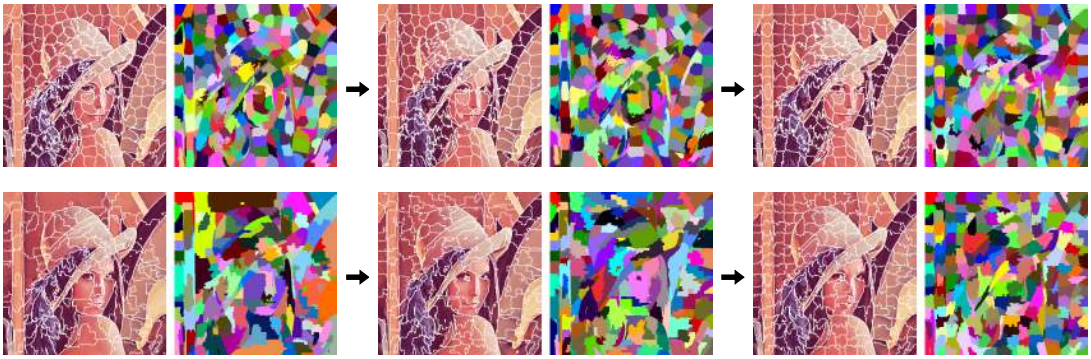


Fig. 4.2.: The top row shows the original SLIC (left) and the proposed Preemptive SLIC. The middle column shows the influence of the switch to the RGB colour space (pSLIC t=0) and the right column the influence of the preemptive termination (pSLIC t=0.25). The resulting segmentations are visually similar, however, the proposed approach is five times faster. The bottom row shows the original Watershed with seeds at local gradient minima (left) and the proposed Compact Watershed. The middle column shows the influence of the grid initialization (CWS c=0) and the right column the influence of the compactness constraint (CWS c=25). The resulting segmentations are visually similar to the above results of SLIC and pSLIC, however, the Compact Watershed is more than twice as fast as pSLIC.

and Meijster, 2000] for an overview. However, to the best of our knowledge, we were the first to present a frame-rate CPU version of SLIC and a ready-to use compact superpixel segmentation based on watersheds. Recently and after the publication of our Compact Watershed approach, Machairas et al. [2014] proposed a similar approach - a very interesting candidate for future inclusion in the benchmark of chapter 3.

Fig. 4.1 illustrates the contributions of this section. The two implementations of SLIC showed to provide high quality segmentations and the grid initialized segments exhibit good compactness properties. SLIC is a repeatedly used superpixel algorithm. However, the runtime of about 100 ms may be too high for consideration as a preprocessing step for time critical applications. We propose an adaption of SLIC, Preemptive SLIC, that preserves the high segmentation quality level of the original implementation and runs at framerates >30 Hz on a standard desktop CPU, e.g. dependent on the parameter setting it runs at about 40-50 frames per second on images of size (481×321). The runtime improvement is based on proposed changes to the algorithm that adjust the trade-off between runtime and quality in favour of a frame-rate acceleration. Section 4.3 evaluates the positive and negative effects on segmentation properties based on the previously presented benchmarks.

Secondly, after speeding up a good algorithm, we try to improve the segmentation of a very fast algorithm: Seeded Watershed [Meyer, 1992]. For application as typical superpixels, the main issue of Watershed segmentations are the highly irregular distribution, size and shape of the resulting segments. We propose and evaluate the extension with a grid initialization and a compactness constraint that turns Watershed segmentation

from an oversegmentation algorithm into a "real" superpixel algorithm. This shifts the balance at several trade-offs between metrics for segmentation compactness, quality and stability which we are going to evaluate in detail in the results section 4.5. Fig. 4.2 shows first example segmentations for the original and the novel algorithms. While the pSLIC segmentations are visually similar to the results of SLIC, they can be computed much faster. The proposed CWS algorithm brings the Watershed segmentation visually closer to the results of the high quality SLIC and pSLIC segmentations and is more than twice as fast as pSLIC.

The following sections introduce the proposed algorithms starting with speeding up SLIC, followed by details on how to introduce compactness in a Watershed segmentation.

4.2. Preemptive SLIC: Making a good algorithm faster

Simple Linear Iterative Clustering (SLIC) [Achanta et al., 2012] is based on the concept of a local k-means clustering. The cluster centres are initialized on a uniform grid in the image plane. *Local k-means* denotes that each cluster considers only pixels in its spatial neighbourhood. The metric during clustering is a weighted combination of colour distance and spatial distance in the image plane. The weight of the spatial component (the compactness parameter) influences the regularity of the resulting segment's shape and size.

SLIC provides good segmentation quality and stability and can be considered as a state of the art superpixel segmentation algorithm according to the benchmarks in section 3). Although SLIC belongs to the class of faster superpixel algorithms (e.g. compared to Normalized Cuts), the runtime of about 100 ms can be considered the main drawback (e.g. compared to Watershed). In the following, we present our approach on speeding up SLIC by about factor five. The results section evaluates the impact of this shift in the trade-off between quality and runtime.

Starting from the implementation of Achanta et. al[1], we modify two parts of SLIC: we change the used colour cues (dropping runtime from about 100 ms to 55 ms) and we propose an algorithmic approximation, yielding a final runtime of about 20 ms for images of size (481×321).

A runtime analysis of the original implementation showed, that almost half of the time is spent for colour conversions from RGB to LAB. There exist several fast approximations and implementations for this colour space conversion. In [Neubert and Protzel, 2014] we replaced the original colour conversion with an approximation of the LAB colour space. However, directly working on the input RGB colour input yields similar results and omits the computations for the colour conversion completely. Therefore, the results presented in section 4.3 compare the segmentations on the original LAB and the input RGB colour cues.

For a further speedup, we propose an approximation of the original algorithm. The main idea of SLIC is to use k-means in a local manner by reducing the potential member

[1] `http://ivrg.epfl.ch/research/superpixels` [Last accessed: 2015-08-14]

pixels for each cluster to a local neighborhood. However, this is just half-way gone: SLIC uses a single, global termination criterion. We propose to use a *local* termination criterion for each cluster to avoid revisiting clusters and image areas without any major changes since the last iteration. This preemptively stops the evolution of segment boundaries in homogeneous image regions. The termination criterion of the original SLIC algorithm is the following:

> *If the maximum number of iterations is reached or if there is no major change in any cluster, the algorithm finishes.*

In fact the original implementation uses a fixed number of 10 iterations. However, this way, potentially large image parts without any changes in the previous iteration(s) are updated every new iteration of the main k-means loop. Thus, we propose to introduce an individual termination criterion for each cluster:

> *If there has not been any major change in this cluster or any neighbouring cluster in the last iteration, do not update this cluster.*

The global termination criterion is modified similarly. The oriSLIC implementation runs for a fixed number of ten iterations, Preemptive SLIC also stops if the cluster converged to a stable solution (as it is very common for such iterative algorithms):

> *If the maximum number of iterations is reached or only few clusters have been updated, we are done.*

This significantly reduces the number of pixel accesses at homogeneous image regions. Even if the same number of iterations in the main loop are processed, not all clusters are updated in each iteration. However, a cluster that has not been updated in one iteration can become active in the next if a cluster in its neighbourhood had a major change. Thus, the additional computational overhead reduces to counting the number of added or removed pixels for each cluster and to evaluate this number for each cluster and its neighbourhood before deciding whether to update this cluster or not in the current iteration.

The listing in algorithm 1 gives details on the algorithmic steps. The implementation is based on oriSLIC by Achanta et al. These are the major changes from the original SLIC:

- The changed termination criteria can be seen in line 4, here the original SLIC implementation runs for a fixed number of iterations. The termination criterion in pSLIC is more similar to the commonly used criterion in k-means, where iteration stops if the clustering converged to a stable solution. The used termination criterion involves a parameter t_i.
- The most important difference is the novel local preemption criterion in line 6. SLIC updates all clusters in each iteration of the main loop. pSLIC only updates

Algorithm 1: Outline of the algorithmic steps of Preemptive SLIC.

```
   Data: Image, number of cluster, thresholds t_i and t_c
   Result: Assignment of pixels to cluster
1  Initialize cluster centres on a regular grid;
2  Initialize region of each cluster as all cells in its 8 neighbourhood on this grid;
3  Mark all clusters as changed;
4  while Less than maximum number of iterations AND More than t_i cluster changed do
       // assign
5      foreach Cluster do
6          if More than t_c pixels changed in the region of this cluster in the last iteration then
7              foreach Pixel in the region of this cluster do
8                  Compute distance of this pixel to the cluster centre;
9                  if Current cluster is the most similar cluster so far then
10                     Assign label of this cluster to the pixel;
11                 end
12             end
13         end
14     end
       // count changes
15     foreach Pixel do
16         if Label changed since last iteration then
17             foreach Cluster that has the pixel in its region do
18                 Increase number of changed pixels for this cluster;
19             end
20         end
21     end
       // update
22     foreach Cluster do
23         Update centroid using all assigned pixels;
24     end
25 end
26 Enforce cluster connectivity;
```

those clusters for which a significant change has happened in the neighbourhood. The significance is evaluated as rate on the total number of pixels controlled by a parameter t_c.

- To evaluate the amount of change in each cluster, the number of changed pixels has to be counted for each cluster. This is done in lines 15-21. This is not necessary for the original SLIC and causes additional computational effort that is also evaluated in section 4.3.

The parameters t_i and t_c can be set independently. In the following evaluation, we derive both from a single parameter t that quantifies the rate of change. We set $t_i = t \cdot n$, the rate t times the number of clusters, and $t_c = t \cdot 9 \cdot (w \cdot h)/n$, the rate t times the average number of pixels in a cluster region (the cluster and its 8-neighbourhood). Fig. 4.4 shows the influence of this parameter t on the number of inner loop calls (line 8 in algorithm 1) and the resulting runtime.

Qualitative evaluation on the segmentation quality can be seen in the example segmentations using SLIC and pSLIC in Fig. 4.3. The effects on segmentation quality and runtime are more exhaustively evaluated in the following results section 4.3.

Contribution 4.1 *This section proposed the integration of a local preemption criterion in the SLIC algorithm.*

4.3. Results of Preemptive SLIC

Fig. 4.4 shows an evaluation of the influence of the preemption threshold on the number of inner loop calls. The parameter evaluation is done on the BSDS *val* dataset. It can be seen how the number of inner loop calls decrease with increasing threshold. It converges to a minimum for thresholds between 0.2 and 0.3. The resulting runtime is shown beneath. A positive influence on the runtime is visible for thresholds larger than about 0.1. The behaviour for RGB and LAB colour spaces is similar, however, the number of calls in LAB colour space decreases faster and reaches the minimum earlier.

The right part of Fig. 4.4 shows the runtime improvement for the proposed Preemptive SLIC. The blue curve is the runtime of the original SLIC implementation, it takes about 100 ms per image. The fastest configuration of the proposed Preemptive SLIC requires about 20 ms per image - a speedup of factor five. The other curves show intermediate steps. The green curve includes the additional time that is spent for counting the changes in the clusters (lines 15-21 in algorithm 1). Our pSLIC implementation with LAB colour space and an preemption threshold of 0 (pSLIC LAB 0) produces the same segmentations (up to minimal differences) as the original SLIC implementation, but needs slightly more time due to counting of the changes in the clusters. Setting the preemption threshold to 0.25 (pSLIC LAB 0.25) yields the black curve that shows a considerable improvement. Using SLIC or pSLIC directly in RGB colour space is even faster (pSLIC RGB 0, the yellow curve). Combining the faster colour space and the preemption provided by the proposed approach is about factor three faster than the original RGB version (pSLIC RGB 0.25).

SLIC already performed well at segmentation quality before (cf. chapter 3). The boundary recall and undersegmentation error measurements in Fig. 4.5 show the slight decrease in segmentation quality for the changed colour space and Preemptive SLIC. The larger decrease is due to the changed colour space. The preemptive termination causes a comparatively small additional loss.

The general similarity between the original SLIC and Preemptive SLIC segmentation results can be seen in the example segmentations in Fig. 4.2. The local termination criterion and the resulting preemptive stop of the evolution of superpixel boundaries cause segments to remain more similar to the initial grid-like shape. This manifests as straight vertical or horizontal segment borders in homogeneous image areas. The results on all other presented metrics can be found in the appendix. In summary: Compactness from isoperimetric quotient indicates that the boundaries of pSLIC are more irregular,

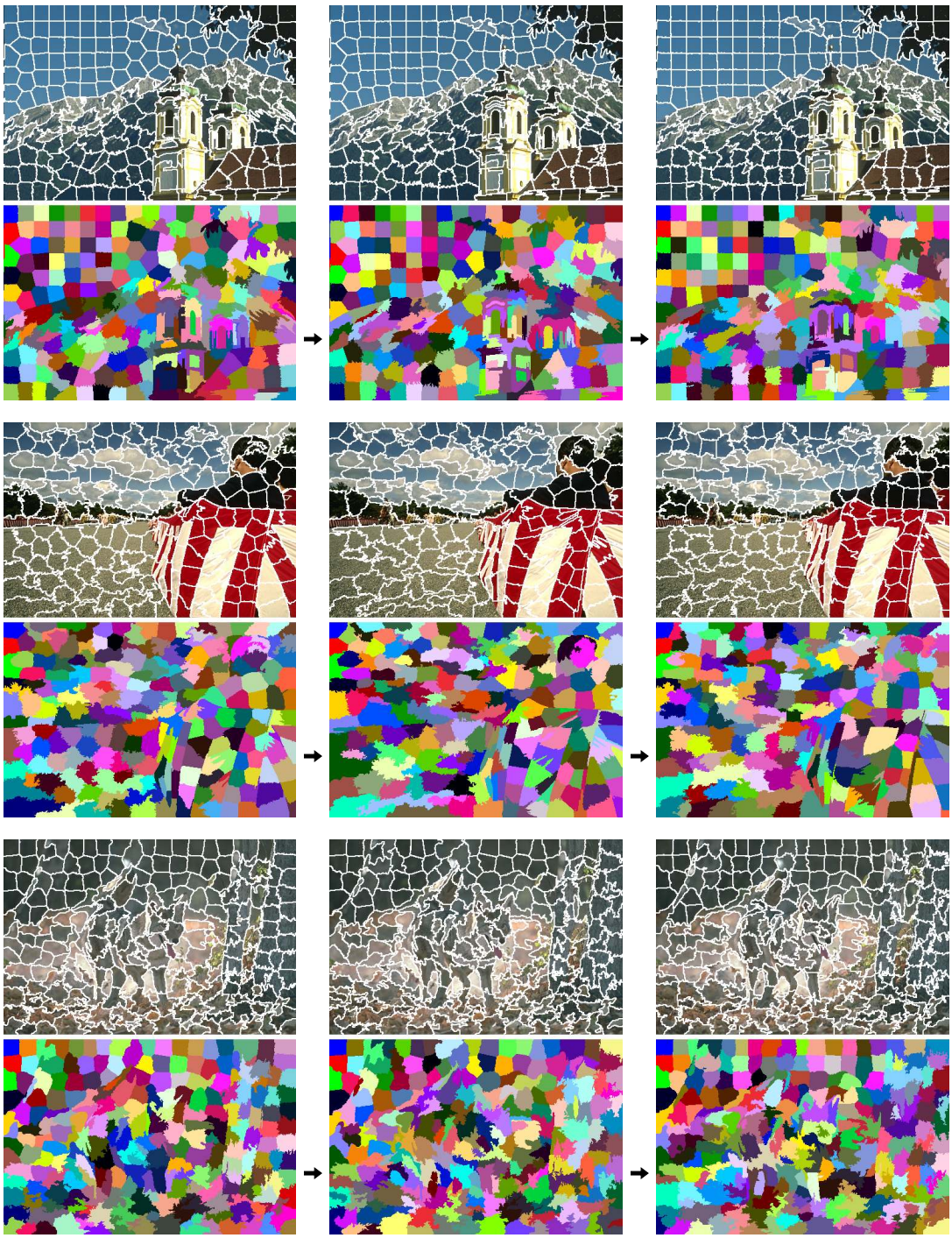


Fig. 4.3.: From SLIC to Preemptive SLIC. These are examples of three images with the original SLIC (left) and the proposed Preemptive SLIC. The middle column shows the influence of the switch to the RGB colour space (pSLIC t=0) and the right column the influence of the preemptive termination (pSLIC t=0.25). The results are visually similar, but pSLIC with t=0.25 is about five times faster.

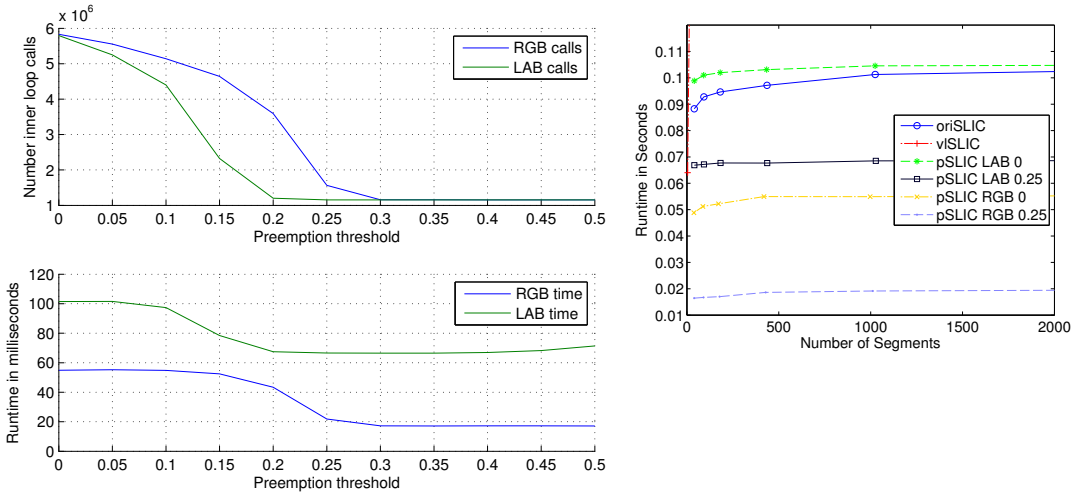


Fig. 4.4.: *(left)* Evaluation of the number of inner loop calls (line 8 in algorithm 1) and resulting runtime for different values of the preemption threshold parameter t on the BSDS *val* dataset. *(right)* Results of the existing SLIC implementations and the proposed Preemptive SLIC on the BSDS *test* dataset. Note: The runtime of vlSLIC (red) rapidly exceeds the shown plot. It is about 300 ms, please see Fig. 3.6 for details.

results on all other compactness measures are similar for SLIC and pSLIC. The RGB colour space version is more robust to Gaussian noise. The preemptive versions are more robust against Salt & Pepper noise, but more sensitive to geometrical image changes (affine image transformations and MUSE).

Result 4.1 *Preemptive SLIC is a faster alternative for SLIC. The segmentation quality is slightly worse while the runtime is improved considerably.*

4.4. Compact Watershed: Getting from oversegmentations to superpixels

While the introduction of superpixels stated that the terms superpixel and oversegmentation are used interchangeably in this thesis, this section deviates from this convention. Here I want to explicitly distinguish a superpixel segmentation as an oversegmentation with regular grid like distribution of segments with similar shapes and sizes - similar to enlarged pixels or the results of SLIC.

Watershed segmentation is a very fast algorithm. Plain watershed [Digabel and Lantuéjoul, 1978] suffers from irregularly sized and shaped segments and strongly varying boundaries. *Seeded* watershed reduces this problem significantly [Meyer, 1992]. However, the user still has no influence on the segmentation characteristics beyond choosing the number of segments. In particular, the resulting segments still vary considerably in size and shape as can be seen qualitatively by example segmentations of WS in sec-

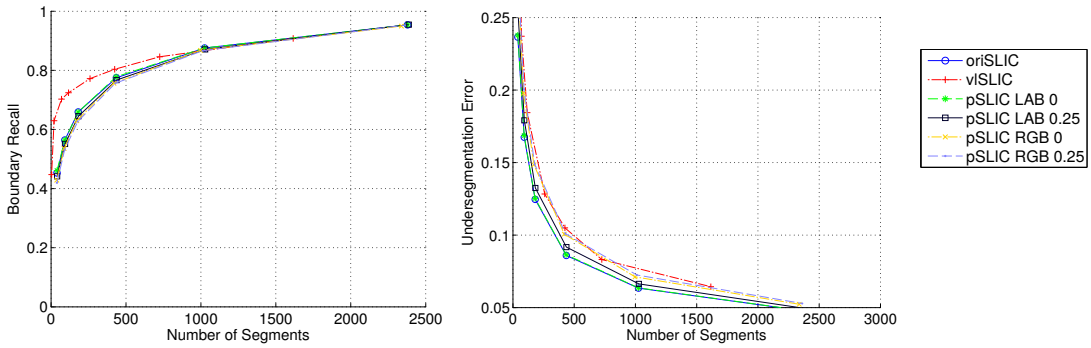


Fig. 4.5.: Results of the existing SLIC implementations and the proposed Preemptive SLIC. The segmentation quality is almost not affected by the preemption criterion. However, changing the colour space to RGB slightly increases the undersegmentation error. At boundary recall, the vlSLIC implementation performs superior - however, it is even slower than oriSLIC.

tion 2.3.2 or in the bottom left image in Fig. 4.2. and quantitatively in the evaluation presented in section 3.6.

This part of the thesis describes our approach, presented in [Neubert and Protzel, 2014], on creating compact superpixel segmentations at the high speed of watershed segmentations. In a first step, arranging the seeds on a grid creates a segmentation that is visually closer to a real superpixel segmentation in contrast to an arbitrary oversegmentation. Secondly, we propose to incorporate a controllable compactness constraint in a watershed segmentation to regularize the segment shape and size and give the user additional influence on the segmentation results. The presentation of the algorithm is followed by an evaluation of its influence on segmentation quality and compactness properties.

Watershed based on an ordered set of priority queues

The idea of watershed segmentation originates from [Digabel and Lantuéjoul, 1978]. The Seeded Watershed algorithm has already been introduced in section 2.3.2.5, I want to shortly refresh the algorithmic approach and give more details on the approach by Meyer [1992] that is the basis of our Compact Watershed algorithm. The intuitive idea comes from geography: when a landscape is flooded by falling waterdrops, dependent on the amount of water, there are basins filled with water and dividing ridges between them. These ridges are the watersheds. Since watershed segmentation is a well known algorithm, there are various algorithmic implementations and adaptations, see [Roerdink and Meijster, 2000] for an overview.

Meyer [1992] formalizes the idea of watershed in terms of a seeded region growing based on an ordered set of priority queues. Fig.4.6 illustrates this central data structure that is also used in our Compact Watershed implementation. Each of the orange coloured bins is a priority queue, a first-in-first-out data structure that can be efficiently implemented

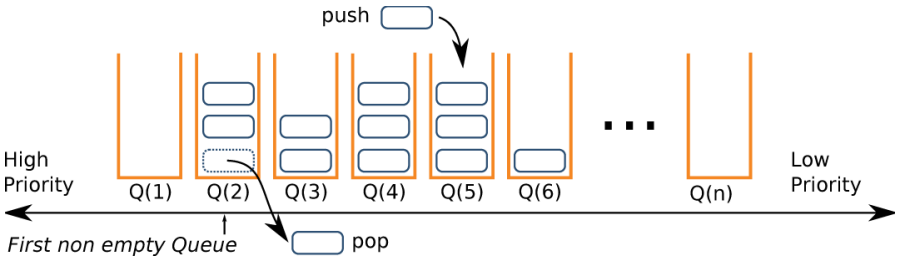


Fig. 4.6.: Illustration of the set of ordered priority queues presented in [Meyer, 1992] and used in the Compact Watershed algorithm.

based on ring buffers with moving start and end indexes or in the form of a doubly linked list. Each priority queue provides the operation *push* to insert an element at the top and the operation *pop* to retrieve (and remove) the bottom element. The set of priority queues is ordered, i.e. all elements in queue $Q(1)$ have a higher priority than elements in $Q(2)$ and so on. Using this combined data structure, elements can be inserted at different priority levels using *push* at the corresponding queue. *pop* operations only happen at the first non empty queue, the queue with highest priority that contains an element. Basically this can be seen as a breadth-first search incorporating priorities.

Meyer [1992] uses an ordering relation for flooding, based on the watershed-height of pixels in the landscape- and thus for insertion in the ordered set of priority queues.

1. A point is flooded before another point if the latter is regarded higher in the landscape.

2. On plateaus: First all lower ordered points are flooded before the water floods from the surrounding of the plateau towards the interior.

For grey level images, the watershed-height of a pixel is its grey level distance to an already flooded pixel. For colour images, Meyer [1992] uses the maximum distance in a colour channel. Starting from the seed pixels, the neighbours are recursively processed by fist computing their watershed-height, quantizing this watershed-height into a priority and inserting the pixel to the corresponding queue in the ordered set of priority queues. The next pixel to be processed is selected by a *pop* operation of the first non empty queue. Neighbours that are already processed are not inserted into the queues again and the process stops when all pixels are flooded.

Incorporating compactness

The original watershed algorithm suffers from highly irregular segments [Digabel and Lantuéjoul, 1978]. The choice of the seeds has a large influence on the resulting segmentation. In the presented basic seeded watershed algorithm the seeds are obtained from "minima of a severely filtered image" [Meyer, 1992, p. 303]. The particular design of these filters has large influence on the number and distribution of the resulting minima. Moreover, using the local minima results in an oversegmentation with very differently

shaped and sized segments, orderless distributed in the image as can be seen in the shown example segmentations and the compactness benchmark results in section 3.

In a first step, we therefore arrange the seeds on a regular grid to obtain a superpixel-like initial arrangement. This yields much more regularly distributed resulting segments. However, they still vary strongly in size and shape. Example segmentations can be seen in the bottom mid image in Fig. 4.2 and in the middle column of Fig. 4.10.

Secondly, to give the user control over the size and shape, we propose to incorporate a compactness constraint in [Neubert and Protzel, 2014]. To integrate a prior on the resulting segment geometry, we change the criteria for the priority queue in which a pixel is inserted based on the distance to the seed. More exactly, the length of the path that has been taken to flood this pixel from a seed. This induces a compactness constraint similar to the weighting of appearance difference and spatial distance in SLIC.

Contribution 4.2 *This section proposes the incorporation of compactness in watershed segmentations by using a regular grid initialization and an additional distance term in the watershed computations.*

Algorithm 2 shows the processing steps. The algorithm is similar to the one presented in [Meyer, 1992] but incorporates a compactness parameter c. In line 4, this parameter is used to compute an increment *stepC* that is subsequently used to adapt the priority of pixel. The computation of this increment is described later in this section, for now let us assume it to be a non negative number, e.g. 1. The key idea is to store for each pixel the length of the path to the corresponding seed. If a new pixel is explored, the length of its path is the sum of this increment and the length of the path of the neighboured pixel from which it is explored.

The priority queues are initialized in lines 5-11: Starting from the seeds and an empty set of priority queues, all neighbours of seeds are inserted into the queue (line 10) corresponding to their distance in appearance (line 7). The *pathLength* is constant for all neighbours to a seed (line 8). The computation of the index of the priority cues is explained beneath together with the computation of the *stepC*.

Lines 12-30 describe the recursive flooding procedure. As long as there is a non empty queue (lines 13 and 14), the remaining pixel with the highest priority is obtained (line 15). If this is a boundary pixel, the pixel is a watershed pixel, otherwise all neighbours are processed similarly to those of the seeds. The crucial difference is the computation of the *pathLength*, that now depends on the previously travelled path (line 22).

A fast implementation of the algorithm described in [Meyer, 1992] is available as part of the OpenCV library. Our implementation of Compact Watershed is built upon this implementation and freely available (cf. section 1.3 on open source software releases).

The influence of the compactness constraint on the resulting segmentations can be qualitatively seen in Fig. 4.7. Section 4.5 will provide a more detailed quantitative evaluation based on the compactness (and other) criteria presented in section 3.

The odd *stepC* values used for the segmentations in Fig. 4.7 are due to the fact that they are not manually chosen but derived from the compactness weight c, the image size

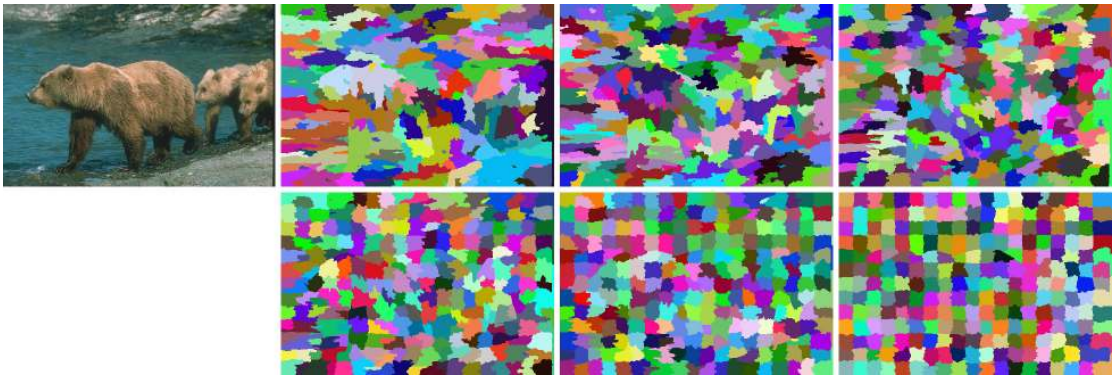


Fig. 4.7.: Example CWS segmentations with varying compactness. *(Row major order)* Input image, segmentation with compactness parameter $c = (0, 10, 25, 50, 100, 250)$ resulting in $stepC = (0.0648, 0.1619, 0.3238, 0.6477, 1.6192)$ for this image.

and the wished number of segments in line 4 of algorithm 2. Why and how this is done is the topic of the following section.

Compactness parameter selection

Dependent on the application of the resulting segmentation, having influence on the distribution, size and shape of the segments may be helpful or even crucial. The above described algorithm gives the user a parameter to influence these properties. However, such a parameter poses the additional problem of selecting a useful value for the task at hand. While we want to be able to select the level of compactness, we do not want to need to adjust this parameter for different numbers of segments or different image sizes.

In the proposed Compact Watershed algorithm this is approached in line 4, where the path increment stepC is computed from the compactness parameter c, the size of the image and the number of segments we want to obtain.

Given an image of size $h \times w$ with n superpixels. Assuming a perfectly compact segmentation as composed from equally sized segments with square shape, the edge length s of this square is

$$s = \sqrt{\frac{h \cdot w}{n}} \tag{4.1}$$

The square is embedded in an image grid. A square in euclidean metric space with norm L_2 equals a circle in Chebyshev norm L_∞ (the maximum distance in a coordinate, also known as chessboard distance). In L_∞ norm, the radius of this circle $r = s/2$ is the minimum number of steps in the Compact Watershed algorithm on a path connecting the seed to the boundary of the perfectly compact segment.

However, the boundary pixel can be reached using other paths than a direct connection. The taken path can be modelled as a random walk on a two dimensional lattice.

Algorithm 2: Outline of the algorithmic steps of Compact Watershed.

```
Data: Image, number n of segments, compactness c
Result: Segment label for each pixel
Initialize seeds on a regular grid;
Assign individual label for each seed;
Create ordered set of priority queues Q[nQueues];
// Compute compactness step for this image and number of segments
stepC = getCompactnessStep(c, size(image), n);
// Initialize neighbourhood of seeds
foreach Neighbour n of a seed s do
    n.label = s.label;
    n.deltaA = getAppearanceDistance(s, n);
    n.pathLength = stepC;
    idxQ = getQueueIdx(n.deltaA, n.pathLength);
    push(Q(idxQ), n);
end
// Recursively fill the basins
while True do
    idxQ = findFirstNonEmptyQueue(Q);
    if idxQ ≤ nQueues then
        // Pop next element of this queue
        p = pop(Q(idxQ));
        if There is a neighbour with different label then
            Mark p as watershed;
        else
            // Insert all neighbours in the ordered set of queues
            foreach Neighbour n of p do
                n.label = p.label;
                n.deltaA = getAppearanceDistance(p, n);
                n.pathLength = p.pathLength+stepC;
                idxQ = getQueueIdx(n.deltaA, n.pathLength);
                push(Q(idxQ), n);
            end
        end
    else
        break;
    end
end
```

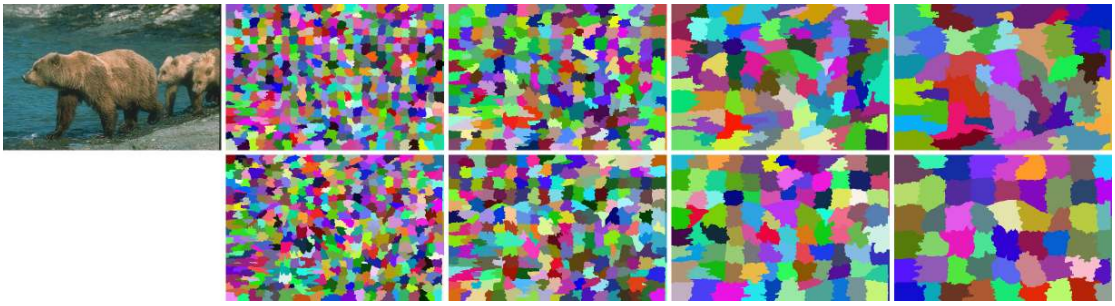


Fig. 4.8.: Robustness of stepC selection for different numbers of segments. The top row shows different segmentations of the left input image with varying numbers of segments n=(500, 250, 100, 50) and a compactness parameter of 50. The characteristic of the compactness for all numbers of segments is visually similar. This is in contrast to the segmentations in the bottom row, where the stepC is fixed for all segmentations and the coarse level segmentation is qualitatively much more (too) compact.

The expected distance from the starting point for a two dimensional random walk with k steps is of order $\sqrt{k}$.

Therefore, to reach the boundary at distance r, we expect the number of steps to be in the order of r^2. The expected total costs c due to compactness for such a perfectly compact boundary pixel and costs per step of *stepC* are in the order of:

$$c = r^2 \cdot stepC = \left[\frac{\sqrt{\frac{h \cdot w}{n}}}{2} \right]^2 \cdot stepC = \frac{h \cdot w \cdot stepC}{4 \cdot n} \tag{4.2}$$

Fixing this cost for segmentations of arbitrary image sizes and numbers of segments yields a constant cost due to the compactness constraint, independent from image size or number of segments. Therefore, we compute the increment that is caused by adding an additional pixel to a segment as follows:

$$stepC = \frac{c}{r^2} = \frac{4 \cdot n \cdot c}{h \cdot w} \tag{4.3}$$

Now the user can select the compactness parameter c that linearly weights the costs due to compactness and those due to pixel colour or intensity values.

The compactness of the resulting segmentations will appear similar, independently of the image size and the number of segments, in contrast to fixed increments. For different numbers of segments this is illustrated in Fig. 4.8 and for different image size in Fig. 4.9.

To get a quantitative evaluation of the influence of the compactness constraint, the following section presents results on the metrics described in section 3.

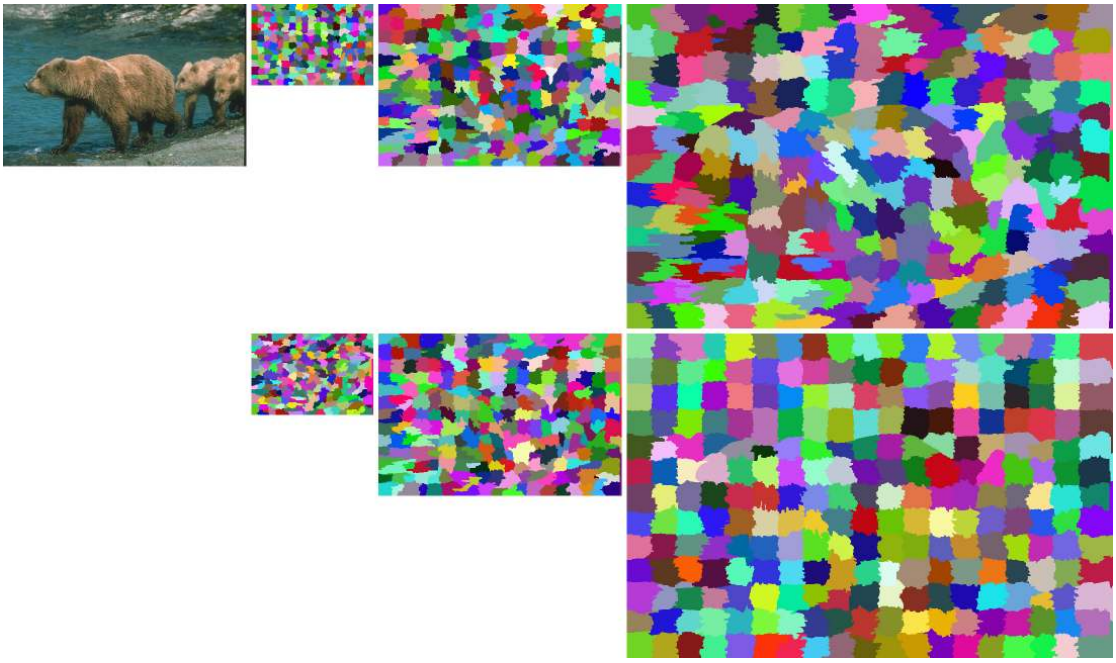


Fig. 4.9.: Robustness of stepC selection for different image sizes. The top row shows different segmentations of the left input image for rescaled images with a compactness parameter of 50 and segmentations with 250 superpixels. From left to right, scale is $(0.5, 1, 2)$. Similar to Fig.4.8, the characteristic of the compactness of the segments for all resized images is visually similar. Again, this is in contrast to the segmentations in the bottom row, where the stepC is fixed for all segmentations and the segmentation of the upscaled image is qualitatively much more (too) compact.

Result 4.2 *The increment for a region growing step should be chosen dependent on the image size and the number of segments. Based on the assumption of a two dimensional random walk, the increment can be estimated as* $stepC = \frac{c}{r^2} = \frac{4 \cdot n \cdot c}{h \cdot w}$. *$c$ is a parameter that is provided to the user to adjust the level of compactness. This choice of the increment creates segmentations with qualitatively similar compactness characteristics for different image sizes and numbers of segments.*

4.5. Results of Compact Watershed

The presented incorporation of a compactness constraint influences the resulting segmentations as can be qualitatively seen by the example segmentations in Fig. 4.10 and other figures in this section. The additional constraint is expected to influence different properties of the original algorithm. Of particular interest is the influence on segmentation quality, since the compactness constraint is independent from colour and intensity values of pixels. To quantitatively evaluate the impact, we can use the above presented metrics to evaluate compactness, quality, runtime and stability.

Fig. 4.10.: From Watershed to Compact Watershed. These are examples of three images segmented with the original Watershed with seeds at local gradient minima (left) and the proposed Compact Watershed. The middle column shows the influence of the grid initialization (CWS c=0) and the right column the influence of the compactness constraint (CWS c=25).

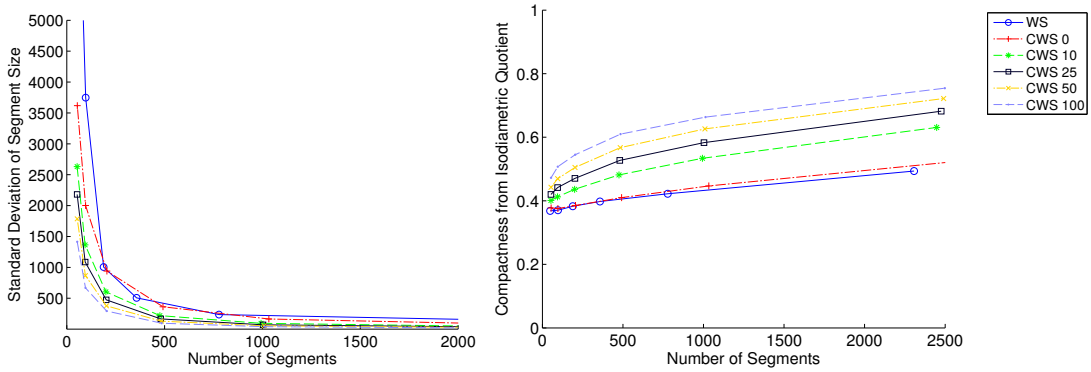


Fig. 4.11.: Results compactness. Results of Watershed and Compact Watershed with varying compactness weights on standard deviation of segment sizes and compactness from isodiametric quotient.

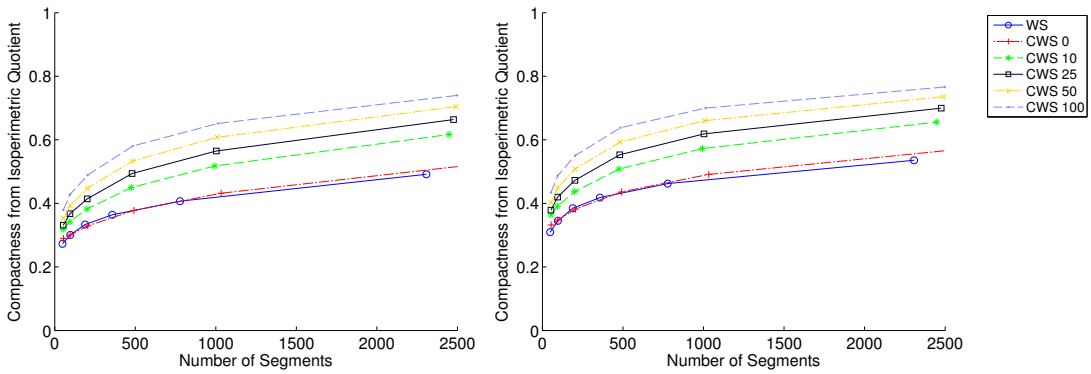


Fig. 4.12.: Results isoperimetric quotient. Results of Watershed and Compact Watershed with varying compactness weights on compactness from isoperimetric quotient. The left figure belongs to the original label images and the right figure for the smoothing with a morphological open of radius 5 as described in section 3.6.

Fig. 4.11 compares the seeded watershed algorithm without compactness constraint **WS** with the proposed Compact Watershed algorithm **CWS** with different compactness values $c \in \{0, 10, 25, 50, 100\}$. The setup $c = 0$ includes the grid initialization, but eliminates the influence of the compactness constraint. The increasing weight of the compactness constraint brings the segments closer to a square form as can also be seen by the example segmentations with increasing compactness in Fig. 4.7. The quantitative evaluation of standard deviation of segment size and the isodiametric quotient in Fig. 4.11 confirms this visual impression and reveals a considerable improvement in compactness already at small compactness values. With increasing compactness weight, the variation of the segment size becomes smaller and the segments eccentricity decreases as is indicated by the isodiametric quotient.

The general improvement in isoperimetric quotient for increasing compactness for the original (left) and the smoothed (right) label images in Fig. 4.12 confirms the result

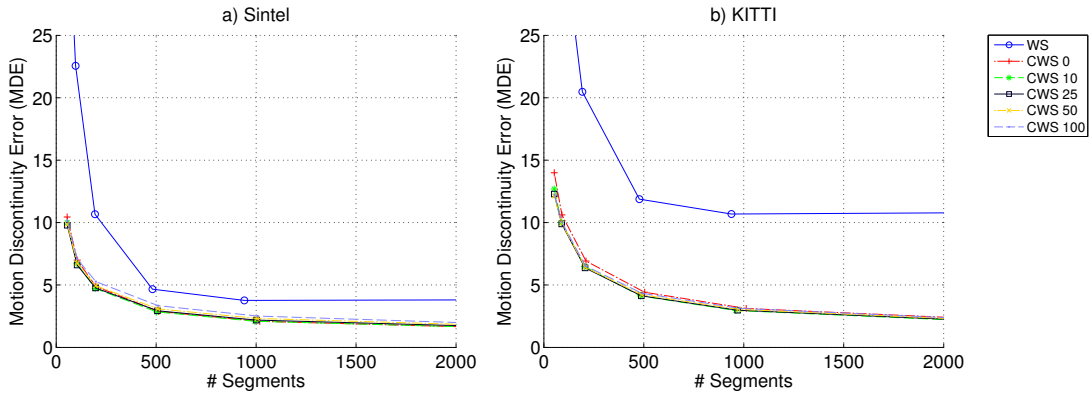


Fig. 4.13.: Results MDE. Results of Watershed and Compact Watershed with varying compactness weights on motion discontinuity error MDE.

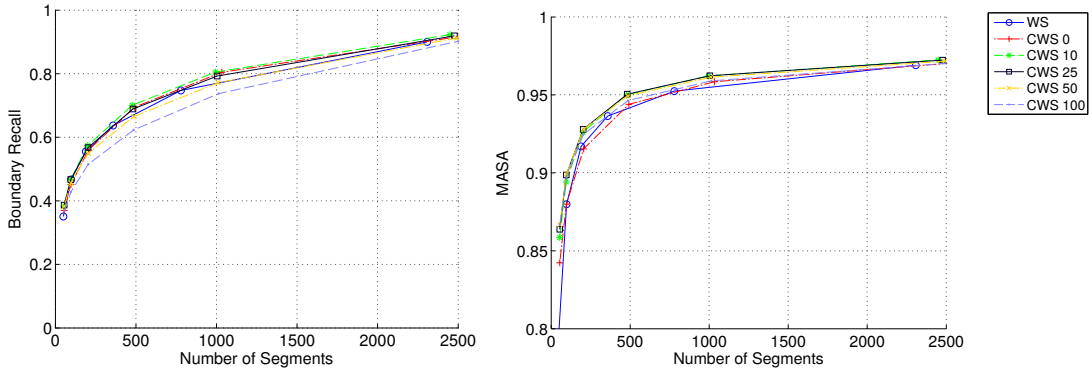


Fig. 4.14.: Results quality. Results of Watershed and Compact Watershed with varying compactness weight on segmentation quality.

from the isodiametric quotient indicating a reduced eccentricity of the segments. The comparison of the two evaluations on the original label image and the smoothed label image shows that the compactness constraint has only a small influence on the small scale irregularities of the boundaries since the improvement in isoperimetric quotient for increasing compactness is very similar for both plots.

While the geometrical compactness measures showed similar behaviour for the seed initialization by local gradient minimum (WS) and the grid initialization (CWS), there is large difference for motion discontinuity error shown in Fig. 4.13. The regularly distributed segments are more likely to create segment boundaries close to the motion gradients caused by differently moving image parts. For this metric, the additional increase of the compactness parameter shows only minor further improvement.

The evaluation of segmentation quality shown in Fig. 4.14 shows an improvement on maximum achievable segmentation accuracy when using the compactness constraint. The compactness constraint improves the reconstruction of objects from the resulting segments by avoiding segments that overflow object boundaries by very large amounts.

A suitable choice of the compactness parameter also slightly improves boundary recall. However, increasing the influence of the compactness term too much, results in lower boundary recall.

Results in the other metrics can be found in the appendix A.2. In summary: The very fast runtime is not affected (please keep in mind that runtime for CWS already includes the computational effort for seed selection). Undersegmentation error improves similar to MASA. The robustness towards image noise is increased but the compactness has a negative influence on the boundary recall in the presence of affine image transformations. The compactness constraint is an appearance independent influence purely based on geometry and is supposed to negatively affect the boundary repeatability in image regions with small image gradients that are subject to a geometric change as it is imposed by the affine transformations. The combined stability evaluation based on the ground truth optical flow confirms these results: While the MUSE metric improves for the Sintel data with moving foreground objects, it decreases for the Kitti dataset with a higher amount of overall background motion.

Result 4.3 *Compact Watershed gives the user ability to create segmentations with different compactness characteristics. This turns Watershed from an oversegmentation algorithm into a compact superpixel segmentation algorithm. The fast runtime is not affected.*

4.6. Conclusion

This chapter presented two adaptations of superpixel segmentation algorithms and evaluated the changes in trade-offs between various performance measures. The first new algorithm is Preemptive SLIC. It runs at frame-rate (>30 Hz) on a standard desktop CPU and conserves the high segmentation quality level of SLIC. The runtime improvement comes from two steps: the change of the used colour cues and a preemptive termination criterion for each local cluster.

The second algorithm is Compact Watershed. It incorporates a compactness constraint in a grid seeded watershed segmentation and gives the user control over important characteristics of the segmentation. This turns Watershed from an oversegmentation algorithm to a real superpixel segmentation algorithm. The compactness is controlled with a single parameter that controls the level of compactness independently of the number of segments or the image size. Compact Watershed takes about 10 ms for segmentation of an image with a size of (481×321) pixels.

We evaluated the influence of the algorithmic changes on the trade-offs between various performance measures. The individual importance of the different performance indicators depends on the application of the resulting segmentations. However, the proposed algorithms improve some performance measures to a large amount, while only slightly decreasing others. Open source implementations of both algorithms are available, cf. section 1.3.

This finishes the first part of this thesis, which was about superpixel segmentation algorithms. The rest of this thesis is about the connection of superpixels and loop closures. The first loop I want to close goes back to the introduction of available segmentation approaches in section 2.3.2, where I can now add two additional superpixel segmentation algorithms to the list of available approaches:

Properties pSLIC	
Technique	Local k-means with preemption
Cue(s)	Colour, size
Compact?	Yes
Runtime	$\sim$ 30 Hz

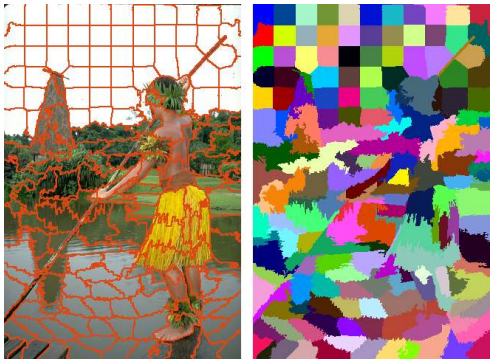


Fig. 4.15.: Example pSLIC segmentation

Properties CWS	
Technique	Seeded region growing with compactness constraints
Cue(s)	Colour, size
Compact?	Yes
Runtime	$\sim$ 100 Hz

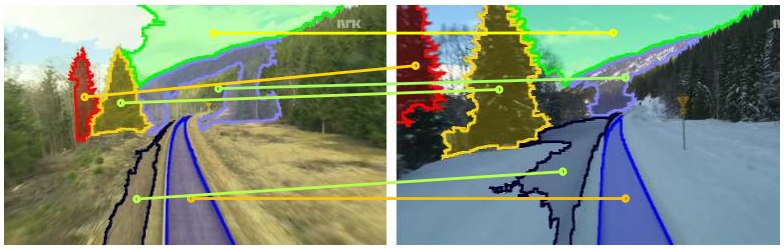
Fig. 4.16.: Example CWS segmentation

These two novel algorithms have already been included in the recommendations for selecting a suitable algorithm for the task at hand in section 3.7. All remaining loop closures of this thesis are in the context of place recognition for mobile robots operating in changing environments.

PART

II

SUPERPIXEL FOR PLACE RECOGNITION IN CHANGING ENVIRONMENTS



The first part of this thesis introduced superpixel segmentations, existing and novel criteria and benchmarks to compare them, and finally two new segmentation algorithms. In this second part, superpixels are used in the context of a particular application: place recognition in changing environments. This part starts with an introduction of the place recognition problem in general and the challenges induced by changing environments. Only few approaches exist that are particularly suited for this challenging problem. I will propose and investigate two novel and somehow contrasting approaches how to use superpixel segmentations for place recognition in changing environments - both will show advantageous compared to existing methods.

The only way to make sense out of change is to plunge into it, move with it, and join the dance.

Alan Watts

5 The Problem of Place Recognition in Changing Environments

Robust long term operation including long term navigation is one of the major challenges in robotics today. Navigation of mobile autonomous robots is related to practical applications in related fields like driver assistance systems, automated transportation, self driving cars and indoor localization from hand held cameras. An essential prerequisite for navigating in an environment is an idea of the own position or location in the world - in form of absolute world coordinates, the metric or topological relative position to the navigation goal, or as information if and when the current place in the world has been visited before. Localization can be based on various sensors including GNSS (e.g. GPS) or 2d/3d laser range finders. This thesis deals with visual localization based on visual input obtained from cameras. Cameras are cheap and readily available in many devices. They are a great source of information about the environment and the human visual system demonstrates that many problems, including navigation, can be solved using vision as a primary source of information.

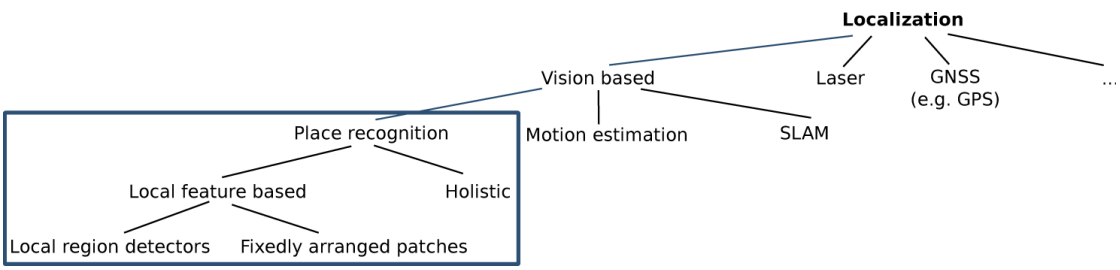


Fig. 5.1.: Overview of localization tasks. Topic of the second part of this thesis is visual place recognition.

Fig. 5.2.: This is meant by changing environments in this thesis. Each column shows an example image pair of the same scene, subject to severe appearance changes. The problem I want to solve is to decide whether two images show the same place or not despite these changes. Image credits per column from left to right: NRKbeta.no (cf. section 5.1.1), Arren Glover (cf. section 5.1.2), Milford and Wyeth [2012] (cf. section 5.1.3).

Fig. 5.1 shows a coarse taxonomy of problems related to vision based localization. Important tasks are visual place-recognition, visual motion estimation and visual simultaneous localization and mapping (SLAM).

> *Place recognition is the task to decide whether a given scene has been observed before. For visual place recognition, images are associated to places and the decision about place matchings is based on the evaluation of image similarities.*

Place recognition is a fundamental step for all types of localization. Recognizing the current position as a known place in the world induces a *topological localization*: the current position is topologically related to a single or multiple priorly seen places. Dependent on the application, this might be subsequently refined to a metric localization including the full transformation between the current pose and a map, which is the topic of visual motion estimation, the second type of visual localization tasks. The third task, SLAM, is the combination of creating a map while simultaneously localizing in this map. This map can be based on pure place recognition, resulting in a topological map (e.g. SeqSLAM [Milford and Wyeth, 2012]), place recognition in combination with motion estimations, coined semi-metric maps (e.g. RatSLAM [Milford et al., 2004]), or full 3d reconstruction resulting in dense or sparse 3d maps (e.g. PTAM, [Klein and Murray, 2007] or ORB-SLAM [Mur-Artal and Tardos, 2015]).

The increasing complexity from a plain place recognition to a full 3d reconstruction, results in different environmental sizes that are nowadays addressed by these algorithms - ranging from accurate mapping of a small desktop or room size environment up to deciding whether places across a several hundred kilometers route, recorded in winter and summer, are the same. The latter, matching image of places between summer and winter, is an example for place recognition in changing environments. Fig. 5.2 illustrates what is meant by "changing environments" in this thesis. There are two types

of dynamics in the environment: (1) dynamic objects and (2) changes of the whole scene due to changing illumination, changing weather, day-night cycles or seasonal changes. For dealing with dynamic objects, a common approach is to explicitly try to detect them and handle them differently from the static elements of the scene. Typical examples are people in indoor environments or cars in street scenes. However, handling single dynamic objects is *not* in the scope of this thesis - I rather want to address severe whole image changes as they are illustrated in Fig. 5.2.

Visual localization in small and medium scale environments (desktop, room or building scale environments) can be considered a mature field. At least from a theoretical point of view, many problems are solved and highly accurate 3d reconstructions can be created. In these environments, open problems are mainly related to challenging environments or illumination conditions and the robustness of the approaches, e.g. introspection capabilities to detect when the provided results are not reliable. Large scale place recognition, e.g. matching places across a 1,000 km route, in *non changing* environments has also been successfully demonstrated by combining techniques from mobile robotics, computer vision and information retrieval [Cummins and Newman, 2011].

In contrast, for recognizing places across severe appearance changes, as they are shown in Fig. 5.2, major problems are yet unsolved. Robots operating autonomously over the course of days, weeks, and months have to cope with significant changes in the appearance of an environment. A single place can look extremely different depending on the current season, weather conditions or the time of day. Since state of the art algorithms for autonomous navigation are often based on vision and rely on the system's capability to recognize known places, such changes in the appearance pose a severe challenge for any robotic system aiming at autonomous long term operation.

In the overall context of this thesis, an obvious question is: Why should *superpixels* be able to support place recognition in changing environments? Fig. 5.1 distinguishes holistic and local feature based approaches to place recognition. In terms of holistic approaches, superpixel segmentations provide an intermediate level representation of the *whole* image. The full image is divided into local parts, well aligned with object boundaries. Chapter 6 will use this intermediate level representation to learn to *predict* image changes in each local part of an image. These predicted parts are then composed into to a new, predicted image. The presented results will show, that existing holistic place recognition algorithms can benefit from such a prediction.

On the other hand, each superpixel is a local image region - better aligned to object boundaries than fixedly arranged rectangular patches. Moreover, independent from the image content, there will be a superpixel representation of the image. This is in contrast to interest point extraction like corner or scale space extrema detection. For example think of an image taken with poor illumination conditions, e.g. nightfall. Maybe there won't be any SIFT features in a dark area, however, there will be a superpixel since they cover the whole image and maybe it can provide useful information, e.g. about the skyline. Chapter 7 will present a novel approach to place recognition in changing environments based on locale region detectors including superpixels and powerful descriptors based on the output of Convolutional Neural Networks. The spatial arrangement of the

landmarks is included using a novel approach based on a star graph model and Hough voting. The experimental results will show how the proposed system outperforms existing methods.

Although this order is rather unconventional, before the presentation of related work in section 5.2, the following section 5.1 will present the datasets used for the later evaluation. This is intended to deepen the understanding of the challenges induced by changing environments. The datasets and related work will pave the way for two novel approaches incorporating superpixel segmentations in place recognition across seasons, changing weather and day-night cycles.

5.1. Datasets comprising changing environments

To run place recognition experiments for evaluation of existing and novel approaches, datasets with traversals of the same places under different environmental conditions are required. In particular, the learning based approach presented in chapter 6 poses strong requirements on the used training data. To better illustrate the problems that are imposed by the changing environments and to deepen the understanding of the severity of the occurring appearance changes, this section presents the datasets that are later used in the evaluation.

5.1.1. The Nordland dataset - A 3000 km journey through all seasons

In [Sünderhauf et al., 2013] we present the Norldand dataset, a more detailed presentation can be found in [Neubert et al., 2015b] [1]. It is based on the TV documentary "Nordlandsbanen – Minutt for Minutt" by the Norwegian Broadcasting Corporation NRK. This documentary provides video footage of the 728 km long train ride between the cities of Trondheim and Bodø in north Norway. The complete 10 hour journey has been recorded from the perspective of the train driver four times, once in every season. Thus the dataset can be considered comprising a single 728 km long loop that is traversed four times. As illustrated in Fig. 5.3, there is an immense variation in the appearance of the landscape, featuring a complete snow cover in winter, fresh and green vegetation in spring and summer, as well as coloured foliage in autumn.

In addition to the seasonal changes, different local weather conditions like sunshine, overcast skies, rain and snowfall are experienced on the long trip. Fig. 5.4 shows the altitude profile of the complete track and illustrates the high variance in appearance in a single season due to the different vegetation zones the train passes. Most of the journey leads through such natural scenery, but the train also passes through urban areas along the way and occasionally stops at train stations or signals.

The videos of the journey have been recorded at 25 fps with a resolution of 1920×1080 using a SonyXDcam with a Canon image stabilizing lens of type HJ15ex8.5B KRSE-V.

[1] This section 5.1.1 including Fig. 5.4 is published in very similar form in Robotics and Autonomous Systems, 69, Peer Neubert, Niko Sünderhauf, Peter Protzel, Superpixel-based Appearance Change Prediction for Long-Term Navigation Across Seasons, Pages 15-27, Copyright (2015), with permission from Elsevier. http://dx.doi.org/10.1016%2Fj.robot.2014.08.005

Fig. 5.3.: The Nordland dataset consists of the video footage recorded on a 728 km long train ride in northern Norway. The journey was recorded four times, once in every season. Frame-accurate ground truth information makes this a perfect dataset to test place recognition algorithms under severe environmental changes. The four images in each row above show the same place in winter, spring, summer and fall. Images licensed under Creative Commons (CC BY), Source: NRKbeta.no `http://nrkbeta.no/2013/01/15/nordlandsbanen-minute-by-minute-season-by-season/` [Last accessed: 2015-08-14]

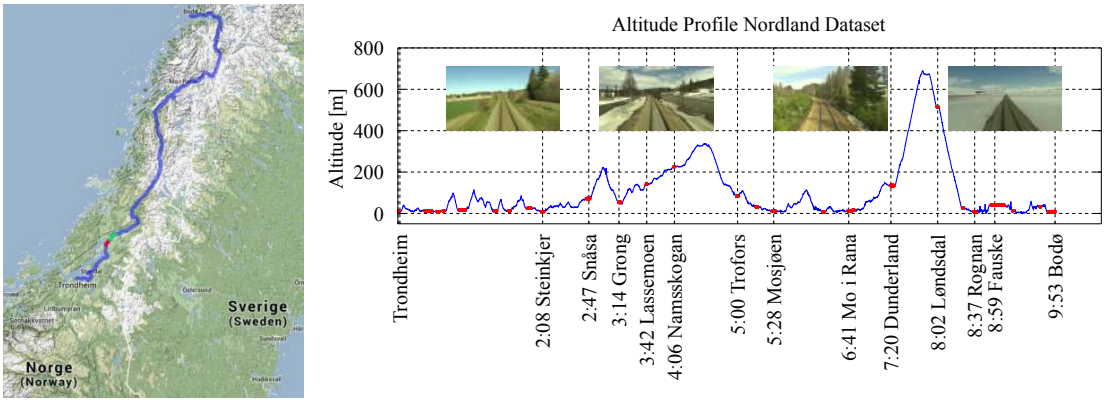


Fig. 5.4.: *(left)* GPS track of the complete Nordland dataset. The position of the training and validation data sets have been marked in red and green respectively. Map data ©2014 Google. *(right)* On its 728 km long way from Trondheim to Bodø, the train moves through different vegetation zones and altitude levels. The plot shows the altitude profile along with typical images from the corresponding areas of the spring journey. Some of the stations and their arrival time in the videos are marked as well. The red dots indicate that the train stopped at stations or signals. The plot has been generated using the GPS data that is published along with the videos.

GPS readings were recorded in conjunction with the video at 1 Hz. Both the videos and the GPS track are publicly available online[2] under a Creative Commons licence (CC BY). The full-HD recordings have been time-synchronized by the TV company NRK such that the position of the train in an arbitrary frame from one video corresponds to the same frame in any of the other three videos. This was achieved by using the recorded GPS positions and interpolating the GPS measurements to 25 Hz to match the video frame rate. The dataset therefore meets the requirements for almost pixel-accurately aligned images during the training phase. Notice that the image alignment is not absolutely perfect due to GPS interpolation errors and the 25 Hz sampling speed while moving at high velocities. The misalignment between two corresponding images in the dataset is, however, rather minor and in most cases only visible in objects close to the train tracks, such as trees or poles. An illustration can be found in appendix A.3.

Fig. 5.4 illustrates the complete journey from Trondheim to Bodø. Similar to the BSDS dataset, we split the track into a training, validation and test dataset. The *avconv* commands to extract the corresponding frames can also be found in the appendix A.3. The large *test* dataset covers the complete spring and winter journey. For testing we subsampled the 10 h videos at 0.1 frames per second resulting in 3,587 test images per season. For a velocity of about 70 km/h, this results in a distance of about 20 m between two frames. From the four available videos, the spring video best resembled typical summer weather conditions at the chosen part of the track. Thus, most experiments are based on the spring and winter videos (otherwise it is explicitly mentioned). All images were resized to 854×480 pixels.

For the training and validation experiments described in the following, we extracted 30 minutes from the spring and the winter videos at a frame rate of 2 Hz, starting at approximately 2 hours into the drive. The first 900 frames (about 8 minutes) from this 30 minute subset form a *training* dataset. The last 2700 frames of the remaining 22 minutes of this video subset are intended to be used as a *validation* dataset to evaluate the influence of parameter settings. There are 200 unused frames between the training and validation dataset since the train stopped at a station during this time. The much larger *test* dataset covers the complete spring and winter journey. For testing we subsampled the 10 h videos at 0.1 frames per second.

5.1.2. The Gardens Point dataset - Day and night with lateral shift

The Gardens Point Dataset is a vision only dataset comprising three runs of the same route at the Gardens Point Campus, Queensland University of Technology, Brisbane, Australia. It has been recorded by Arren Glover and is available online[3]. The dataset provides a total of $3 \cdot 200 = 600$ images captured with a hand held iPhone 5. There are two traversals at day and one at night. The first daytime traversal and the night traversal are on the right side of the path, the second daytime run is on the left side of

[2] `http://nrkbeta.no/2013/01/15/nordlandsbanen-minute-by-minute-season-by-season/` [Last accessed: 2015-08-14]

[3] `https://wiki.qut.edu.au/display/cyphy/Day+and+Night+with+Lateral+Pose+Change+Datasets` [Last accessed: 2015-08-14]

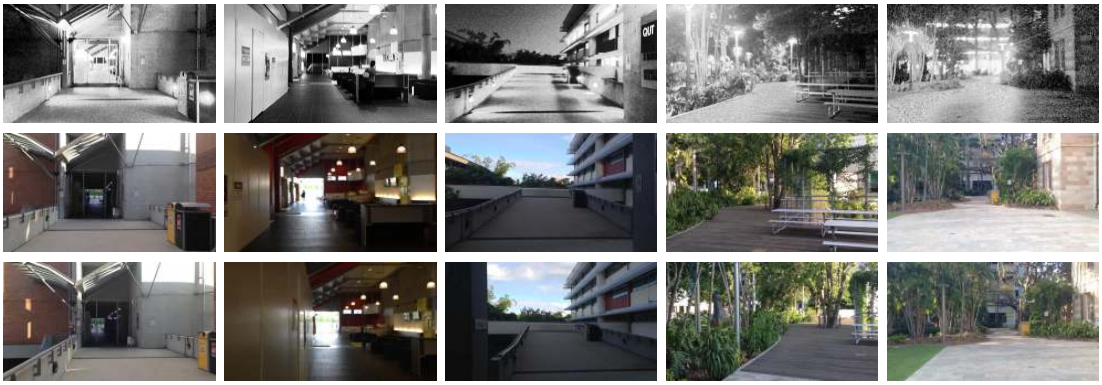


Fig. 5.5.: Example images of five places from the Gardens Point dataset. Each column shows three views of the same place, from top to bottom: NightRight, DayRight, DayLeft. GardensPoint image credits: Arren Glover.

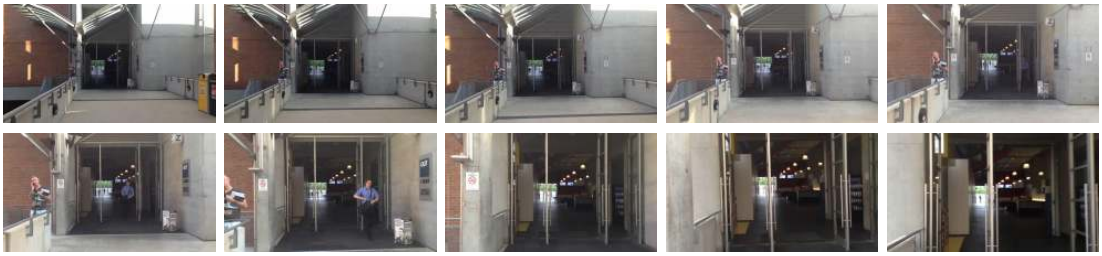


Fig. 5.6.: The first ten images of the Gardens Point DayLeft dataset. How many places are shown by these images? Supposedly there is no single true answer, this has to be considered in the typical evaluation based on precision-recall curves. GardensPoint image credits: Arren Glover.

the path. In contrast to the Nordland datasets, this enables the evaluation of day-night cycles and the influence of an additional lateral viewpoint change that in particular challenges holistic approaches. Fig. 5.5 shows example images. Each column shows the same scene from all three traversals. The severe appearance change due to the day-night cycle can be seen in the first and second row. The additional lateral shift can be seen between the second and the third row.

In the Gardens Point dataset, images showing the same place are indicated by their index in the dataset (the number in the file name). In particular, there is exactly one image for each traversal and place. Fig. 5.6 illustrates an additional problem during evaluation of all datasets: Which images of such a sequence are considered to show the *same* place? For scale invariant descriptors all images may appear quite similar. In the later presented experiments, this will either be considered by definition of the ground-truth matchings (e.g. to allow matching images with distance <3 frames in the sequence) or by selecting a subset of distinct places.

Fig. 5.7.: Example images of eight places from the Alderley Dataset. Each image pair shows the same place at daytime and during a rainy night. Alderley image credits: [Milford and Wyeth, 2012].

5.1.3. The Alderley dataset - From sunny day to rainy night

The Alderley dataset by Milford and Wyeth [2012] provides two image sequences captured from a car driving through the suburb of Alderley in Brisbane, Australia. The first sequence shows the traversal of the 8 km route on a sunny day, the second during a stormy night with heavy rain and low visibility. The videos were captured using a Panasonic Lumix DMC-TZ7 digital snapshot camera with a frame rate of 50 frames per second and were subsequently cropped, resized to (640×260) pixels and manually time aligned.

It was first presented in [Milford and Wyeth, 2012] and is available online[4]. Fig. 5.7 shows example images illustrating the severe appearance changes. These images nicely demonstrate why approaches that can match images in static environments fail for these images. E.g. think of the chance of a typical keypoint detector or descriptor to find corresponding regions and to match them. Nevertheless, the results using the novel superpixel based approach presented in chapter 7 demonstrate that it is not beyond all hope.

5.2. Related work and own contributions to the field

The datasets presented in the previous section showed that changing environments extend the problems of basic scene recognition to recognition of places that are subject to severe appearance changes due to changing illumination conditions, weather, day-night cycles or seasonal changes. This can be considered the more general problem compared to place recognition in non changing, static environments. However, baseline and foun-

[4] `https://wiki.qut.edu.au/pages/viewpage.action?pageId=181178395` [Last accessed: 2015-08-14]

dation for solving the more complex problem are established solutions that showed to handle the simpler static case.

As illustrated in Fig. 5.1, for *non changing* environments, place recognition has been approached by local and holistic image features. The set of available established holistic image descriptors include GIST [Torralba et al., 2003] and BRIEF-GIST [Sünderhauf and Protzel, 2011]. Both compute a single, global feature vector for the complete image. GIST is built upon image filtering and BRIEF-GIST uses simple binary comparisons similar to the BRIEF keypoint descriptor [Calonder et al., 2010]. In terms of local image features, the combination of interest point detectors and hand engineered descriptors for the surrounding image patch showed to be useful in the context of localization. For the matching of local features in successive video frames or rectified stereo image pairs, correlation based measures on the image patch are often sufficient [Szeliski, 2010]. For the matching of images from more varying viewpoints, more sophisticated descriptors like the Scale Invariant Feature Transform (SIFT) [Lowe, 1999] can provide a large benefit [Mikolajczyk and Schmid, 2005]. The success of SIFT inspired a lot of similar approaches, including PCA-SIFT [Ke and Sukthankar, 2004] and SURF [Bay et al., 2008] (section 6.2 will give more details on these descriptors). Cummins and Newman [2008] presented FAB-MAP, that uses SURF features in a bag-of-words approach for place recognition (more detail on visual vocabularies will be given in section 6.3). The current version FAB-MAP 2.0 [Cummins and Newman, 2011] can be considered state of the art for large scale visual place recognition in fairly static environments. However, FAB-MAP is not designed for changing environments resulting in poor performance in the presence of severe appearance changes [Milford and Wyeth, 2012].

Finding solutions that perform well in changing environments is an active field of research. The related work can be arranged on two dimensions. The first being the *time* dimension and the second being the *approach* to handle the changing environments. In this second dimension, I want to distinguish the following approaches:

1. Using powerful standard localization techniques (and hope for the best)
2. Increasing robustness by matching image sequences
3. Using seasonally invariant types of features or sensors
4. Accept that the places cannot be matched and organize the different appearances in the map
5. Attempt to learn or reason about systematic changes

Both dimensions are correlated since at different points in time, new ideas arose and different approaches were popular. Fig. 5.8 illustrates both dimensions - the milestones arranged on the timeline include colour codes for the type of approach that is used. In the following, I want to give a short overview of attempts in each direction. This extends the related work presented in [Neubert et al., 2015b] and roughly reproduces the chronological order.

Using powerful standard localization techniques (and hope for the best) - Part I

Based on local keypoint features, [Valgren and Lilienthal, 2007, 2010] show high recognition rates on single image matching of five places across seasons and global topological localization on the same dataset. They compare SIFT and SURF variants on omnidirecional images. They found U-SURF to perform best in this comparison and conclude that high-resolution omnidirectional images and additional constraints on the matched keypoints (epipolar geometry and reciprocal matchings) are necessary. Unfortunately, it remains unclear what portion of matchings are on seasonally invariant objects (like building facades) and how this approach generalizes to larger datasets.

Interest point based approaches in changing environments have to rely on the detection of keypoints on objects that vary strongly in their appearance. To overcome this shortcoming, a fixed distribution of points can be used (e.g. a keypoint grid). Sift Flow [Liu et al., 2008] computes a SIFT descriptor at each pixel and matching is based on local and global constraints. While they show impressive results on scene alignment under strongly varying conditions, this approach has not yet been used for across season place recognition. Glover et al. [2010] present a combination of the sophisticated local feature recognition system FAB-MAP [Cummins and Newman, 2008] and the biologically inspired SLAM approach RatSLAM [Milford et al., 2004; Milford and Wyeth, 2010] based on pose cell filtering and experience mapping. RatSLAM is robust to false-positive loop closures from the image processing front-end and integrates matching information over time. The hybrid FAB-MAP + RatSLAM system has shown that mapping in challenging outdoor conditions with variances in illumination across different times of day is possible. However, the authors conclude that the SURF features on which it is based, are too variable under those varying conditions to form a truly reusable map.

Increasing robustness by matching image sequences

The pose cell filtering of RatSLAM is a step towards using sequences for matching. In their subsequent work, the RatSLAM group presented two further approaches for matching image sequences: CAT-SLAM and SeqSLAM. Continuous Appearance-based Trajectory SLAM (CAT-SLAM) [Maddern et al., 2012] uses a particle filter with particle weighting based on local appearance and odometric similarity. They were able to outperform FAB-MAP on the New College vision and laser data set. While this dataset does not include environmental changes, Lowry et al. [2014b] use a combination of the underlying CAT-graph and a probabilistic whole image matching framework for place recognition in changing environments.

The second approach, SeqSLAM [Milford and Wyeth, 2012], builds upon a lightweight visual matching front-end and explicitly matches local sequences of images. They show impressive results on matching challenging scenes across seasons, time of day and weather conditions. Although their system is limited to constant velocity motion and the holistic visual front-end is sensitive to viewpoint changes, if these conditions are met, it provides state-of-the-art results under changing conditions. Moreover, it is one of the very few

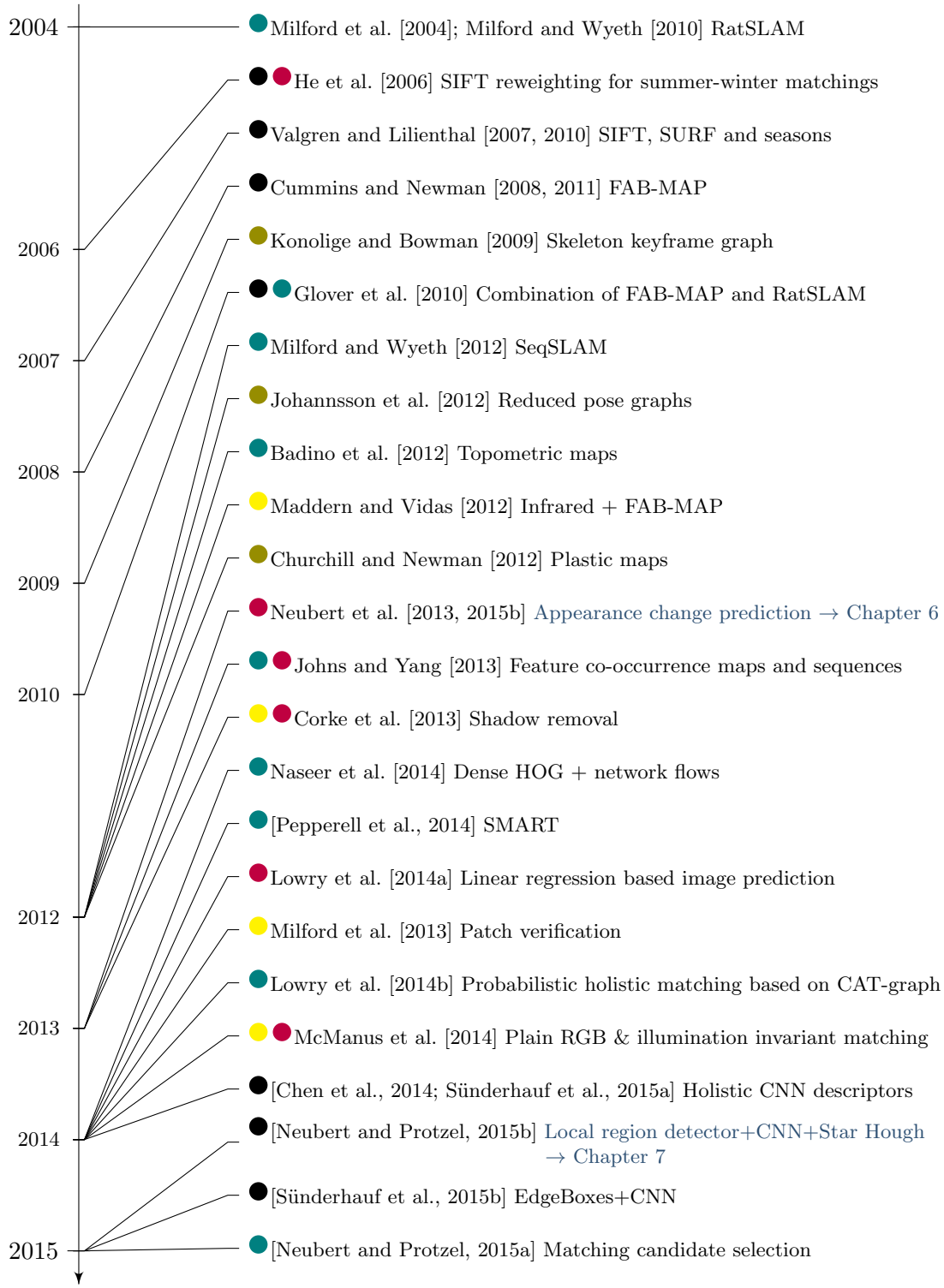


Fig. 5.8.: Timeline of place recognition approaches for changing environments. The coloured dots indicate the area(s) of contributions: ●using standard descriptors, ●exploiting sequences, ●managing the different appearances, ●using seasonal invariant sensors or features, ●learning or reasoning about changes.

systems for which an implementation is available[5].

Badino et al. [2012] implement the idea of visual sequence matching using a single SURF descriptor per image (WI-SURF) and Bayesian filtering on a topometric map. They show real-time localization on several 8 km tracks recorded at different seasons, times of day and illumination conditions. Johns and Yang [2013] quantise local features in both feature and image space to obtain discriminative statistics on the co-occurrences of features at different times of the day. They combine their approach with a sequence matching that can also handle non-zero acceleration. In contrast to the holistic visual front-end of SeqSLAM, the usage of local features improves the robustness towards viewpoint changes. They show results on a 20 km track over the course of one day, where their method outperforms FAB-MAP and SeqSLAM.

Sequence Matching Across Route Traversals (SMART) [Pepperell et al., 2014] is another approach to extend the robustness of sequence matching towards varying view points and differences of the speed along the camera trajectories. They include a variable offset image matching to increase robustness against viewpoint changes and sample images at constant trajectory intervals, in contrast to constant time intervals, to handle varying speed between the two traversals of the same route. Therefore, a source of translational velocity is necessary. In [Pepperell et al., 2014], they used wheel encoders of the cars on which the cameras were mounted.

Using a graph theoretical approach, Naseer et al. [2014] formulate image matching as a minimum cost flow problem in a data association graph. For computation of image similarities, they use a dense grid of HOG descriptors [Dalal and Triggs, 2005] and generate multiple route matching hypotheses by optimizing network flows. They show competitive results to SeqSLAM.

While most of the sequence based approaches exploit the grouping of images to reduce the number of false positive matchings, the grouping can also be used to select promising matching candidates. Very recently, in [Neubert and Protzel, 2015a], we proposed a method that exploits similarities between images inside the given database (that are typically not affected by environmental changes) to find matching candidates for the current query image (that might be subject to severe appearance changes). Preliminary result showed that in many configurations, the number of comparisons can be reduced by about 90 % without considerable influence on the place recognition performance.

In summary, using groups of images can greatly improve the place recognition performance. When using sequences, special care has to be taken on the length of the sequences (to allow for different trajectories) and the ability to handle varying velocities. The novel approaches presented in chapters 6 and 7 are single image based. However, they can both be combined with the sequence matching techniques presented above - in particular section 6 includes an evaluation in combination with SeqSLAM.

Using seasonally invariant types of features or sensors

Maddern and Vidas [2012] combine visible and long-wave infrared imaging for place

[5] `https://openslam.org/openseqslam.html`

recognition through a day-night-cycle. Their system is based on FAB-MAP and combines words of SURF features from the visible and infrared images (using two separate vocabularies). They find the combination of both modalities to give the best results: infrared is more robust to extreme changes while the visible modality provides better recall during the day. They present preliminary results on data of a 1.5 km track traversed several times during a single day-night cycle.

He et al. [2006] learn an intermediate representation of images such that the distance of two images in this intermediate representation reflects the distance between the places in the world, where these images were taken. The intermediate representation is a vector of weighted SIFT feature prototypes. Since they train their system on summer and winter images, they search for a set of SIFT features that are suitable for place recognition under this seasonal change. Their approach also relies on the extraction of local keypoints on the same world object under the seasonal change.

Zhang and Kosecka [2005] focus on recognizing buildings in images. They use a hierarchical matching scheme based on localised colour histograms and SIFT features to search for buildings in an image database. While they did not explicitly design their system for place recognition across seasons, their test data (ZuBud) covers different weather conditions and seasons. Thus, building facades could serve as seasonally invariant landmarks.

Robustness to environmental changes can also be addressed based on image preprocessing: The heavy image normalization involved in the visual front end of SeqSLAM is an important ingredient for its robustness towards appearance changes. Milford et al. [2013] present another approach that involves the same patch normalization procedure but uses single images for place recognition (in contrast to the sequences used in SeqSLAM). To decrease the number of false positive place matchings, they apply a patch verification step on a set of candidate matchings. The candidates are obtained from a holistic image comparison using the sum of absolute differences on patch normalized images - similar to the SeqSLAM front end. They show impressive results on the Alderley dataset comprising images at daytime and during a rainy night. However, the candidates are selected based on a holistic image comparison and for the patch verification, only small variations of the position of compared patches are considered. Therefore, this approach is supposed to be sensitive to viewpoint changes.

Accepting that the places cannot be matched and handling the different appearances in the map

Changing environments are challenging for visual place recognition systems. But they are also a challenge for the mapping side of the problem. Dayoub and Duckett [2008] approach the problem of changing environments using long-term and short-term memory concepts. They update the representation of places in the world incrementally, by gradually adding information about current features and removing information about outdated features. Churchill and Newman [2012] present a mapping system based on a plastic map, a composite representation of multiple experiences connected in a relative framework. Each experience handles a sequence of images, motion and 3D feature data. Multiple localizers match the current frame to stored experiences. Several experiences

can be active at once, when they represent the same place. The complexity of the plastic map varies according to the amount of variation in the scene. They present results in changing lighting and weather conditions over a three month period.

For pose graph SLAM, Biber and Duckett [2009] showed that the map grows unbounded in time, even for small environments that are repeatedly traversed. Johannsson et al. [2012] proposed the reduced pose graph that reuses already existing poses in previously mapped areas and incorporates new measurements as new constraints between existing poses. This can be used if the place recognition front end can match the poses. Konolige and Bowman [2009] present a mapping system based on a skeleton graph of keyframes from a visual odometry system. Views of keyframes are updated and deleted to preserve view diversity while limiting their number. They showed their system to handle changing illumination conditions in an office environment.

However, being able to associate places despite severe changes in their appearance is advantageous to the mapping process since the rate at which new experiences [Churchill and Newman, 2012], poses [Johannsson et al., 2012], or views [Konolige and Bowman, 2009] have to be introduced to the map can be reduced.

Learning or reasoning about systematic changes

Using (1) powerful descriptors that can recognize places despite severe appearance changes or (2) accepting that those places can not be matched and the different appearances have to be organized in the map, are two very different approaches to place recognition in changing environments. In [Neubert et al., 2013] we present a somehow orthogonal approach by learning to *predict* systematic appearance changes from training data. This approach is based on superpixel vocabularies and will be the topic of chapter 6. However, except for the learning of weights for SIFT matchings in [He et al., 2006], it has been one of the first approaches to involve learning in place recognition in changing environments. Since then, several other learning based approaches have been presented.

For example, the sequence based method presented in [Johns and Yang, 2013] also involves learned feature co-occurrence maps. Ranganathan et al. [2013] show the benefit of learning illumination invariance in terms of local image feature matching. They model lighting variations using a probability distribution on a discretised descriptor space. Lowry et al. [2014a] propose to use linear regression techniques to predict images across different times of day. Based on training images showing the same scene at morning and afternoon, they learn a linear whole image transformation that can be applied to images from one condition to create a predicted image at the other condition. To cope with different scene types, they also propose to cluster similar images based on k-means before learning a transformation for this scene type. Query images are then first associated to a scene type and the corresponding transformation is applied. In their experiments, using the predicted images results in higher place recognition performance with an additional benefit from distinguishing different scene types.

In her thesis, Lowry [2014] also presents several other approaches to place recognition in changing environments. In terms of modelling appearance changes, one of the presented approach uses principal component analysis and builds upon the assumption

that the condition dependent information are encoded in the first principal components. This assumption is in accordance with results, presented in [Jacobs et al., 2007], on the principal components of webcam images captured over large periods of time. If these information of the first components are removed, the remaining image is only location specific and can be used for place recognition.

The understanding how the appearance of visual features can change over time, can either be obtained by learning, or by a model derived from natural and in particular physical properties of the world. For example, the illumination changes that appear in different images of the same place have a clear physical foundation. In particular, shadows are supposed to significantly change over the course of a day. Corke et al. [2013] evaluate the benefit from removing shadows in images before place recognition. Exploiting a number of simplifying assumptions on the light source and the used sensor, a colour image pixel can be mapped to a one dimensional pixel value which is a function of only material reflectance properties. Although these assumptions do not perfectly hold in real world applications, they show significant place recognition improvement using the framework presented in [Churchill and Newman, 2012]. This insight was later used by Upcroft et al. [2014] for the classification of urban streets and extended to a localization system based on the combination of RGB and illumination invariant matching pipelines in [McManus et al., 2014].

Using powerful standard localization techniques - Part II

The first class of approaches that uses available powerful image descriptors *directly* for place recognition, currently experiences a revival with the rise of multi purpose image features based on Convolutional Neural Networks (CNN). In recent years, CNN based image descriptors showed impressive performance on a variety of computer vision tasks, like image classification [Krizhevsky et al., 2012], object detection [Girshick et al., 2014; Chatfield et al., 2014] and other recognition tasks [Razavian et al., 2014].

Chen et al. [2014] and Sünderhauf et al. [2015a] proposed to use descriptors obtained from CNNs for place recognition in changing environments. Sünderhauf et al. [2015a] obtained image descriptors from the stacked output of a single CNN layer. They evaluated different layers and found the first convolutional layers to be the most robust against image changes, but sensitive to viewpoint changes. These descriptors showed impressive performance on a set of challenging datasets, including the cross-seasonal Nordland dataset. For single image matching of images with similar viewpoint, the CNN based holistic descriptor can be considered as state of the art - in particular since there are ready to use networks available enabling the inclusion of this approach as baseline in new experiments.

However, the used CNN descriptors are holistic representations and sensitive to viewpoint changes. Therefore, in [Neubert and Protzel, 2015b] we propose to use local region detectors in combination with CNN descriptors. We evaluate different types of region detectors including keypoints, superpixel based approaches and object proposal methods. We further incorporate the landmarks' spatial arrangement in the image matching using a novel star graph model and Hough voting based approach. This novel approach

will be presented in section 7.

Congruently and independently from our approach, [Sünderhauf et al., 2015b] proposed to use CNN descriptors in combination with a particular object proposal method, EdgeBoxes [Zitnick and Dollar, 2014]. They show impressive place recognition results on several datasets including severe appearance changes. However, they do not incorporate the spatial arrangement of landmarks and use a single region detector. The experimental comparison of the components of these two CNN-landmark based approaches ([Neubert and Protzel, 2015b] and [Sünderhauf et al., 2015b]) in section 7.7 will show the benefit of the incorporation of the spatial arrangement and the variety of local features that are used in our approach.

Summary

The timeline shown in Fig. 5.8 provides two general insights: (1) Place recognition in changing environments is an active field of research. In particular since 2012 a lot of approaches have been presented. (2) Even in the covered short period of time, different approaches were popular at different times: After the usage of available image descriptors like SIFT or SURF, exploiting sequence became popular. Starting in 2013, several learning based approaches were developed. My own contribution to this development and one of the first approaches of this kind, "Appearance Change Prediction" [Neubert et al., 2013, 2015b], will be presented in detail in chapter 6.

At this point in time (before 2015), many different approaches existed - and (at least) two major issues:

1. None of the existing methods showed to be robust against severe appearance changes *and* viewpoint changes. The latter is due to the fact that the most successful recognition methods for changing environments avoid the local feature detection step and use holistic image descriptors.
2. Only very few methods are readily available for comparison to new approaches, namely FAB-MAP and SeqSLAM. This hampers the measuring of progress in this field.

The topic of chapter 7 is the novel CNN-landmark based approach briefly introduced above [Neubert and Protzel, 2015b]. This approach, in particular, addresses the first of the above issues since it is robust to appearance changes *and* viewpoint changes. The second issue is addressed by a publicly available implementation (cf. section 1.3) and by the fact that no dataset specific training is involved which enables the straight forward application to new datasets. The consideration of these two issues together with the overall performance of the proposed system is supposed to facilitate progress towards real world application of algorithms for place recognition in changing environments.

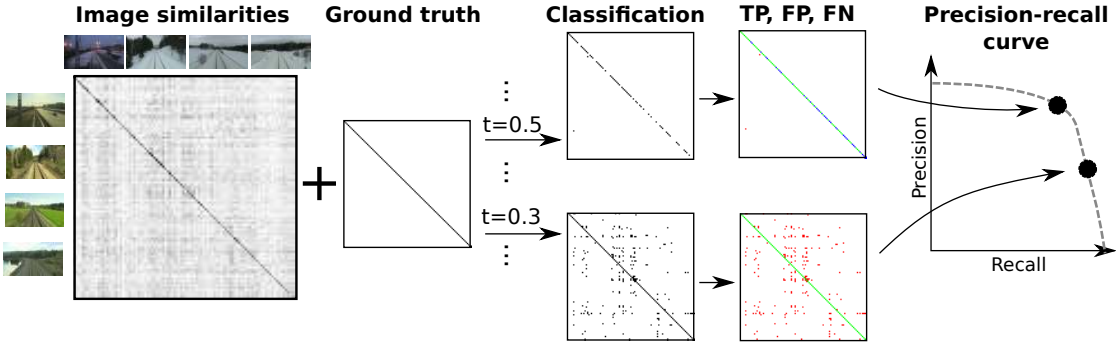


Fig. 5.9.: Creation of precision-recall curves. Input are image similarities illustrated by the similarity matrix (dark values indicate high similarity) and knowledge about the ground truth matchings (e.g. a diagonal matrix as it is the case for the Nordland dataset). Each value of the similarity matrix is classified to be a matching or not based on a threshold t. The comparison to the ground truth is used to obtain TP (green), FP (red) and FN (blue). Each threshold t yields a point on the precision-recall curve (t=0.5 and t=0.3 are shown).

5.3. Using precision-recall curves for evaluation of place recognition experiments

Precision and recall were already used for evaluation of superpixel segmentations in chapter 3. However, they are an established method for evaluation of place recognition experiments as well, e.g. [Milford and Wyeth, 2012]. As a reminder, given true positives (TP), false positives (FP) and false negatives (FN) of a classification experiment, they are computed as follows:

$$\text{Precision: } P = \frac{TP}{TP + FP}$$

$$\text{Recall: } R = \frac{TP}{TP + FN}$$

Precision is the proportion of the found true place recognitions on all recognitions. Recall is the proportion of the found true place recognitions on all existing true matchings. To use them for evaluation of place recognition experiments, we have to define the values TP, FP and FN.

This process is illustrated in Fig. 5.9. The result of a typical visual place recognition experiment is a set of image similarities. In the most simple case, two sets of images are given: database images and query images, and each image of the query set is compared to all images of the database set. However, in alternative setups, the query images may be iteratively inserted in the database set. In this case, it is also possible to start with an empty database set. However, the resulting set of image similarities can be illustrated as a (partially filled) similarity matrix. For evaluation based on precision-recall curves, a set of classification tasks on these similarities is conducted.

In the experiments presented in the following chapters, the performance evaluation is based on ground truth knowledge about image pairings showing the same place and those that show different places. Given the similarities between all possible image matchings, they are divided into matchings and non-matchings based on a threshold t. All image matchings that correspond to a ground truth place correspondence are counted as true positives, all matchings that do not show the same place according to the ground truth are considered false positives, and false negatives are all image pairings of the ground truth that are not in the set of matchings. From these three values, a point on the precision-recall curve is computed. To obtain the curve, the threshold t is varied.

However, there are alternative methods to obtain the precision-recall curves - with a considerable influence on the shape of the curves. An example will be shown in section 7.7.4, where an alternative evaluation method is used in addition to the here presented method to demonstrate the difference. Moreover, some place recognition algorithms already select the *single* best matching for each place. This poses an upper bound on the number of possible false positives - there can only be a single false positive for each image of the query database. Although it increases the performance in the resulting curves, this should not be used in a practical application, since it prevents the recognition of multiply revisited places. Therefore, it is not used in this thesis - except for comparisons using SeqSLAM. There the reduction to a single matching is part of the processing chain.

Dream about the pictures that I play of changes
Green leaves of summer turn red in the fall
To brown and to yellow, they fade

Phil Ochs, from the song "Changes", 1973

6 Superpixel-based Appearance Change Prediction

Section 5.2 provided an overview of existing approaches on place recognition in changing environments. In the year 2013, the two dominant directions for dealing with changing environments were either to try to find image descriptions that can match places despite the severely changed appearances, or to deal with the fact that the places can not always be matched. In previous work in [Neubert et al., 2013], [Sünderhauf et al., 2013] and [Neubert et al., 2015b] we[1] presented an approach that is somehow orthogonal to these two directions and that is the topic of this chapter: Superpixel-based appearance change prediction (SP-ACP).

6.1. The concept of appearance change prediction for long term localization

What current approaches to place recognition (and environmental perception in general) lack, is the ability to *reason* about the occurring changes in the environment. Most approaches try to merely *cope* with them by developing change-invariant descriptors or matching methods. Potentially more promising is to develop a system that can *learn* to

[1] While the underlying idea and the implementation are my own contributions, Niko Sünderhauf had large influence on the work presented in this chapter 6 of the thesis. This includes theoretical contributions in form of discussions and practical contributions in particular for the preparation of the Nordland dataset, its description and the evaluation based on Open FAB-MAP and SeqSLAM.
The sections 6.1, 6.4 and 6.5 including Fig. 6.1 and 6.9-6.18 are published in very similar form in Robotics and Autonomous Systems, 69, Peer Neubert, Niko Sünderhauf, Peter Protzel, Superpixel-based Appearance Change Prediction for Long-Term Navigation Across Seasons, Pages 15-27, Copyright (2015), with permission from Elsevier. `http://dx.doi.org/10.1016%2Fj.robot.2014.08.005`

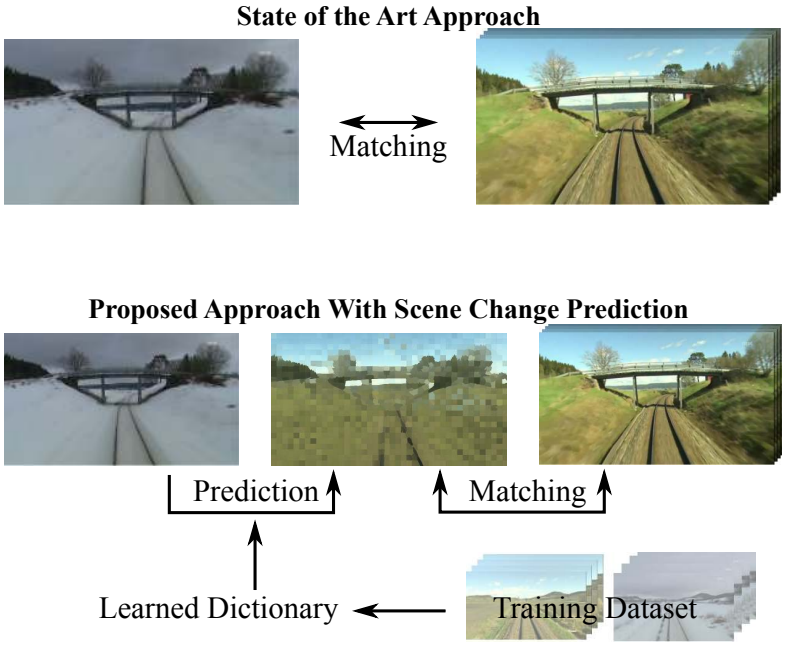


Fig. 6.1.: State of the art approaches to place recognition will attempt to directly match two scenes even if they have been observed under extremely different environmental conditions. This is prone to error and leads to bad recognition results. Instead, we propose to *predict* how the query scene (the winter image) would appear under the same environmental conditions as the database images (summer). This prediction process uses a dictionary that exploits the systematic nature of the seasonal changes and is learned from training data.

predict certain systematic changes (e.g. day-night cycles, weather and seasonal effects, re-occurring patterns in environments where robots interact with humans) and to infer further information from these changes. Doing so without being forced to explicitly know about the *semantics* of objects in the environment is the topic of this chapter.

Fig. 6.1 illustrates the core idea of our work and how it compares to the current state of the art place recognition algorithms. Suppose a robot re-visits a place under extremely different environmental conditions. For example, an environment was first experienced in summer and is later re-visited in winter time. Most certainly, the visual appearance has undergone extreme changes. Despite that, state of the art approaches would attempt to match the currently observed winter image against the stored summer images.

Instead, we propose to *predict* how the current scene would appear under the same environmental conditions as the stored past representations, before attempting to match it against the database. That is, when we attempt to match against a database of summer images but are in winter time now, we predict how the currently observed winter scene would appear in summer time or vice versa.

The result of this prediction process is a synthesized summer image that preserves the

structure of the original scene and is close in appearance to the corresponding original summer scene. This prediction can be understood as *translating* the image from a winter vocabulary into a summer vocabulary or from winter language into summer language. As is the case with translations of speech or written text, some details will be lost in the process, but the overall *idea*, i.e. the gist of the scene will be preserved. Sticking to the analogy, the error rate of a translator will drop with experience. The same can be expected of our proposed system: It is dependent on training data, and the more and the better training data it gets, the better can it learn to predict how a scene changes over time or even across seasons.

Contribution 6.1 *We propose to incorporate an additional prediction step before attempting to match an image to a database of images captured under different conditions, e.g. a different season, weather or time of day. This prediction is supposed to be learned from training data.*

6.1.1. Related Work: Correspondence to the texture transfer and image analogy problems

The idea to predict images from training examples has some relations to two other image processing tasks:

- *The texture transfer problem* [Efros and Freeman, 2001]: Given two images A_S, A_W and a correspondence map C that relates parts of A_S to parts of A_W, synthesize the first image with the texture of the second. C typically depends on image intensity, color, local image orientation or other derived quantities.
- *The image analogy problem* [Hertzmann et al., 2001]: Given an image pair (A_S, A_W) and a query image B_S, compute a new "analogous" image B_W that relates to B_S in the same way as A_W to A_S.

Speaking in the context of predicting image change across seasons: A_S, A_W are given summer (S) and winter (W) training images and we learn to synthesize a new winter image B_W given a new summer image B_S or vice versa. The approaches of Efros and Freeman [2001] and Hertzmann et al. [2001] create visually appealing results but have not yet been used in the context of place recognition. They focus on using single image pairs instead of large collections of training data. Nevertheless, such approaches could be used to improve the visual coherence of the images predicted by the proposed prediction framework.

6.1.2. Overview of the main components

How can the severe changes in appearance a landscape undergoes between winter and summer be learned and predicted? The underlying idea of our approach is that the appearance change of the whole image is the result of the appearance change of its parts - i.e. its superpixels. If we had an idea of the behaviour of each part, we could

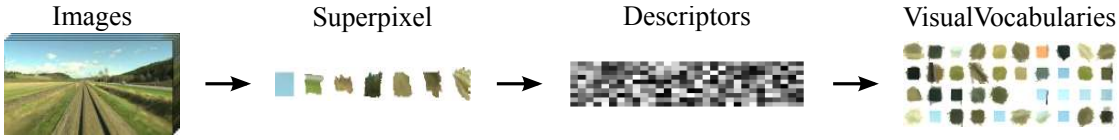


Fig. 6.2.: The key ingredients involved in the novel Appearance Change Prediction approach.

predict the whole image. Instead of trying to recover semantic information about the image parts and model their behaviour explicitly, we make the assumption that similarly appearing parts change their appearance in a similar way. While this is for sure not always true, it seems to hold for many practical situations (e.g. changing colour of the sky from sunny day light to dawn, appearance of a meadow in summer to its snow-covered winter counterpart). This idea can be extended to groups of parts, incorporating their mutual relationships.

Before diving deeper into the details of the proposed algorithmic approach in section 6.4, we want to introduce the main components shown in Fig. 6.2. Starting from a set of input images, we use superpixel segmentations to obtain the parts of the image for whose we attempt to learn the appearance changes. Superpixel segmentations were the topic of the first part of this thesis. The remaining two main ingredients, descriptors for superpixels and visual vocabularies, will be presented in the following two sections 6.2 and 6.3.

The following Section 6.2 will present the concept of image based descriptors and the particular challenges for describing superpixels in a very general context. Section 6.3 will introduce visual vocabularies and evaluate a property of superpixel descriptors that is closer related to the proposed Appearance Change Prediction task: What amount of semantic meaning is provided by a superpixel word? This is relevant since the proposed approach builds on the assumption that similar superpixel words undergo similar appearance changes during the environmental changes - an assumption related to the amount of common semantics of the words.

6.2. Superpixel descriptors

The first part of this thesis provided criteria and metrics to select an appropriate superpixel segmentation method for the task at hand. While neither the development of new descriptors nor the extensive comparison of existing descriptors are in the particular scope of this thesis, the descriptors are an important component of the proposed SP-ACP approach. Moreover, there are particular challenges when describing superpixels in contrast to typical interest points.

Therefore, this section 6.2 will first give a gentle introduction to descriptors and commonly used approaches. The particular challenges in the context of superpixels are the topic of section 6.2.2. The overlap criterion presented in section 6.2.3 is a novel approach to evaluating the performance of descriptors for superpixels. It measures how well such descriptors can distinguish to what extent two superpixels show the same part of the

world. It is also published in [Neubert and Protzel, 2015a]. The there described metric evaluates a very general property of superpixel descriptors. However, section 6.2.4 will present some results of commonly used descriptors to pave the way for the subsequent evaluation of superpixel descriptors in the context of visual vocabularies in section 6.3.

6.2.1. A gentle introduction to descriptors

In general, a *descriptor* is a collection of properties or attributes derived from an image or a local region of an image such that the shown part of the world can (1) be distinguished from other parts, and (2) the properties are invariant (or at least robust) to changes during image acquisition like different scales, orientations, or viewpoints. Descriptors can be used for identification of an individual aspect of the world (e.g. tracking an object in a video sequence) or for recognizing instances of a group with common properties that are also reflected in the descriptors (e.g. assigning a class label). For practical reasons, in image processing, this set of attributes is typically given in form of numerical values and combined to a vector, matrix or arbitrary tensor describing this image part.

While it is often distinguished between *global* (or holistic) descriptors for the whole image and *local* descriptors that are computed for a defined sub region of the image, both are based on similar underlying concepts and struggle with similar challenges. For local features, the first step is the detection of a region that can be repeatedly found in other images of the same scene, maybe seen from a different point of view or under varying illumination. The second step is the description of this local region with a set of attributes that is constant under the expected variations of the local region (e.g. affine or other geometrical transformations or changed illumination).

Dependent on the application and the expected amount of variation in the images, different descriptors are preferable [Szeliski, 2010]. In case of minor motion between the located regions, simple measures like the sum of squared differences or normalized cross-correlation of image patches can be used, e.g. for subsequent video frames [Shi and Tomasi, 1993] or rectified stereo image pairs [Zhang et al., 1995]. These patch based descriptors are sensitive to geometrical changes. Estimating an orientation and scale of the local features and using them to resample the patch before computing the descriptor is usually preferable [Szeliski, 2010]. In case of colour images, the statistics of the colour are often used as well. For example in form of first and second order moments [Barnard et al., 2003] or histograms [Gu et al., 2009], sometimes in combination with other descriptors [Tighe and Lazebnik, 2010].

Lowe [1999] proposed an influential approach to feature description: Scale invariant feature transform (SIFT). Basis for the SIFT descriptor are the gradients at each pixel in a 16×16 window. These gradients are represented by orientation histograms with 8 orientations, one histogram for each 4×4 quadrant. To increase robustness, several additional steps are processed: The gradient magnitudes are weighted by a Gaussian to reduce the influence of off-centre gradients since these are supposed to be more affected by small misregistrations. Further, the original 256 weighted gradient magnitudes are softly added to $(2 \times 2 \times 2)$ histogram bins using trilinear interpolation. Finally, the resulting 128 dimensional raw SIFT vector is normalized in three steps: (1) The vector

is normalized to unit length. (2) All values are clipped at value 0.2. And (3), the clipped vector is normalized to unit length again.

SIFT is an intensively *engineered* image descriptor which had important influence on many computer vision applications, e.g. object recognition [Lowe, 2004], image retrieval [Sivic and Zisserman, 2003] and mobile robot navigation [Barfoot, 2005]. It also paved the way for a multitude of similar descriptors. For example, Speeded-Up Robust Features (SURF) [Bay et al., 2008] use Haar features and integral images to approximate and speed up the computations. Ke and Sukthankar [2004] propose to use the (x, y) gradients over a 39×39 patch directly and reduce the resulting $2 \cdot 39 \cdot 39 = 3042$ dimensional vector to 36 dimensions using principal component analysis, hence the name of the descriptor: PCA-SIFT. Mikolajczyk and Schmid [2005] compare a subset of the above descriptors and further propose Gradient Location-Orientation Histogram (GLOH), an adaption of SIFT with a log-polar space binning, that performs superior to the existing methods in their evaluation.

In particular in the context of object detection, there exists a wide range of further approaches. For example Histogram of Oriented Gradients (HOG) [Dalal and Triggs, 2005] combines gradient orientations similar to SIFT, but on a dense grid of uniformly spaced cells and with overlapping local contrast normalization. In case of regions with discriminative contours, descriptors like shape context [Belongie and Malik, 2000] can be used. It represents the shape by connecting a boundary point to all other boundary points and collecting the resulting vectors in a normalized log polar histogram. For practical applications, a subset of the boundary points is used.

While the above techniques are hand engineered (except for involved dimensionality reduction techniques), there are also approaches based on learning. For example Lepetit et al. [2004] treat wide baseline point matching as classification problem, each class corresponding to the set of all possible views of such a point. Given a training image, they synthesize a large number of views of individual keypoints and use statistical classification tools to create a compact description of this view set. However, in subsequent work they found that image patches can be effectively classified on the basis of a relatively small number of pairwise intensity comparisons. First, they learned these comparisons by training based on random forests [Lepetit et al., 2005] and later proposed the simplistic Binary Robust Independent Elementary Features (BRIEF) [Calonder et al., 2010] that describe image patches by binary comparisons of a *randomly* chosen set of points of the patch.

Recently, rich descriptors based on deep learning, in particular the output of Convoutional Neural Networks (CNN), demonstrated promising results for many applications, particularly for object recognition, e.g. [Chatfield et al., 2014; Girshick et al., 2014], but also for other recognition tasks [Razavian et al., 2014]. Since they are computationally more complex than the previous descriptors and, dependent on the used network layer, very high dimensional (e.g. 86,528 dimensions for a conv3 layer of the VGG-M network), special care has to be taken for their application. Chapter 7 and in particular section 7.2 will give more details on their usage for place recognition in changing environments.

6.2.2. From keypoint descriptors to recognizing superpixels

The proposed application of superpixel segmentations in the context of place recognition demands descriptors particularly suited for combination with superpixels. Descriptors like SIFT, SURF or BRIEF were designed for combination with *keypoint* detectors like the scale space extrema used in the SIFT detector [Lowe, 2004] or image corners, e.g. [Harris and Stephens, 1988]. The relation between keypoints and superpixels is multifarious: Keypoint descriptors can be involved in the description of superpixels [Tighe and Lazebnik, 2010]. In the other direction, segmentations can be involved in the keypoint detection phase. For example, the Maximally Stable Extremal Regions (MSER) [Matas et al., 2002] are results of a set of segmentations obtained by thresholding an image and searching for regions that are stable for a wide range of thresholds. In another direction, Navarro et al. [2014] propose to postprocess SIFT keypoints with the result of a superpixel segmentation to split the SIFT region into multiple segments and compute a descriptor for each.

While keypoints and superpixels are both image parts, there are fundamental differences which put somehow different requirements on their descriptors. For superpixels, there is less a priori knowledge about their appearance. This contrasts with, for example, keypoints obtained from a corner detector, where the surrounding image gradients can be used for their description. Since the superpixels cover the complete image, there is much higher visual ambiguity than for selected salient keypoints (similar problems occur for dense SIFT approaches). Further, good keypoint detectors provide image locations that can be repeatedly detected in images showing the same scene and that can be accurately localized. For some descriptors, i.e. patch based descriptors, even small deviations of the keypoint location can drop recognition performance significantly [Szeliski, 2010]. This is also one of the mayor challenges when describing superpixels - although there exist superpixel algorithms that are considered stable according to the benchmark in section 3.5, the location of, for example, the midpoint or the centre of gravity of a segment varies much more than that of a typical SIFT or Harris keypoint.

While a complete overview of descriptors and their performance is beyond the scope of this thesis, we want to point out some important steps of their development. Many superpixel descriptors are hand-crafted combinations of large sets of features, sometimes in combination with dimension reduction techniques: E.g. Barnard et al. [2003] use a combination of 40 features: size, position, averages and standard deviations in multiple colour spaces, as well as responses to texture filters. Gould et al. [2008] extend this set to a combination of 83 features, Gu et al. [2009] introduced more advanced contour and edge shape features as well as Lab colour and texton histograms. Tighe and Lazebnik [2010] further extend the feature set by keypoint features (SIFT features and histograms of SIFT words). While there exist various comparisons of keypoint descriptors (e.g. [Mikolajczyk and Schmid, 2005] or [Bauer et al., 2007]), a comparably systematic evaluation of superpixel descriptors doesn't exist.

As already mentioned in the above introduction of section 6.2, descriptors can be used for two types of recognition: (1) A matching-like recognition, where the task is to find exactly the same feature (e.g. for tracking or wide baseline stereo correspondences).

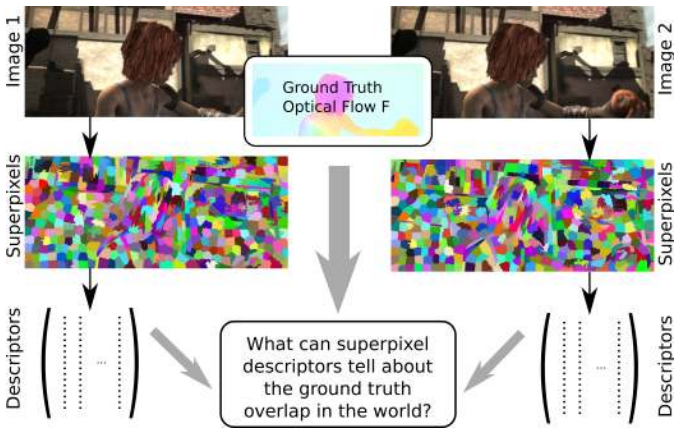


Fig. 6.3.: Given superpixel segmentations of two sequence frames, we can use ground truth optical flow to evaluate superpixel descriptors.

And (2), a classification-like recognition, where the descriptor is used to associate the described image part to a set of groups or to assign a class label (e.g. for object detection). The following section shortly presents our approach to evaluation of the matching-like recognition performance,[2] an evaluation of the classification-like performance will be topic of section 6.3.

6.2.3. What can a descriptor tell about superpixel overlap?

Fig. 6.3 illustrates the proposed approach which is related to the above presented metrics for superpixel segmentation evaluation based on ground truth optical flow. Given pairs of images for which we know the ground truth correspondences of each pixel (the optical flow), we can create superpixels, compute descriptors for them and use the knowledge about the ground truth optical flow to evaluate these descriptors.

A major challenge when comparing superpixels is that they almost never show exactly the same part of the world - they vary in terms of location, size and shape. However, it is an intuitive demand that their descriptors should be more similar the larger the fraction of the world is that is common to both segments. E.g. two segments showing completely different parts of the world should have a high descriptor distance, superpixels that share 70 % of common content should be more similar and segments that share 99 % should have very close descriptors. For evaluating superpixel descriptors we render the above question more precisely:

> *Given two arbitrarily shaped segments from two images that may show to some rate x the same part of the world, what is a good segment descriptor to evaluate this rate x?*

How can this property be concisely evaluated? Let us assume we have a large set of segment pairs for which we can compute any descriptor we want to evaluate, and for

[2]Sections 6.2.3 and 6.2.4 including Fig. 6.3 and 6.4 are published in very similar form in [Neubert and Protzel, 2015a].

which we have the ground truth overlap rate x. We propose to answer this question by running a set of binary classification tasks: How well can the descriptor distinguish whether there is ...

... any overlap between two segments?
... more than 5% overlap?
... more than 10% overlap?
⋮
... more than 95% overlap?
... more than 99% overlap?

To evaluate the performance of a descriptor, we run these 21 binary classification experiments and compute precision recall curves. However, this results in a vast number of curves when comparing several descriptors (21 curves for each compared descriptor). For a condensed illustration, we plot the maximum F-score in each classification experiment over the overlap threshold of this experiment. The resulting curves can be seen in Fig. 6.5.

In detail, the overlap criterion is computed as follows: Given two images I_1, I_2 and their pixel wise associations F (the ground truth optical flow), such that $F(I_1) \sim I_2$. This means that $F(I_1)$ and I_2 are equivalent up to pixels that newly appear in I_2. Let there further be segmentations S_1, S_2 assigning each pixel of I_1, I_2 the unique label of the corresponding segment. We can now compute $F(S_1)$ by applying the ground truth optical flow F on these labels (i.e. move each pixel with its label according to its optical flow vector). This results in two segmentations S_2 and $F(S_1)$ from two different images that are now represented in the same image space of I_2. Now we can easily compute the rate of pixels p that are shared between each pair of segments $s_1 \in S_1$, $s_2 \in S_2$ as the "intersection over union" (IoU), also known as the Jaccard index:

$$O_F(s_1, s_2) = \frac{|F(s_1) \cap s_2|}{|F(s_1) \cup s_2|} \tag{6.1}$$

This constitutes the overlap of two segments from different images given the ground truth optical flow between these images. For each classification problem, we can obtain test data with ground truth from $O_F(s_1, s_2)$.

The proposed overlap criterion should be evaluated on prior-free test data: The binary classification tasks are performed on pairs of segments of images related by ground truth optical flow. The vast majority of all possible pairs have no overlap. This results in a high skewness of the data. More importantly, the prior distribution is different for each binary classification task. The sensitivity of precison-recall curves and F-scores to these changes in the prior distribution would hide other effects: the F score would systematically decrease with growing overlap. Therefore, we enforce constant prior distributions by resampling positive and negative samples for each classification task. A reasonable assumption for the prior in a real application would be to find for each segment of one image the most similar segment in another. In case of 1000 segments per image,

this would result in about 1000 negative samples for each positive sample. Therefore, we resample about one million segment pairs for each image pair keeping a prior distribution of 1:1000 (i.e. we randomly select one million negative and 1000 positive samples).

Contribution 6.2 *The overlap criterion is a novel measure to evaluate the matching performance of superpixel descriptors based on ground truth optical flow.*

6.2.4. Experimental comparison

Similar to the MUSE and MDE metrics of chapter 3, evaluating this metric requires ground truth knowledge of pixel wise associations between image pairs. For this descriptor evaluation, we use the Sintel dataset from the segmentation benchmark, that provides dense ground truth optical flow. We further include another dataset that is well known in the context of keypoint and descriptor evaluation: the affine covariant region dataset (ACR) [Mikolajczyk et al., 2005] available online[3]. Fig. 6.4 shows an example image of the famous Graffiti scene. The dataset comprises eight image sequences, each containing six images which are related by known homographies, that serve as source for the ground truth optical flow. However, this ground truth is not perfect, e.g. due to a not perfectly planar scene. The image sequences are designed for evaluation of the influence of viewpoint changes, rotation, zoom, blur, illumination changes and JPEG compression to keypoint detectors and descriptors. We present combined results of all sequences excluding the JPEG compression sequence since this is not a natural image change.

We include the following set of descriptors in the comparison. This selection is by far not exhaustive, but includes examples for different types of descriptors.

- A native descriptor is a rescaled and contrast normalized **Patch** of the superpixel. We rescale the bounding box around the superpixel to a 10×10 colour patch and blacken the parts that do not belong to the segment.
- **Colour** histograms are an often used descriptor for regions with varying shapes. There exist various colour spaces with different properties, e.g. RGB, CIE Lab or opponency colour space. Based on baseline experiments, we chose an RGB histogram with 30 bins, equally spread over the three channels.
- As descriptor for the segment **Shape**, we compute the mean and standard deviation over the shape context [Belongie and Malik, 2000] of boundary pixels.
- For a **Keypoint** like descriptor, we compute a SURF descriptor [Bay et al., 2008] for each superpixel (parts that do not belong to the segment are blackened).
- We also include a 100 bin histogram of densely sampled SIFT [Lowe, 2004] words in a bag-of-words **BOW** scheme. This showed to be a powerful representation for complex textures [Tighe and Lazebnik, 2010]. More details on visual vocabularies in the context of superpixels can be found in the next section 6.3. The vocabulary was trained on the disjunct BSDS500 training data.

[3] `http://www.robots.ox.ac.uk/~vgg/research/affine/` [Last accessed: 2015-08-14]

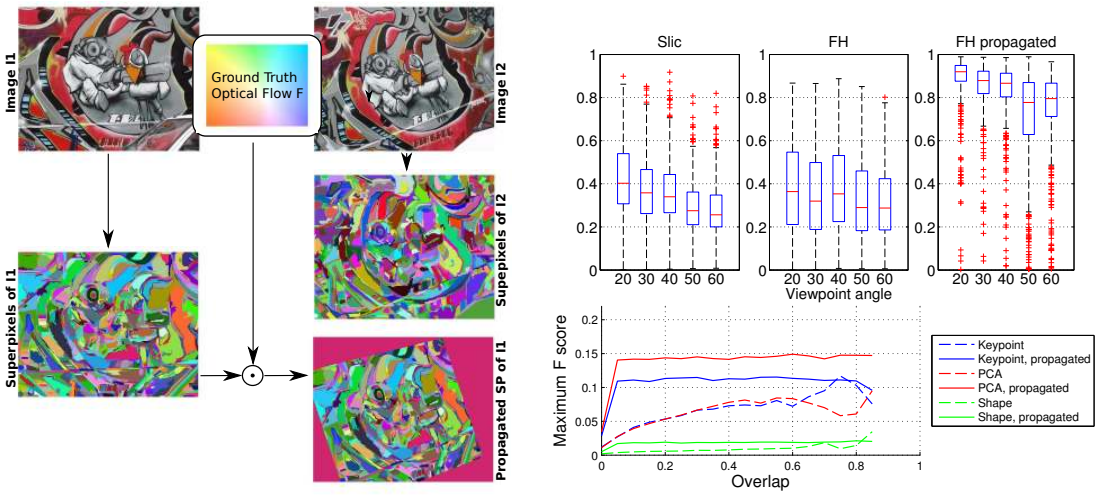


Fig. 6.4.: *(left)* Propagating superpixels scheme. *(right)* The boxplots in the upper right part show the best possible overlap for each segment and different viewpoint changes for individual and propagated segmentations. The curves below illustrate the benefit when using the propagated segments instead of independent segmentations for each image in the ACR "graf" sequence: comparing "Superpixels of I1" with "Superpixels of I2" results in the dashed curves, comparing "Superpixels of I1" with "Propagated SP of I1" results in the solid curves.

- We further evaluate a combination of colour, keypoint, shape and texture features by creating a **PCA** representation of the concatenated RGB, SURF, shape context and BOW features. We segment images and compute descriptors on the BSDS500 training data to obtain a 100 dimensional representation by principal component analysis (PCA).

From the compared superpixel segmentation algorithms of the first part of this thesis, we select FH for an example of an oversegmentation algorithm that produces stable segments and SLIC (more precisely the oriSLIC implementation) for an example of a compact superpixel segmentation algorithm. We create about 1000 superpixels per image. Although, in particular, FH showed to be robust against image changes in the above benchmarks, there are only a few segment pairs with very high overlap in the presence of severe image changes. The boxplots in Fig. 6.4 illustrate the maximum overlap that can be reached for each superpixel for the challenging "graf" image sequence of ACR (viewpoint changes up to 60°). While the average values for SLIC and FH are similar, FH provides more high overlap segments for large viewpoint changes. To identify effects caused by these properties of the segmentation algorithms, we created a further test set with higher overlap: FH propagated. Instead of segmenting both images of the corresponding pair, we segment the first and apply the ground truth optical flow to propagate the segments to the image space of the second frame. This procedure is illustrated in the left part of Fig. 6.4, the descriptors are of course computed on both images.

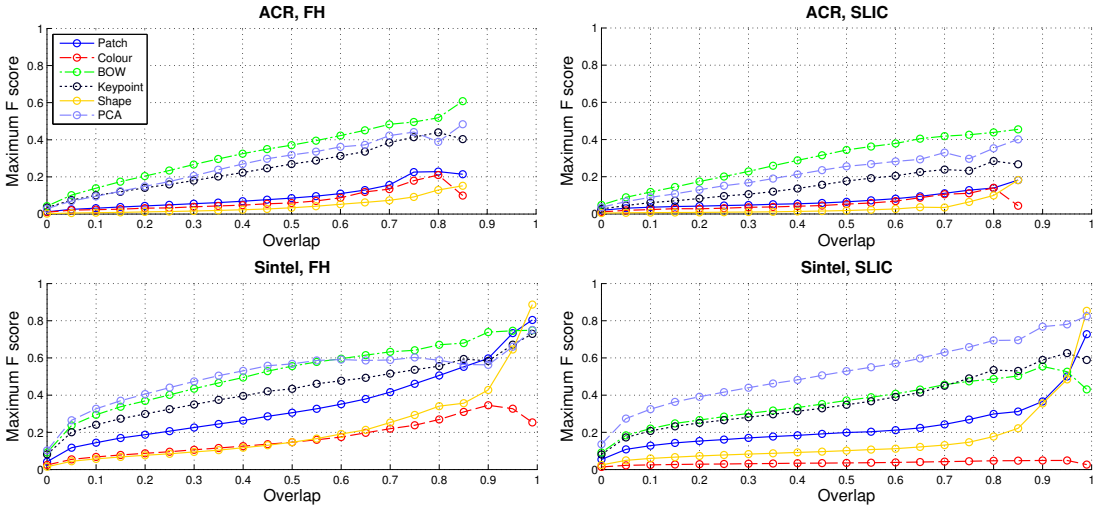


Fig. 6.5.: Results on Overlap criterion for both datasets, using SLIC and FH segmentations. (Higher is better)

Fig. 6.5 shows results of the above superpixel descriptor types on the proposed overlap criterion. As a reminder, each point of these curves illustrates how well the descriptor can decide whether superpixel pairs have at least the corresponding overlap or not. There are two general tendencies: (1) More sophisticated descriptors tend to perform better. (2) The higher the overlap, the easier it is to solve the classification problem. It seems to be very hard for the evaluated descriptors to decide whether there is a small overlap between segments or no overlap. However, this may be related to important generalisation capabilities in tasks like semantic label assignment. The larger the overlap, the more similar is the problem to the keypoint matching problem for which well studied descriptors exist. The plots on the right part of Fig. 6.4 show that particularly the low overlap classification problem could benefit from more stable segmentations. The ACR dataset does not provide enough segments with valid ground truth overlap >85% for a robust evaluation. In contrast, the Sintel dataset provides enough segment pairs with valid high ground truth overlap to see a massive improvement for most of the descriptors at near perfect overlaps. While the Sintel dataset was created in a fundamentally different way, the results are similar. However, the overall performance increased and ranking order of some of the descriptors changed.

Obviously, the comparison presented here is not exhaustive but yielded the insight that it is particularly hard to distinguish segment pairs with small overlap from non overlapping ones and more stable segmentation algorithms could particularly help in these cases. A more detailed evaluation on this rather general evaluation of superpixel descriptors can be found in [Neubert and Protzel, 2015a]. There, we also evaluate a more specific matching task and show preliminary results on including a saliency measure to select interesting superpixels for the matching.

Result 6.1 *In terms of matching superpixels, more sophisticated descriptors like BOW and PCA tend to perform superior. The higher the requested overlap of the superpixels, the easier it is to solve the classification problem. In particular for the low overlap classification task, the descriptors are supposed to suffer from the variation of the superpixel shape between compared images.*

6.3. Superpixel vocabularies

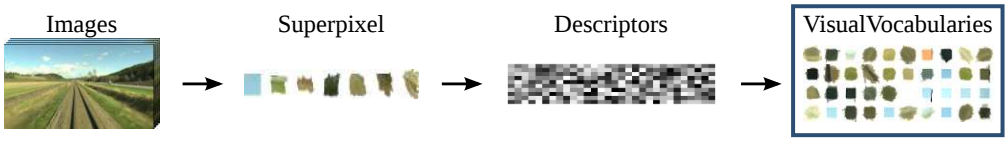


Fig. 6.6.: Visual vocabularies are the final component for the preparation of the SP-ACP approach.

Fig. 6.6 recaps the main ingredients of the proposed SP-ACP system. The previous section introduced available descriptors for superpixels and evaluated them with regard to the rather general question what the descriptor can tell about the superpixel overlap after some modification of the image. In the context of the proposed appearance change prediction system, the descriptors are used to learn a superpixel vocabulary. Therefore, I want to first give a gentle introduction to visual vocabularies, followed by an evaluation of superpixel descriptors in this particular context. The vocabularies are used to predict appearance changes based on the assumption that similar looking parts of the image change similarly. However, the underlying property of this assumption is that for all images, the *same* superpixel words are used to represent objects and parts of the world that change similarly. For example, the same subset of superpixel words are associated to image parts showing trees, and another subset is associated to buildings, and so on. Section 6.3.2 approaches this assumption by evaluating the amount of semantic that is contained in the superpixel words.

6.3.1. A gentle introduction to visual vocabularies

What are visual vocabularies? In natural languages, a *word* can be used to represent a concept, e.g. the concept of an action or an object. For example the term “chair” is used for objects that may have very different properties and appearances but share the functionality that a human can sit on them [Murphy, 2002]. In computer vision, a *visual word* is a representative for a class of similar image elements. Instead of a common functionality (like in the chair example), image elements that are assigned to the same visual words, share a common appearance - or at least some (the measured or described) aspect of their appearance is similar. For practical implementation of this concept, the measured appearance aspect is given by a descriptor as they were introduced in the previous section.

According to natural language, a *visual vocabulary* is a set of visual words. Sometimes, visual vocabularies are also coined codebooks or visual dictionaries. However, the latter term will be used for a different concept in this thesis, introduced in section 6.4.2. More formally, according to [Delhumeau et al., 2013], a visual vocabulary $V = \{w_1, ..., w_k\}$ is a quantization function on a d dimensional input (e.g. image descriptors):

$$q : \mathbb{R}^d \to V \subset \mathbb{R}^d$$
$$x \mapsto q(x) = argmin_{w \in V} ||x - w||^2 \tag{6.2}$$

A well know application are Bag-of-Visual-Words (BoW) approaches for image retrieval. In their "Video Google" paper Sivic and Zisserman [2003] proposed to use a BoW approach in the context of image search. There, the visual elements are keypoint descriptors that are quantized into $\sim 10^4$ visual words using the k-means clustering algorithm. Each resulting cluster is a visual word and is used as a representative for a set of similar keypoint descriptors. More specifically, it is a representative for all descriptors for which there is no other word with a smaller descriptor distance. In [Sivic and Zisserman, 2003], the image is then represented by the frequency histogram of visual words that were assigned to all keypoint descriptors in this image. To compute the similarity of images, they further incorporate term-frequency-inverse-document-frequency (tf-idf) weighting, known from natural language processing and information retrieval [Baeza-Yates and Ribeiro-Neto, 1999].

Since then, BoW approaches were successfully used and extended. For example, Lazebnik et al. [2006] extended the BoW approach with spatial information in the form of a spatial pyramid matching and applied it to scene categorization. In terms of the generation of the visual vocabulary, several improvements over the plain k-means have been proposed. More complex vocabulary based representations like Fisher vectors [Perronnin and Dance, 2007] and vectors of aggregated local descriptors (VLAD) [Jegou et al., 2012] showed improved performance on image retrieval compared to plain BoW. While they replace the frequency histogram of the BoW with a more sophisticated representation based on Gaussian mixture models and residual vectors, they also build upon visual vocabularies.

However, in this thesis, we are less interested in image retrieval, but in properties and recent progress in the computation of visual vocabularies. Nister and Stewenius [2006] generate a vocabulary tree that provides a hierarchical clustering. Words are not only stored in leaves, but also in the inner nodes of the tree. These virtual or generic words have a naturally lower inverse-document-frequency weight since more words are assigned to them. This hierarchy provides some kind of soft-assignment. However, quantization effects do not vanish, they are just shifted a few levels in the hierarchy. Therefore, Perronnin [2008] propose a continuous space soft-assignment. Each feature is assigned to three (approximate) nearest neighbours with a weighting of $e^{-\frac{d^2}{2\sigma^2}}$, based on the descriptor distance d. While this refines the assignment, it also causes higher computational effort. Another approach of codebook refinement was presented by Jegou et al. [2008]. They use Hamming embedding to encode the localization of the descriptor in the Voronoi

cell of the visual word obtained by a coarse quantization based on k-means. They show how this binary signature in combination with a weak geometric consistency check based on scale and angle improves the large scale image search performance. However, this also increases runtime and requires additional memory to additionally store the binary signatures, which is an important drawback in large scale image retrieval.

Mikulik et al. [2013] present another approach on learning visual vocabularies and similarity measures for image retrieval based on bag-of-words. They estimate the probability of observing a visual word w_j in a matching database image when the word w_q is observed in the query image. This is based on the observation that the similarity of SIFT descriptors is not only dependent on their distance but also only on their location in the descriptor space [Lowe, 2004]. This observation is also exploited in [Mikolajczyk and Matas, 2007], where a globally learned Mahalanobis distance showed improved performance for matching SIFT features.

Except for the approach presented by Mikulik et al. [2013], visual vocabularies are an approximation to the direct matching of individual descriptors resulting in some performance loss. In this chapter 6 visual vocabularies are used in a somehow different way: Here we use the representatives of visual elements to *learn* something common for all represented elements.

6.3.2. Evaluating the semantics of superpixel words

The proposed approach to use superpixel vocabularies to predict systematic appearance changes relies on the assumption, that similar superpixel words are associated to parts of the scene that changes similarly. This is the basis for learning a dictionary to translate the words, e.g. from summer to winter. The goal of this section is to investigate the validity of this assumption. The key idea is to evaluate the amount of semantic meaning contained in a superpixel word. In particular, if the words do not provide some semantic meaning, i.e. if there is no relation between the words and semantic concepts like *tree*, *street*, or *house*, we cannot expect the proposed SP-ACP system to learn useful predictions of appearance changes of these concepts.

In the here conducted experiments, the semantic meaning is provided in terms of class names: given an image of a chair, its semantic information is "these pixels of this image show a chair". This is similar to the outcome of semantic segmentations where the task is to assign a class label to each pixel. Superpixel segmentations can be used as a preprocessing step for such segmentations and superpixel vocabularies accordingly as a preprocessing step for assigning such class labels. Therefore superpixel words cannot be expected to directly provide (or correspond to) class labels, but to provide information about possible class labels. The question is, how can the amount of this information be quantified?

In information theory, the *entropy*, or more exactly Shannon entropy, is the expected amount of information contained in a message. For a discrete random variable X with

Fig. 6.7.: Example images from the SUN dataset. This dataset contains images showing a broad variety of indoor and outdoor scenes, ranging from everyday street scenes from different countries to living rooms and the interior of public buildings up to rare scenes like rooms for acoustic measurements (like the bottom right image). In each image, several objects are outlined and annotated with terms like window, sign or chair. The term window is used for windows seen from a street and from the interior of a building or an airplane.

possible values $\{x_1, \ldots, x_n\}$ with probabilities $p(x_i)$, the entropy can be computed as:

$$H(X) = -\sum_{x \in X} p(x) \log_b p(x) \tag{6.3}$$

The basis of the logarithm determines the unit of the resulting entropy. In the following, the basis will be $b = 2$ and the resulting unit bits.

The *conditional* entropy $H(X|Y)$ quantifies the amount of information needed to describe the outcome of a random variable X given that the value of another random variable Y is known. In other words, it describes the remaining randomness of X.

$$H(X|Y) = \sum_{y \in Y} p(y) H(X|Y = y) = \sum_{y \in Y, x \in X} p(y, x) \log_2 \frac{p(y)}{p(y, x)} \tag{6.4}$$

The relation between $H(X)$ and $H(X|Y)$ quantifies what proportion of uncertainty is removed. In this section, I want to use this measure to evaluate the remaining randomness in the class labels (the "random variable" X) when the distribution of the superpixel words (the "random variable" Y) is given. The removed randomness or uncertainty is the answer to the above question, what the superpixel word tells about the class of an image area.

For such an evaluation, we need a set of images with ground truth class labels. The Scene Understanding database (SUN) [Xiao et al., 2010] is a scene categorization and object detection database containing about 130,000 images and more than 4,000 object

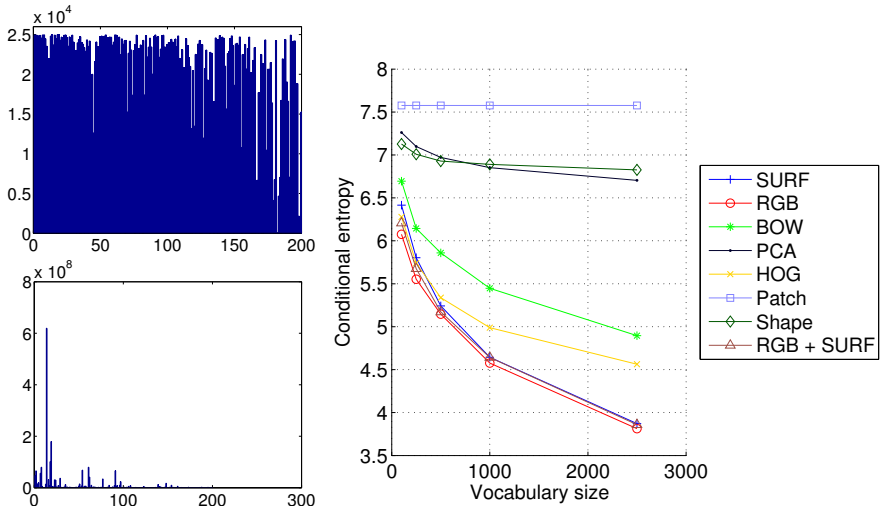


Fig. 6.8.: *(left)* Distribution of the samples of the 200 class labels (top: resampled, bottom: original). *(right)* Results on the SUN dataset. The uncertainty of the class label can be reduced from 7.6 to about 3.8 if the word index is known.

categories. The following experiments use the 16,873 images subset provided for the SUN2012 Object Detection Benchmark. For each image, there is a set of manually annotated image regions. These annotations provide labels for objects like trees, cars, buildings, tables or books, but also other regions like the sky, streets, walls and so on. Example images are shown in Fig. 6.7.

The following experiments use the SUN data with additional tag unification. This reduced the number of labels to about 200 by merging similar labels like "trees" and "tree" or "person" and "child". The labels have very different frequencies, e.g. there are much more pixels labeled as "tree" than there are "speaker" pixels. To remove this prior, the pixels are resampled to obtain not more than 25,000 samples for each class. The distributions over the class labels before and after the resampling can be seen in Fig. 6.8. For some classes, there are less than this number of samples available, resulting in lower bars in the histogram. For an increasing number of samples, the number of classes that do not provide the requested number of samples also grows. Setting a sample threshold also reduces the number of effectively incorporated images to 1,000. The maximum entropy would be achieved by a uniform distribution, classes with less than the maximum number of samples reduce the entropy of the distribution of the class labels. The resulting entropy of the distribution of the 200 class labels shown in the upper histogram in Fig. 6.8 according to equation 6.3 is $H = 7.6$ bits, this would correspond to a uniform distribution of 194.6 classes.

In this evaluation a visual vocabulary is composed by a segmentation procedure (a superpixel algorithm and its parameters, e.g. the number of segments), a descriptor that can be computed for each segment and a clustering scheme to quantize the descriptors according to equation 6.2. The vocabulary is learnt on the BSDS500 training data: each

image is segmented and a descriptor is computed for each segment. The set of descriptors is then quantized. Although more sophisticated clustering schemes are available, k-means clustering (cf. section 2.3.1.2) is used in all experiments. Due to the intended inclusion in the SP-ACP system, a compact segmentation algorithm is selected. In the following experiments, the proposed Preemptive SLIC algorithm with $n = 250$ segments is used.

Fig. 6.8 shows the resulting curves for the descriptors from the previous evaluation in section 6.2.4. Since this task is closer to object detection than the general evaluation of superpixel descriptors, the histogram of oriented gradients (HOG) descriptor, that has been developed for such tasks, is also included. It can be seen, that with increasing numbers of words, the uncertainty drops. The intuitive explanation is that with larger vocabulary size, more words can be used to encode the class labels and individual words can correspond to more specific instances of classes. This comes at the cost of generalisation of the words - in these larger vocabularies, words are more specialized. This will, for example, obstruct the generalization capabilities in the prediction step of the SP-ACP system. Thus, the maximum vocabulary size in the comparison is limited to 2,500. The order of the performance of individual descriptors is different to the order in the above matching-like evaluation. This is not surprising since the two comparisons evaluate the two different applications of the superpixel descriptors: the matching of exactly the same image part and the recognition of image parts that belong to the same class. In particular, the image patch has very weak generalization capabilities while the colour histogram performs better than before. Since the combination of different descriptors in the PCA descriptor includes the shape context, its performance also drops. The combination of the two best descriptors, RGB and SURF (the descriptors were simply normalized to unit length and concatenated), performs comparable to its individual components. The best vocabularies in this evaluation remove about half of the uncertainty. For comparison, the initial entropy of 7.6 bits corresponds to a classification problem with uniform distribution of about 194 classes, the remaining entropy of 3.8 bits corresponds to a classification problem with about 14 classes.

Similar to the above evaluation of superpixel descriptors in section 6.2.4, this comparison is by far not complete. However, it showed that superpixel words carry some semantic information, although they are far away from providing class labels directly. Moreover, we gained some insights on useful descriptors in this context.

Result 6.2 *For the classification-like task, other descriptors than for the matching-like task are preferable. Colour histograms and the keypoint descriptors are good choices. The amount of semantic meaning in superpixels is promising for usage in the proposed appearance prediction system.*

6.4. Algorithmic approach SP-ACP: Learning to predict scene changes

In this section we explore how the changing appearance of a scene across different environmental conditions can be predicted based on the introduced components superpixel

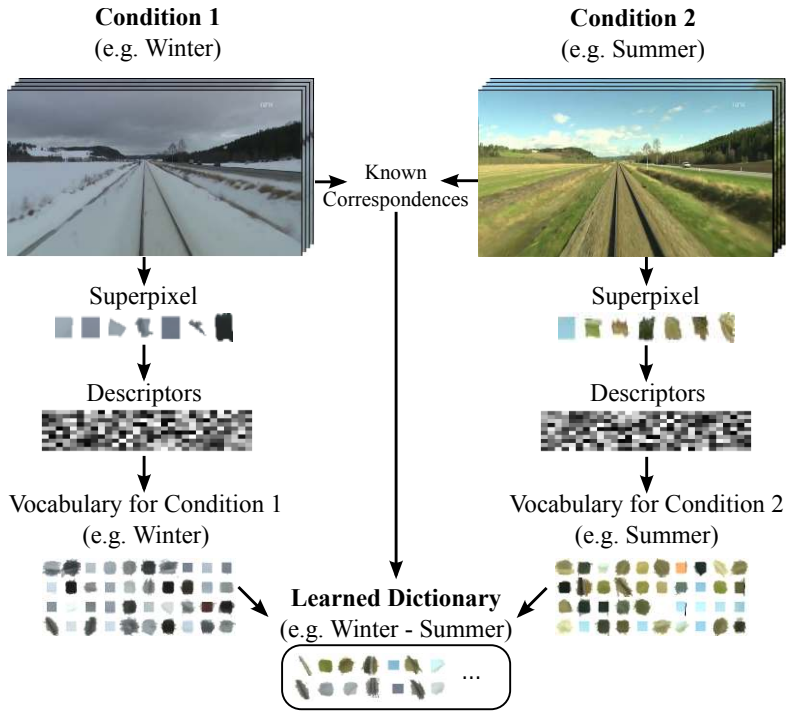


Fig. 6.9.: SP-ACP learning a dictionary between images under different environmental conditions (e.g. winter and summer). The images are first segmented into superpixels and a descriptor is calculated for each superpixel. These descriptors are then clustered to obtain a vocabulary of visual words for each condition. In a final step, a dictionary that translates between both vocabularies is learned. This is based on known pixel-accurate correspondences between the input images.

segmentations, descriptors and vocabularies. Throughout the remainder of this section these changing environmental conditions will be summer and winter. However, the concepts described in the following can of course be applied to other sets of contrasting conditions such as day/night or weather conditions like sunny/rainy etc.

We use superpixels as image parts and cluster them to vocabularies using a descriptor. To predict how the appearance of a scene changes between summer and winter, we first conduct a learning phase on training data (6.4.1) which comprises scenes observed under both summer and winter conditions. In the subsequent prediction phase (6.4.3), the appearance of a new image seen under one of the conditions is predicted as it would be observed under the other viewing condition.

6.4.1. Learning a vocabulary for summer and winter

During the training phase we have to learn a vocabulary for each viewing condition and a dictionary to translate between them. In a scenario with two viewing conditions (e.g. summer and winter), the input to the training are images of the same scenes under both viewing conditions and known associations between pixels corresponding to the

same world point. Obviously the best case would be perfectly aligned pairs of images. This requirement for almost pixel-accurately aligned training images is clearly a major limitation of our current system. The Nordland dataset discussed in Section 5.1.1 fulfils these needs.

Fig. 6.9 illustrates the training phase. Each image is segmented into superpixels and a descriptor for each superpixel is computed. The set of descriptors for each viewing condition is clustered to a vocabulary using hierarchical k-means. Each cluster center becomes a word in this visual vocabulary. The descriptors and the average appearance of each word (the word patch) are stored for later synthesizing of new images.

6.4.2. Learning a dictionary to translate between vocabularies

The learned visual vocabularies for both summer and winter conditions are able to express a typical scene from their respective season. The next step is learning a dictionary that allows translating between both vocabularies. This is illustrated in the lower part of Fig. 6.9. Since the images from the training dataset are aligned, we can determine how single words behave when the environmental conditions change. By overlaying the two aligned images from both summer and winter conditions, every pixel is associated with two words, one from the winter and another from the summer vocabulary. For each combination of words from the summer and winter vocabulary we can then count how often they have been associated to the same pixel coordinates.

This process is repeated for every pair of corresponding images in the training dataset, step-by-step building a distribution over the occurring translations between words from one vocabulary into the other. The final dictionary can be compiled by either storing the full distribution or ignoring it and using a winner-takes-all scheme that stores only the transition that occurs most often. The experimental results of section 6.5 will compare both approaches.

6.4.3. Predicting image appearances across seasons

Fig. 6.10 illustrates how we can use the learned vocabularies and the dictionary to predict the appearance of a query image across different environmental conditions.

The query image is segmented into superpixels and a descriptor for each superpixel is computed. Using this descriptor, a word from the vocabulary corresponding to the current environmental conditions (e.g. winter) is assigned to each superpixel. The learned dictionary between the query conditions and the target conditions (e.g. winter-summer) is used to translate these words into words of the target vocabulary.

Since the vocabularies also contain *word patches*, i.e. an expected appearance of each word, we can synthesize the predicted image based on the word associations from the dictionary and the spatial support given by the superpixel segmentation. Notice that when the dictionary provides the full distribution over possible translations for a word (as opposed to the winner-takes-all scheme), the resulting synthesized image patches are built by the weighted mean over all patches from the target words in the distribution. No further processing (e.g. as proposed in [Efros and Freeman, 2001; Hertzmann et al.,

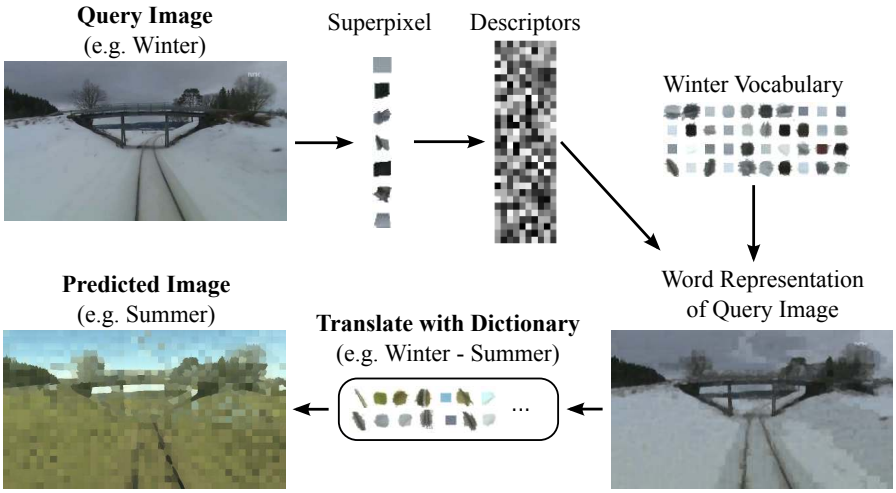


Fig. 6.10.: SP-ACP predicting the appearance of a query image under different environmental conditions: How would the current winter scene appear in summer? The query image is first segmented into superpixels and a descriptor is calculated for each of these segments. With this descriptor each superpixel can be classified as one of the visual words from the vocabulary. This word image representation can then be translated into the vocabulary of the target scene (e.g. summer) through the dictionary learned during the training phase (see Fig. 6.12). The result of the process is a synthesized image that predicts the appearance of the winter query image in summer time.

2001]) is done to improve the appearance or smoothness of the resulting word images. Example word images and predictions are shown in Fig. 6.12.

Contribution 6.3 *The described learning and prediction procedures provide a superpixel-based approach to create predicted images for usage for place recognition in changing environments.*

6.5. Results on the Nordland dataset

After the previous sections explained our proposed appearance change prediction system we can now describe the conducted experiments and their results. For the here presented evaluation, we use the Preemptive SLIC algorithm as described in section 4.2 to segment the image into 1,000 superpixels. According to the previous insights on superpixel descriptors for superpixel vocabularies, we use a combination of colour and SURF descriptor. With respect to the characteristics of the image data that typically occurs in place recognition tasks, we use an upright SURF descriptor (128 Byte) and also include the y-coordinate of the superpixel centre. The influence of this additional information is evaluated in Fig. 6.13. For colour, we use a 30 bin colour histogram in Lab colour space.

We evaluate the proposed SP-ACP prediction system by using it as a preprocessing step to the existing place recognition algorithms FAB-MAP [Cummins and Newman,

Place Recognition Algorithm	Validation Dataset	Test Dataset
FAB-MAP	Section 6.5.2	-
BRIEF-Gist	Section 6.5.3	-
SeqSLAM	Sections 6.5.4 and 6.5.4	Section 6.5.4.1

Table 6.1.: Overview of conducted experiments on the Nordland dataset

Parameter	Value
Number of Superpixels	1,000
Number of Words	10,000
Incorp. translations	all
Superpixel Descriptor	normalized Lab histogram, 10 bins per channel 128 Byte upright SURF descriptor superpixel center y coordinate

Table 6.2.: Overview of the SP-ACP default parameters.

2008], BRIEF-GIST [Sünderhauf and Protzel, 2011] and SeqSLAM [Milford and Wyeth, 2012]. For these established approaches, we will compare the respective performance of

1. directly matching between images of different seasons, e.g. winter vs. spring
2. using the proposed SP-ACP to predict the changed appearance of one of the seasons and e.g. match a predicted winter image against the real winter images

We will calculate precision and recall curves according to section 5.3 and apply the resulting F-score as the primary performance measure. As a reminder: Given the similarities between all possible image matchings, they are divided into matchings and non-matchings based on a threshold t. All image matchings that correspond to a ground truth place correspondence are counted as true positives, all matchings that do not correspond to a single place according to the ground truth are considered false positives, false negatives are all image pairings of the ground truth that are not in the set of matchings. From these three values, a point on the precision-recall curve is computed, the threshold t is varied to obtain a curve.

Since modern SLAM systems do not have to rely on their place recognition front ends to operate at 100% precision anymore [Sünderhauf and Protzel, 2013], the recall at 100% precision is used as a secondary performance indicator only. Experiments with all three mentioned algorithms will first be conducted on the *validation* dataset. We will use SeqSLAM to perform an analysis of the parameters of SP-ACP and determine the optimal values for these parameters. Finally, SP-ACP using these optimal settings is applied to the *test* dataset that covers the complete Nordland journey to demonstrate how place recognition in changing environments can benefit from our proposed appearance change prediction. Table 6.1 gives an overview of the experiments and table 6.2 shows the default parameters. Deviations from these setting are indicated for each experiment.

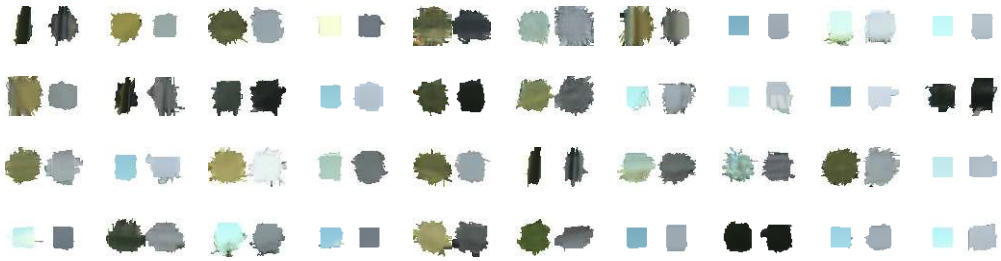


Fig. 6.11.: Example words and their translations. Each tiny image pair shows a word from the spring vocabulary and the word from the winter vocabulary with the highest impact on the translation.

6.5.1. Applying SP-ACP: Predicting images of the Nordland dataset

Using the Nordland training dataset (section 5.1.1) and the *training* procedure introduced in section 6.4.1, we can learn vocabularies for spring and winter conditions and a translation (dictionary) between them. Fig. 6.11 shows example pairs of words from both vocabularies. For each spring word, we show the winter word with the highest impact on the translation (which is the one that would be applied in a WTA setup). Following the proposed *prediction* procedure of section 6.4.3, we can use the learned vocabularies and dictionary to create a predicted winter image for a given spring image or vice versa. Example predictions are visualized in Fig. 6.12 and Fig. 6.17. To evaluate the benefit of the proposed prediction step for place recognition, we can now use such predicted images as input for existing place recognition algorithms.

6.5.2. Experiments with FAB-MAP

In a first experiment we evaluated the performance of FAB-MAP [Cummins and Newman, 2008] (using the openFAB-MAP implementation) on the Nordland dataset. We let FAB-MAP learn its visual vocabulary on either the spring training dataset, the winter training dataset or a combination of both. As expected, directly matching winter against spring images was not successful with each of this setups: The maximum measured recall was 0.025 at 0.08 precision. This is presumably because FAB-MAP fails to match common features in the images from both seasons. The images produced by our proposed scene change prediction approach are not suitable for FAB-MAP since the patch structure of the synthesized images interferes with the necessary keypoint detection. In the following, we therefore examine two holistic approaches.

6.5.3. Extending and improving BRIEF-Gist

BRIEF-Gist [Sünderhauf and Protzel, 2011] is a so called holistic descriptor, i.e. a descriptor that describes the appearance of the complete image and not of single regions in it. The idea of a holistic scene descriptor was e.g. examined by Torralba et al. [2003] with the introduction of the Gist descriptor. We chose the faster and more simple BRIEF-Gist descriptor on the opponency colour space, using 32 bytes per channel.

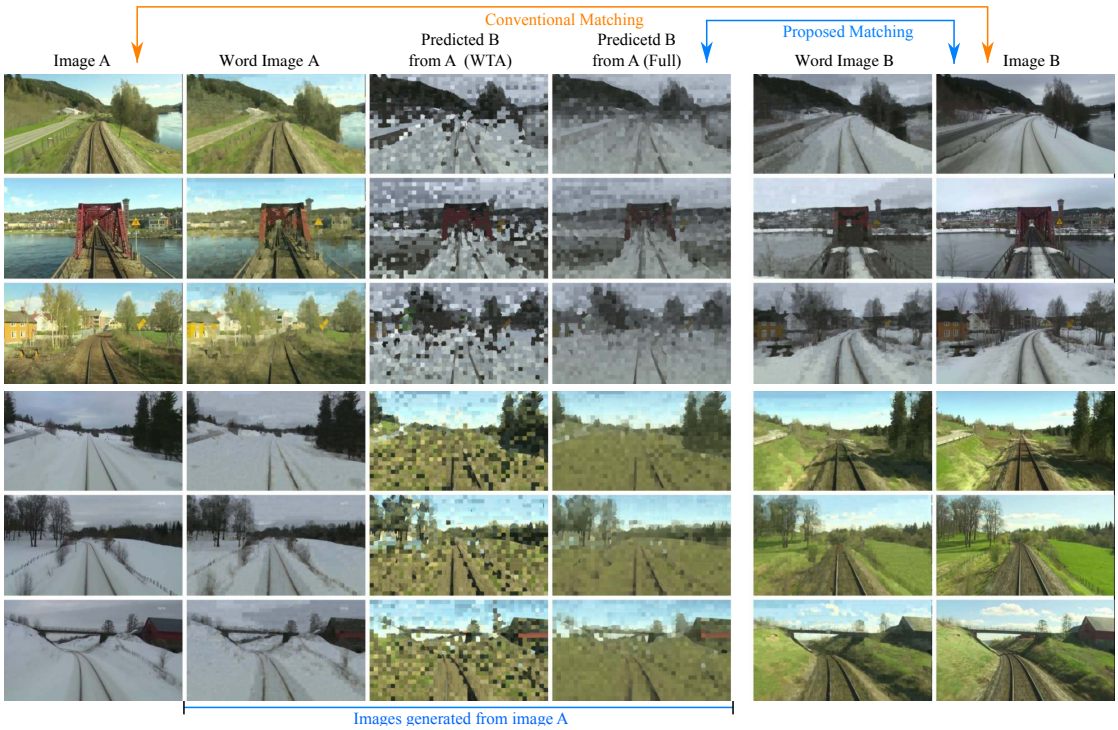


Fig. 6.12.: Example images of the Nordland dataset, their word representations and predictions. The first column shows input query images A given to the prediction framework. The second column is a representation of the query image with words of the first vocabulary. All superpixel segments are replaced by word patches (word image). Applying a winner-takes-all dictionary (WTA) or a dictionary that uses the full distribution translates the words to the second vocabulary. Column three and four show the resulting predicted images B. For comparison column six shows the corresponding real image B and column five its word image representation. We propose not to match the visually very different images A and B directly, rather we propose to use a predicted image B for matching.

Experimental setup

In the following, the performance of BRIEF-Gist to recognize places of the Nordland dataset between spring and winter images is evaluated. We contrast the performance with and without the proposed prediction step and compare different setups of the prediction framework using the validation dataset. For each setup we compute a similarity matrix by comparing each combination of a spring and (potentially predicted) winter image. Since we know that spring and winter image sequences are synchronized, the ground truth similarity matrix is a diagonal matrix. For a quantitative evaluation we apply thresholds and compute precision-recall curves (cf. section 5.3). Due to inaccuracies during synchronization and local self similarity we allow matchings of images with up to five frames distance in the sequence. To evaluate a setup of the prediction framework, we predict a winter image for each spring image based on the learned superpixel

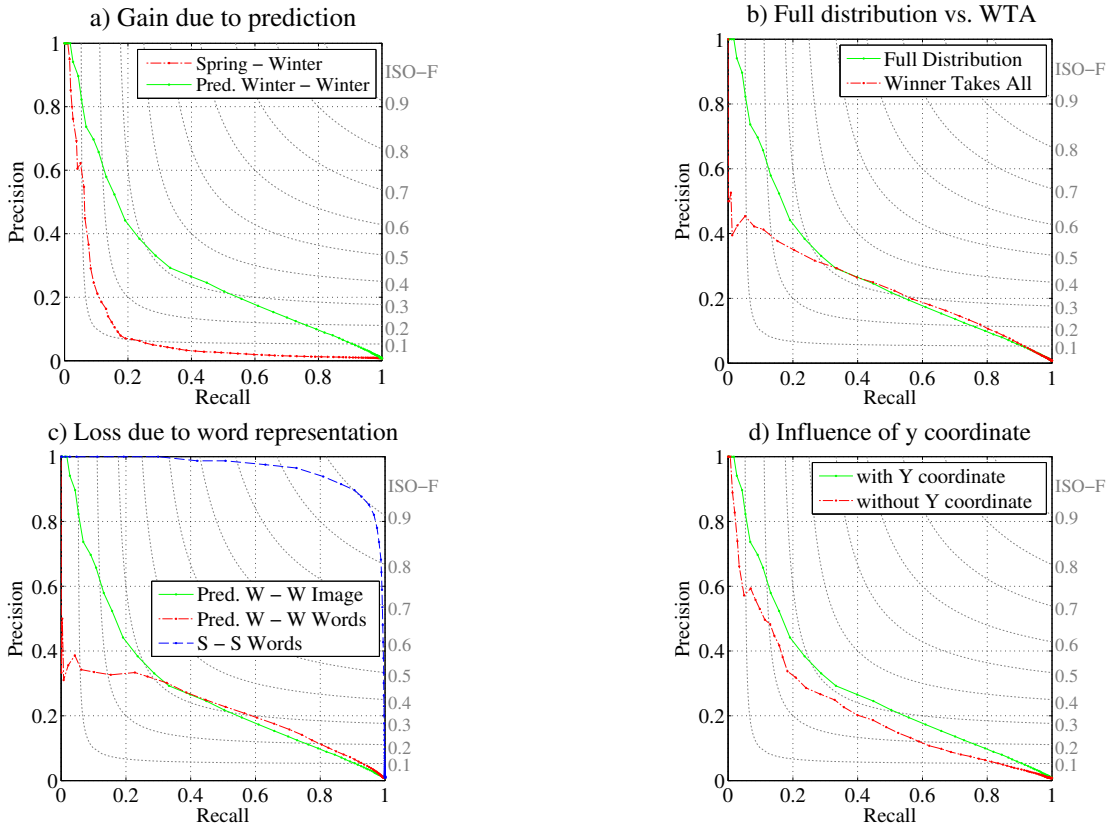


Fig. 6.13.: Evaluation of the SP-ACP framework with BRIEF-Gist. a) Matching predicted winter images to winter images performs better than matching spring to winter images directly. b)-d) Comparison of several setups of the prediction framework. See text for details. The green curve represents the same setup in all plots.

vocabularies and dictionary and use this for matching against the real winter images.

Results

The results of the evaluation with BRIEF-Gist are illustrated in Fig. 6.13. The red curve in Fig. 6.13 a) shows that due to the extreme appearance variations, direct matching of spring to winter images fails. However, the green curve shows the performance improvement when the proposed additional SP-ACP step is applied and matching is done between the winter and a *predicted* winter image. Although the recall at 100% precision does not benefit from the prediction, the maximum F-score improves from 0.14 to 0.31.

Fig. 6.13 b) compares the two proposed methods to build the dictionary, namely winner-takes-all (WTA) and storing the full probability distribution. The green curve in b) is the same as in a). From the red curve we can conclude that the WTA scheme has disadvantages in the important high precision area and storing the full distribution is beneficial.

Matching the predicted winter images against the *word* representation of the original

winter images leads to a very similar loss of performance as can be seen in Fig. 6.13 c). To illustrate what is lost due to the transition from real images to word images, the blue curve in c) represents the performance of BRIEF-Gist when matching the spring images to their own (spring) word representation. In a final experiment shown in d), we removed the y-coordinate from the superpixel descriptor. The red curve illustrates the slight performance drop if this additional knowledge is omitted.

We can conclude that predicting the changed appearance of a scene improves the place recognition performance of BRIEF-Gist. This was clearly illustrated by Fig. 6.13 a). The best results were obtained when exploiting the full distribution over possible translations in the dictionary, matching predicted images against original images, and including the y-coordinate into the word descriptor.

Result 6.3 *The overall performance of place recognition on the challenging Nordland dataset using BRIEF-Gist is poor. There is clear benefit from the inclusion of the proposed prediction step using the novel SP-ACP approach. Including the y-coordinate in the descriptor and using the full distribution in the dictionary is preferable.*

6.5.4. Extending and improving SeqSLAM

Published by Milford and Wyeth [2012], SeqSLAM performs place recognition by matching whole *sequences* of images. This is in contrast to previous approaches like FAB-MAP or BRIEF-Gist that search for a *single* globally best match. Milford and Wyeth [2012] reported impressive recognition results on the Alderley dataset that contains footage recorded from a moving car during bright daylight and a rainy night. However, the matching performance comes at a price: SeqSLAM relies on relatively long sequences to be matched in order to reject false positive candidates. If loop closures in the trajectory form many but short overlapping sequences that are shorter than the required minimum length, SeqSLAM would fail. In order to be applicable in more general settings for long term navigation, this minimum sequence length has to be kept as short as possible.

Our goal is therefore to show that SeqSLAM's performance on short sequence lengths can be improved by combining it with our proposed scene change prediction.

Experimental setup

SeqSLAM preprocesses the camera images by first downsampling them to e.g. 64×32 pixels before performing patch normalization. A simple sum of absolute differences measure determines the similarity between two images. Combining SeqSLAM with scene change prediction is particularly easy, SP-ACP can be executed as a preprocessing step before SeqSLAM starts with its own processing. Since in the experiments we attempted to match spring against winter images, we predicted the visual appearance of each spring scene in winter and fed the predicted winter images together with the original real winter images into SeqSLAM. We use the open source implementation OpenSeqSLAM [Sünderhauf et al., 2013] available online[4]. The processing chain for computing place

[4] `https://openslam.org/openseqslam.html` [Last accessed: 2015-08-14]

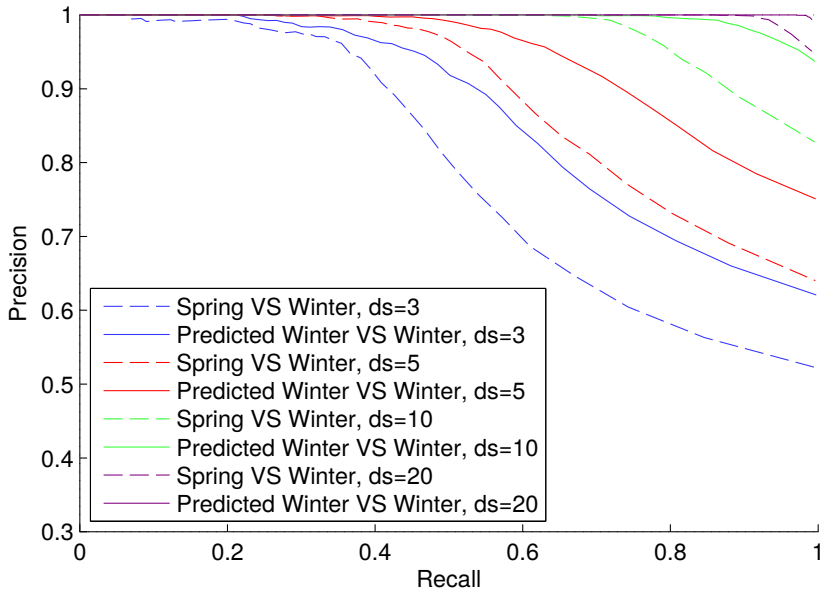


Fig. 6.14.: Evaluation on the validation dataset. Precision recall plots obtained by combining SeqSLAM [Milford and Wyeth, 2012] with the proposed SP-ACP approach (solid lines) compared with SeqSLAM alone (dashed lines). Color indicates different trajectory lengths (d_s) used by SeqSLAM during the sequence matching. It is apparent that our proposed approach can significantly improve SeqSLAM's performance for all values of d_s. (For this experiment: #superpixels=2,500)

recognition using SeqSLAM incorporates a selection of the *single* best matching for each query image. This influences the resulting precision-recall curves as was described in section 5.3. However, this is not a problem since we compare SeqSLAM with and without the additional prediction steps and both cases are affected identically.

Results on the validation dataset

Fig. 6.14 compares the achieved results on the validation dataset similar to the experiments using BRIEF-Gist. The precision-recall plot shows the performance of SeqSLAM alone (i.e. *without* scene change prediction) using the dashed lines. The precision-recall curves for the combination of SP-ACP and SeqSLAM are drawn with solid lines. We show the results for different settings of the SeqSLAM's trajectory length parameter d_s, as indicated by the different colors.

The apparent result is that SeqSLAM can immediately benefit from the change prediction. The gain in precision and recall, as well as the increased recall at 100% precision is visible for all trajectory lengths d_s. The F-score increases by almost 0.1 for short and mid sequence lengths and tends towards 1 for the longest length. Notice that $d_s = 20$ corresponds to a trajectory length of 10 seconds, since the validation data was captured with 2 Hz from the original video footage.

We have to remark that the Nordland dataset is perfectly suited for SeqSLAM since the whole dataset consists of one single long sequence and the camera observes the scene from almost exactly the same viewpoint in all four seasons as the train follows its tracks. Even more, the velocity of the train is equal most of the time in all seasons. In robotic applications these conditions would usually not be met and we can expect SeqSLAM in its current form to perform worse in general. Our point however, was to show that SeqSLAM can in any case benefit from a combination with the proposed appearance change prediction.

We can conclude that although SeqSLAM alone reaches good matching results, they can be significantly improved by first predicting the appearance of the query scene under the viewing conditions of the stored database scenes.

In the following, we are going to evaluate various influences on the prediction quality and the resulting place recognition performance using the validation dataset. This is followed by final results on the complete Nordland track using the test dataset.

Parameter evaluation on the validation dataset

Besides the characteristics of the training data, the number of words in the learned vocabularies, the number of superpixel segments per image, and the amount of incorporated transitions in the prediction are important parameters of the proposed SP-ACP system. Starting from a default parameter setting (#words=10,000, #superpixels=1,000, incorporate all translations during prediction) we vary each of these three parameters to evaluate its influence on the prediction and resulting place recognition performance. Fig. 6.15 shows results of the conducted experiments based on the spring and winter training and validation datasets.

Number of visual words: To evaluate the influence of the number of words in the visual vocabularies, we varied the branching factor of the hierarchical k-means we used to cluster the superpixel descriptors. Since the depth of the tree was held constant (depth 4), this resulted in 81 to 10,000 words. Fig. 6.15 *a)* shows an obvious trend that more words perform better. However, there are some peculiarities: e.g. 256 words perform better than 625, and the performance for 4096 and 10,000 words is almost identical. We assume this results from the limited amount of training data in our experiments. Supposedly we cannot expect to learn the true full distribution to translate between two 10,000 word vocabularies from only one million training samples.

Number of superpixels: Fig. 6.15 *b)* shows the influence of the number of superpixel segments per image. We can observe an expected trade-off in the performance: The higher the number of superpixels, the better object boundaries are covered. However, this also means smaller superpixels, that cover less image content and are less meaningful. In our experiments we observed that the optimal number of superpixels is 2500 per image.

Incorporated translations: Fig. 6.15 *c)* evaluates the last of the three parameters: the percentage of words incorporated in the prediction. Remember from section 6.4.3 that in order to predict an image from e.g. summer conditions to winter conditions, we first compute a word representation of the summer image and then synthesize a winter

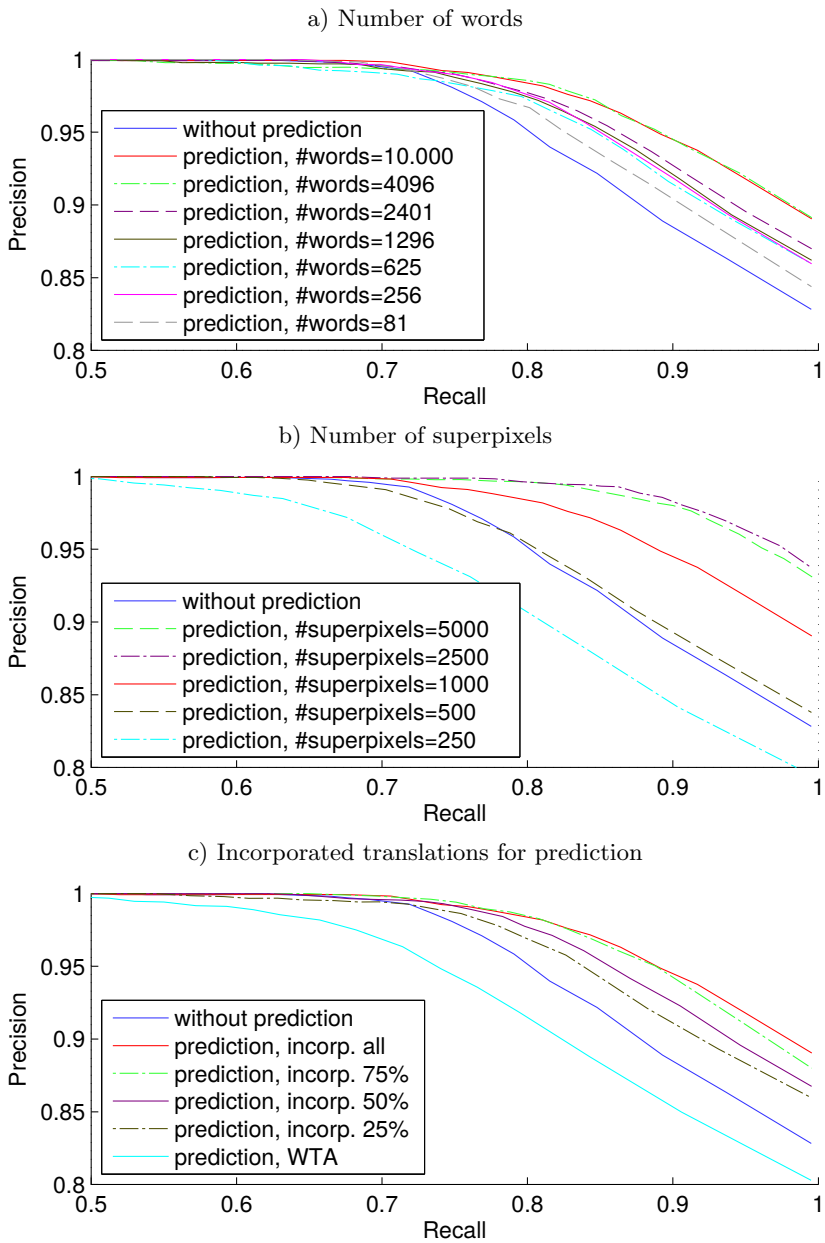


Fig. 6.15.: Parameter evaluation of the proposed prediction framework using SeqSLAM on the Nordland validation dataset. Starting from a default set of parameters, we vary a single parameter to investigate its influence on the overall performance (see text for details). In each plot, the blue solid line is the performance without the proposed prediction framework and the solid red line shows the same default parameter setup of the prediction framework in all plots (#words=10,000, #superpixels=1,000, incorporate all translations during prediction).

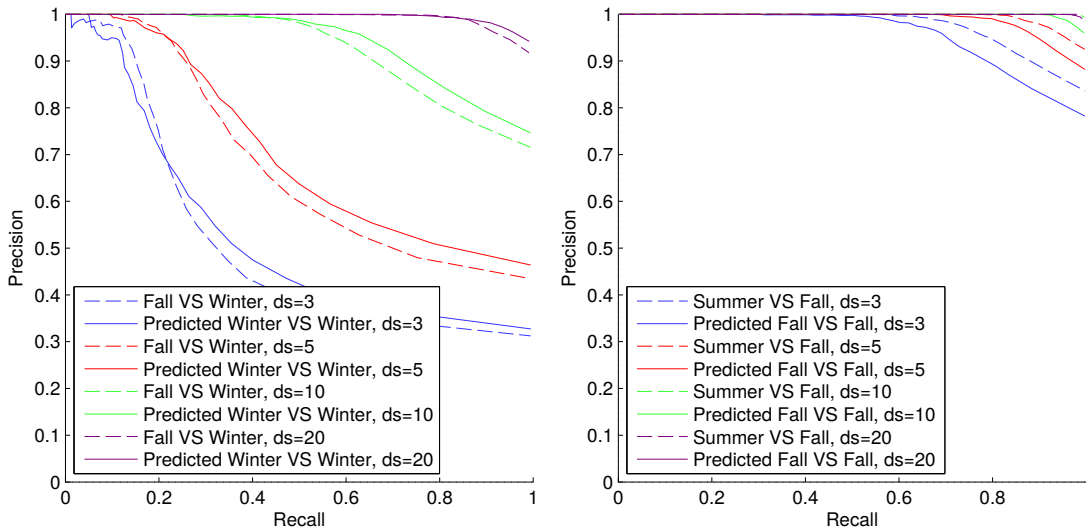


Fig. 6.16.: Predicting between different seasons. Evaluation based on SeqSLAM and the validation dataset. *(left)* Place recognition between fall and winter is a challenging task. Here the prediction framework can help to improve the overall performance. *(right)* In contrast, places in summer and fall (at least for the Norland dataset) are more similar and place recognition is much easier (see Fig. 6.17). In such cases, the prediction result can become worse. However, the overall place recognition performance remains reasonable. (For these experiments: #superpixels=2,500)

representation for each of the summer words using the learned dictionary. During this synthesis, we can either use only the single winter word that translates best according to the training data (winner takes all, WTA); or we can use a weighted combination of the words that explain e.g. 50% of the transitions from the training data or even a weighted combination of all words. The results of this comparison are illustrated in 6.15 *c)*. The obvious conclusion is that incorporating more words yields better results, however, incorporating more than 75% of the probability mass does not yield much improvement.

In this setup, WTA breaks the prediction. The similar experiment using BRIEF-Gist (Fig. 6.13 *b)*) also shows problems in the high precision regime but a clear benefit in the mid- and high-recall regimes. The example prediction results using WTA and the full distribution in Fig. 6.12 show that the WTA predictions have higher contrast between neighbored patches while the predictions from the full distribution are more smoothed. These high local contrasts may have a negative influence in combination with the local patch normalization of SeqSLAM.

Dataset characteristics: Fig. 6.16 shows another important factor for the influence of the proposed prediction step for place recognition: the characteristics of the dataset. These figures show the results for applying SP-ACP and SeqSLAM on other combinations of seasons (i.e. other than spring-winter). In [Sünderhauf et al., 2013] we explored the performance of SeqSLAM (*without* SP-ACP) for the various seasonal combinations

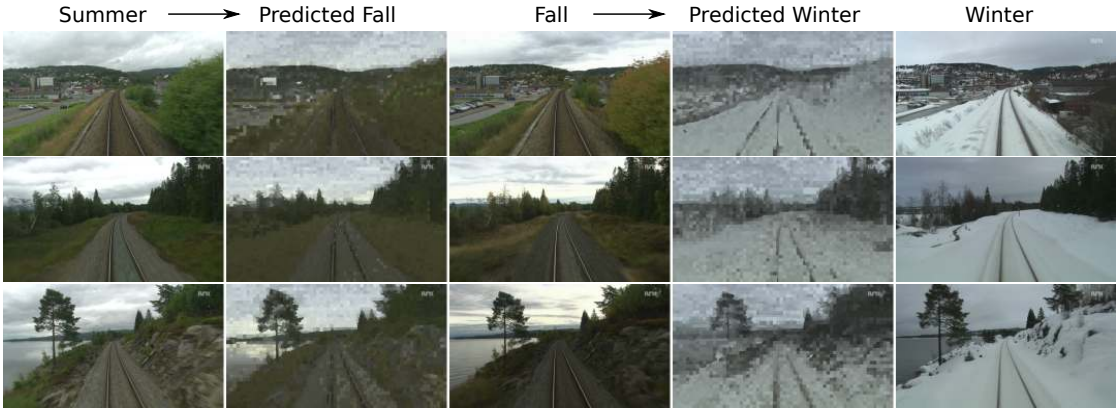


Fig. 6.17.: Example images and predictions for summer, fall and winter. Each row shows the same scene at three different seasons. The images between the original Summer, Fall and Winter images are predictions from summer to fall and from fall to winter.

and found that fall-winter was the most difficult and summer-fall was the easiest combination for place recognition using SeqSLAM without appearance change prediction. We therefore chose these two combinations for comparison.

Both plots in Fig. 6.16 illustrate the results for various sequence lengths with and without the prediction step. We can see that independent of using the prediction or not, finding correct matchings between summer and fall is much easier than between fall and winter. This is an expected result. However, although using the additional prediction step improves performance for the difficult (fall-winter) case, the performance actually decreases for the easy (summer-fall) case. Although the overall performance still remains reasonable, this result needs to be explained: A look at example images in Fig. 6.17 shows that the predicted fall images are visually very similar to the real fall images. However, we suppose that the smoothing and artifacts introduced by the SP-ACP prediction step are a drawback in comparison to the high similarity between original summer and fall images. The same effect is also visible on the comparison of spring images and their word representation using BRIEF-Gist in Fig. 6.13 *c)*. Although these disturbing factors are the same for the comparison between fall and winter images, the prediction still introduces a visually evident benefit since the original imaging conditions are very different.

In the conducted place recognition experiments the prediction step improves the place recognition performance with a prediction of winter from fall but the benefit is not as large as for predicting winter from spring. This is due to the higher diversity in appearance of trees, bushes, meadow etc. in the spring images compared to the fall images. The example images of spring and fall in figures 6.12 and 6.17 give an impression of the richer information in the spring images. This enables the proposed prediction system to better learn the different appearance changes of different image content and thus to produce better predictions from the spring images compared to the predictions from fall images. Incorporating more contextual information (e.g surrounding image

patches or high level knowledge) could help to better exploit the provided appearance diversity. Directions for future work in this direction can be found in section 6.6.

6.5.4.1. Final results on the 728 km test dataset

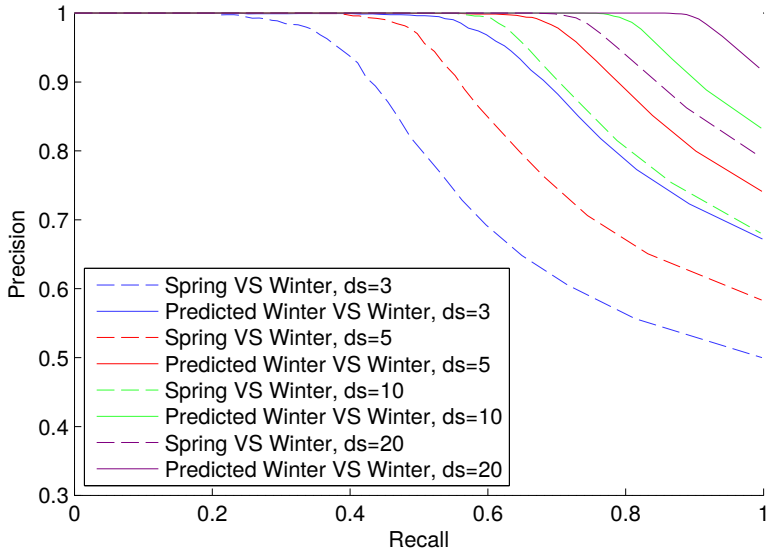


Fig. 6.18.: Final result on the complete Nordland track (*test dataset*). Again, we compare the performance of the combination of SeqSLAM [Milford and Wyeth, 2012] with the proposed scene change prediction approach (solid lines) and SeqSLAM alone (dashed lines). Color indicates different trajectory lengths (d_s) used by SeqSLAM during the sequence matching. The proposed prediction framework, trained on an 8 minute subset, can significantly improve the place recognition performance on the complete track. (For this experiment: #superpixels=2,500)

The final result of this work is a place recognition experiment using SeqSLAM on the complete 728 km Nordland track between spring and winter using the *test* dataset of section 5.1.1. The training dataset remains the same as for the previous experiments. The ratio of training and test dataset has been illustrated in Fig. 5.4, where the red part indicated the training part and the test dataset was shown in blue. Fig. 6.18 shows the result. Again we compare the place recognition performance with (solid) and without (dashed) the prediction for different values for the SeqSLAM sequence length *ds*. Since the test dataset is recorded with 0.1 frames per second, a sequence length of 3 corresponds to a 30 seconds trajectory in the original video.

It is apparent how SeqSLAM benefits from the proposed prediction step for all sequence lengths on the test dataset. Both the maximum F-score and the recall at 100% precision improve significantly. This is a remarkable result, since it shows that our proposed SP-ACP system is able to extract enough knowledge from the training dataset of only 8 minutes to significantly improve the place recognition on the complete journey of

almost 10 hours. Considering the very different environmental conditions met along the journey (e.g. lowlands, highlands, mountains) and the rather homogeneous appearance (lowlands) in the training dataset makes this result even more remarkable. However, we have to clearly point out that this generalization capability is of course limited to environmental conditions with similar systematic change. The proposed system can not be learned on the Nordland dataset and applied to images of e.g. Manhattan.

Result 6.4 *The performance of SeqSLAM on place recognition on the Nordland dataset using reasonable sequence lengths is encouraging. Again, there is a clear benefit from the combination with the proposed SP-ACP approach for challenging combinations of seasons. The approach showed to be robust for a wide range of parameter configurations. However, for less severe appearance changes (i.e. summer-fall) the artefacts introduced by the prediction decrease the place recognition performance.*

6.6. Current limitations of the approach and future work

The proposed SP-ACP system is a rather straightforward implementation of the idea of incorporating an additional prediction step for place recognition in systematically changing environments. However, there is plenty of space for improvements.

Obviously the prediction step incorporates smoothing and artefacts in the predicted images. This can cause a decrease in place recognition performance if the compared original sequences are very similar (e.g. summer and fall). However, the predicted images are visually appealing and the place recognition performance remains reasonable. This can be interpreted as a kind of "upper bound" for the recognition performance which is introduced by the smoothing and the artifacts of the prediction.

In its current form, our algorithm requires perfectly (near pixel-accurate) aligned images in the training phase. This requirement is clearly a key limitation of the proposed approach, since it is hard to fulfil and limits the available training datasets. An interesting research direction would be to investigate ways to ease or overcome this requirement, e.g. by anchoring the training images on stable features. This would increase the availability of potential training datasets collected by robots or vehicles in realistic scenarios that are close to real-world applications.

Currently, we synthesize an actual image during the prediction. This simplifies the qualitative evaluation by visually comparing the predicted with the real images and further allows for using existing place recognition algorithms for quantitative evaluation. However, the proposed idea of scene change prediction can in general be performed on different levels of abstraction: It could also be applied directly onto holistic descriptors like BRIEF-Gist, onto visual words like the ones used by FAB-MAP or onto the downsampled and patch-normalized thumbnail images used by SeqSLAM.

Furthermore, the learned dictionary can be as simple as a one-to-one association (like the mentioned winner-takes-all scheme) or capture a full distribution over possible translations for a specific word. In future work this distribution could also be conditioned on the state of neighbouring segments, and other local and global image features and

thereby incorporate mutual influences and semantic knowledge. This could be interpreted as introducing a *grammar* in addition to the vocabularies and dictionaries. How such extended statistics can be learned from training data efficiently is an interesting direction for future work.

If the dictionary does not exploit such higher level knowledge (as in the superpixel implementation introduced here) the quality of the prediction is limited. In particular, when solely relying on local appearance of image segments for prediction, the choice of the training data is crucial. It is especially important that the training set is from the same domain as the desired application, since image modalities that were not well-covered by the training data can not be correctly modeled and predicted.

Exploring the requirements for the training dataset and how the learned vocabularies and dictionary can best generalize between different environments will be an important direction of future research. However, in its current form the approach is not expected to generalize well to new environments. As already mentioned, the Nordland dataset provides somewhat optimal conditions (apart from the season-induced appearance changes) for place recognitions, since the camera observes the scene from almost exactly the same viewpoint in all four seasons and the variability of the scenes in terms of semantic categories is rather low. These conditions would usually not be met in a typical robotic application.

In conclusion, the idea of learning systematic image changes is compelling. However, while this presented implementation showed that there are cases where existing place recognition approaches can benefit from the created predictions, the strong requirements on the training data and unclear generalization capabilities limit the prospect of real practical applications.

One new feature or fresh take
can change everything.
Neil Young

7 Landmark-based Place Recognition in Changing Environments

The previous chapter 6 used superpixels in the context of *holistic* descriptors for place recognition - the evaluation showed, that for the Nordland dataset BRIEF-GIST and SeqSLAM (that builds upon a lightweight holistic visual front-end) can both benefit from a prediction based on superpixel vocabularies. According to the taxonomy of localization approaches shown in Fig. 5.1, besides *holistic* image descriptors, there are also *local* feature based approaches to visual place recognition. Holistic approaches are known to be sensitive to viewpoint changes [Pepperell et al., 2014]. Moreover, the proposed Appearance Change Prediction approach poses strong requirements on training data, that are supposed to be very hard to fulfil for practical applications. Therefore, in this chapter, I want to investigate whether local feature based approaches can also benefit from superpixel segmentations.

7.1. The components of place recognition based on local image features

Fig. 7.1 illustrates the components of a typical, simple pipeline for place recognition based on local regions. In fact, these are only the steps of place recognition up to the computation of the similarities of two images - for practical place recognition these image similarities are just the beginning. This is further discussed in the conclusions chapter 8. However, the image similarities are an essential intermediate step and in this pipeline they are obtained in three steps: Given a query image, the first step is to compute a set of local regions. Typical choices for local regions are areas around interest points like scale space extrema or corner points. In a second step, a numerical descriptor is computed for

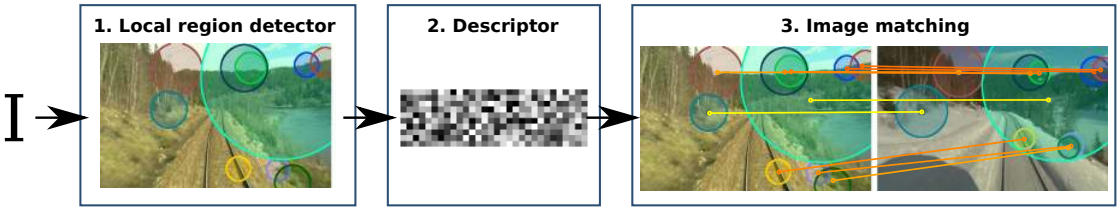


Fig. 7.1.: Components of a typical, simple pipeline for place recognition based on local regions. The first step is the extraction of local regions from the input image, followed by the computation of a descriptor for each region (illustrated in form of a numerical matrix). In a third step, these descriptors are evaluated to compute similarities between images.

each region, e.g. by computing a SIFT descriptor for the image patch associated to the local region. Finally, these descriptors are used to find one-to-one matchings between local regions from different images and to combine these matchings to a resulting image similarity that can be used for place recognition.

For place recognition in *changing environments*, particular care has to be taken on each of these steps. The local region detector is required to repeatedly extract corresponding regions from images subject to severe appearance changes. Moreover, even if the detected regions show the same parts of the world, their changed appearance may pose a major challenge to the descriptor. For the final image matching step, the extracted image regions are supposed to be less precisely detected in the images than in non changing environments. Moreover, a small number of correct region matchings and a large number of outlier region matchings can be expected. This poses additional challenges for established outlier detector schemes like RANSAC or LMSE [Lowe, 2004].

In this chapter, I will lay hands on each of the steps shown in Fig. 7.1. The main question, in the overall context of this thesis, will be if and how superpixel segmentations can contribute to local region detection for place recognition in changing environments. I will propose and evaluate four approaches: a segment soup, a multi scale superpixel grid, a descriptor weighting based on superpixels and the usage of object proposal algorithms that are related to superpixel segmentations.

To evaluate these methods in place recognition experiments and to compare them with existing methods, I will further propose a novel, complete pipeline comprising the three steps in Fig. 7.1. This pipeline can be used with a broad range of existing and novel local region detectors. The key ingredients are a powerful descriptor based on the output of intermediate layers of a Convolutional Neural Network (CNN) as well as a novel image matching method based on a star graph model and Hough voting. This image matching method is particularly suited for weakly localized image features and high numbers of outliers. The resulting place recognition system is training free and single image based, although the resulting image similarities can be combined with sequence matching methods seamlessly. The proposed approach can handle changing environments and viewpoint changes. It is intended to be the basis for a solution for real world application in changing environments.

The following sections 7.2 and 7.3 deal with the second and third steps of Fig. 7.1 and

will introduce the CNN-based descriptor and the star graph model and Hough voting scheme used for computation of image similarities based on the local regions. Section 7.4 introduces the novel approaches to local region detection involving superpixel segmentations. Section 7.5 presents metrics that will subsequently be used in section 7.6 for evaluation of the repeatability of the local region detectors, independently of a descriptor or particular place recognition experiments. Finally, extensive place recognition experiments on the datasets presented in section 5.1 are the topic of section 7.7

7.2. Local region descriptors based on Convolutional Neural Networks

The related work presented in section 5.2 revealed problems of the established combinations of local image feature detectors and descriptors (e.g. SIFT) in changing environments. The results presented later in section 7.7 will in particular show, that the processing pipeline illustrated in Fig 7.1 using SIFT keypoints fails at the *description* step for the Nordland Spring-Winter dataset. Therefore, alternative descriptors are required.

Recently, deep learning based Convolutional Neural Networks (CNN) showed impressive performance on creating descriptive representations for several computer vision tasks, most notably image classification [Krizhevsky et al., 2012], object detection [Girshick et al., 2014; Chatfield et al., 2014] and other recognition tasks [Razavian et al., 2014]. In this thesis, readily trained and publicly available CNNs are used. Therefore, I will reduce their presentation to an overview of their properties, their relation to place recognition and remarks on the resulting computational efforts.

7.2.1. What is a Convolutional Neural Network?

A recent overview on deep learning and CNNs can, for example, be found in [LeCun et al., 2015]. From the user's perspective, the convolutional neural network computes a feature vector (more specific a feature tensor at each layer) describing the input image. Although it is higher dimensional, the feature vector at each layer can be used in a similar way to other feature vectors like SIFT or HOG.

Convolutional Neural Networks (CNNs or ConvNets) are a class of feed-forward artificial neural networks with a specific structure and arrangement of their layers. The lower levels are composed of convolutional and pooling layers. A convolutional layer typically contains several feature maps. Each unit of a feature map has weighted connections to local areas of feature maps in the previous layer. This set of weights is called a filter bank. An important property is that all units of a feature map share the same filter bank. Thus, the evaluation of a feature map can be implemented as convolution of the corresponding areas of the previous layer with this filter bank - hence the name of these networks. The results are then passed through a non-linearity, e.g. a rectified linear unit (ReLU). The convolutional layers are intended to learn and detect distinct local motifs of increasing complexity from layer to layer. A subsequent pooling layer collects the

response of local regions of a feature map, e.g. by taking the maximum response of a neighbourhood. This creates an invariance to small shifts and distortions. Several stages of this combination, convolution, non-linearity and pooling, are stacked and finally create the input for fully connected layers that combine information over the whole image area. Typically, all the weights in the filter banks are learned using backpropagation [Werbos, 1974] on large amounts of data.

The architecture of these networks has been inspired by insight on the early human visual system. The local areas incorporated in the filter banks correspond to receptive fields of cells in our early visual system. Common to both is also the increasing complexity of visual structures that are regarded at later processing stages. While the lowest layers respond to gradients similar to the hand engineered SIFT and HOG descriptors, higher layers learn increasingly complex structures. However, the implementation and applied learning mechanisms in CNNs are significantly different from the human visual system and are adapted for execution on typical computational hardware like CPUs and in particular GPUs. Moreover, important simplifications like the pure feed-forward structure and weight sharing across feature maps are used to keep learning feasible.

7.2.2. CNN for place recognition

Although their structure, details and in particular training is quite complex, a readily trained CNN can be used directly to create image descriptors. The image has to be rescaled to the size of the input layer and properly normalized. The result of the run through the feed-forward net is a set of tensors - one for each layer of the network. The tensors are structured according to the structure of the corresponding network layer. For example, a convolutional layer of size 13 by 13 comprising 512 filter banks creates a $13 \times 13 \times 512$ tensor, or in other words 512 feature maps of size (13×13). This inherent spatial structure can be exploited - or the tensor can be simply reshaped to a feature vector similar to a SIFT descriptor.

Sünderhauf et al. [2015a] investigated the application of CNN-based holistic image descriptors for place recognition in changing environments. They compute a single CNN descriptor from a particular network layer for the whole image and compare images based on the cosine distance of these vectors. They found the lower convolutional layers to be robust against seasonal changes but sensitive to shifts, e.g. induced by viewpoint changes or rotations.

The sensitivity to viewpoint changes is a drawback of *holistic* image descriptors. Therefore, I want to use these powerful CNN-based descriptors in combination with *local* region detectors to create a system that is robust to both, changing environments and viewpoint changes. To create the CNN-based descriptor for a local region, we feed the image patch provided by the bounding box around the local regions into the convolutional neural network. We use the publicly available VGG-M network[1]. The CNN descriptor is the vectorized output of the third convolutional (*conv3*) layer. To compare two CNN descriptors, we compute their cosine distance.

[1] `http://www.vlfeat.org/` [Last accessed: 2015-08-14]

While the idea of computing CNN features for regions is not new (e.g. Girshick et al. [2014] presented R-CNN: Regions with CNN features), the usage as landmarks for place recognition is novel. We presented this idea in [Neubert and Protzel, 2015b]. In concurrent work and independently from our approach, Sünderhauf et al. [2015b] also proposed the combination of CNN-descriptors with local image features for place recognition. They show impressive place recognition results based on a single type of local regions (EdgeBoxes) and a simple image matching scheme. Although an implementation of their complete processing chain is not available, the results section 7.7 will evaluate the region detector they used and their matching scheme and show that both are outperformed by our proposed approaches. There, the proposed approach will also be compared to the holistic CNN.

Contribution 7.1 *The output of the lower layers of convolutional neural networks are promising descriptors for landmarks in changing environments. There are publicly available, ready to use networks. We propose to use these descriptors in combination with local region detectors for place recognition in changing environments.*

7.2.3. Remarks on the computational efforts for CNN-based descriptors

In general, the training of a CNN is very time consuming. In contrast, the computation of the network response for a particular input patch is a single run through the network. On a GPU this can be done in few milliseconds. On a CPU the computation of the above described conv3 descriptor takes about 40 ms including the computations for the Matlab interface (Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz). Currently, each local landmark has to be run through the system independently. However, novel CNN architectures like SPPnet [He et al., 2014] and Fast-R-CNN [Girshick, 2015] aim at fast evaluation of *all* local regions by a *single* run through the network.

The CNN descriptor is supposed to provide more information than a hand engineered SIFT or HOG descriptor. However, at the lower layers, this comes at the cost of very high dimensional feature vectors. For example, the 13×13 tensor of the used networks conv3 layer has in total 86,528 dimensions. Fortunately, we are not interested in the feature vector itself, but in the cosine distance of feature vectors - the distance between the descriptors of landmarks from two different images. The literature provides methods for efficient approximation of the cosine distance of such high dimensional vectors using binary locality-sensitive hashing [Charikar, 2002]: The cosine distance is obtained from the angle between the vectors. Given a random hyperplane in the regarded vector space, the probability that this hyperplane separates the two vectors is directly proportional to the angle between the vectors. [Ravichandran et al., 2005] demonstrate how a large number of randomly selected hyperplanes can be used to closely and efficiently approximate the cosine distance in a high dimensional vector space. Given a set of k random hyperplanes and a set of feature vectors (e.g. CNN descriptors) in a high dimensional space, they compute for each feature vector a binary hash of length k. The i-th bit indicates the sign of the scalar product of the normal of the i-th hyperplane and the feature vector. The cosine distance of two feature vectors can then be approximated by

the Hamming distance of the two binary hash vectors. Sünderhauf et al. [2015a] evaluate this method in the context of holistic CNN descriptors for place recognition and found that an approximation of the CNN vector using a 8,192 bit vector retains approximately 95% of the place recognition performance.

Thus, if efficiency of descriptor comparison plays a role, the 86,528 dimensional CNN descriptor can be reduced to e.g. a 8,192 bit vector while retaining a large fraction of its descriptive power for place recognition. This is eight times the storage required for a 128 byte uint8 SIFT descriptor. The comparison of two vectors is then done by computing the Hamming distance of the bit vectors. The required operations (XOR + bit count) are supported by native processor instructions in modern CPUs (≥SSE 4.2).

While runtime optimization is not in the particular scope of this chapter, the runtime of the conducted experiments has to be feasible. For the later presented experiments, we use the plain CNN descriptor and restrict the number of local landmarks to about 50 per image. Thus, the descriptors for an image are computed in about $50 \cdot 40ms = 2$ seconds using the Matlab interface of VLFeat. For the novel Fast-R-CNN, Girshick [2015] reports a speed-up of factor >200 for the computation of the network output compared to existing networks in the context of object detection. The place recognition based on the full CNN descriptors is feasible for up to a few hundred thousand place comparisons per experiment (e.g. about 3 hours for 100,000 place comparisons including the Matlab interface, 2,500 descriptor comparisons are required for each place comparison when using 50 landmarks). For larger datasets, the binary locality-sensitive hashing proposed by Charikar [2002] can be used for a significant speed-up. Sünderhauf et al. [2015a] report a runtime of 134 ms for 100,000 comparisons of binary hash vectors using an Nvidia Quadro K4000 GPU. However, one has to additionally compute the binary hash for each vector (each bit is basically obtained by computing a scalar product in the high dimensional space).

Result 7.1 *The computational effort for using CNN-based descriptors for local landmarks is feasible. There are promising approaches to further reduce the runtime for descriptor computation and comparison.*

7.3. The novel Star-Hough matching scheme: From local regions to image similarities

To use local regions and their descriptors for place recognition, the information about the similarity of individual landmark comparisons (e.g. the CNN-based descriptor distance) between two images has to be combined to an overall image similarity. This section discusses the influence of changing environments on the existing methods and introduces a novel matching scheme based on star graph models and Hough voting that is suitable for matching images subject to severe appearance changes. In particular, it can cope with small numbers of features including outliers and it is robust to variations in the exact landmark locations.

7.3.1. The basic problem and the challenges of changing environments

An intuitive first step to compute an image similarity is to find for each landmark in the first image the most similar landmark in the second image. The similarity of an image I^1 to another image I^2 can then simply be computed as the average best similarity of a landmark of I^1 to a landmark of I^2. However, this is sensitive to outliers and ignores available information about the spatial relation between landmarks. The literature provides several general approaches and additional tricks, that significantly increase the robustness and matching performance, e.g. left-right checks (also known as reciprocal matchings), evaluation of the ratio of the best to the second best matching and outlier detection based on epipolar geometry, cf. [Hartley and Zisserman, 2003].

More formally, we want to solve the following problem: Given the set of landmarks $F^1 = \{f_1^1, f_2^1, ..., f_n^1\}$ from image I^1 and the set of feature $F^2 = \{f_1^2, f_2^2, ..., f_m^2\}$ from image I^2, we want to find the distance of the two images using these feature sets $dist_{Images}(I^1, I^2) = dist_{Landmarks}(F^1, F^2)$. Each landmark provides a descriptor $d_i^j = descriptor(f_i^j)$ and similarly the position $[x_i^j, y_i^j]$ of each landmark, e.g. the coordinates of the centre of gravity.

For a baseline combination, we can compute the similarity between two images with landmark sets F^1 and F^2 directly from the distance of all left-right matchings and the total number of landmarks:

$$sim(f_i^1, f_j^2) = 1 - dist(f_i^1, f_j^2)$$

$$sim(F^1, F^2) = \frac{1}{\sqrt{|F^1| \cdot |F^2|}} \sum_{\{f_i^1, f_j^2\} \in matches(F^1, F^2)} sim(f_i^1, f_j^2) \tag{7.1}$$

This the sum of the similarities of left-right matches weighted by the number of features in F^1 and F^2. A left-right matching is a matching of two landmarks which best match one another considering both search directions, that is searching the best match for a landmark $f_i^1 \in F^1$ the best match in F^2 and vice versa. The similarity measure in equation 7.1 is, for example, used in [Sünderhauf et al., 2015b] and as a baseline method in [Neubert and Protzel, 2015b].

However, while the number of outliers is reduced by the left-right check, the spatial consistency of the matchings provides additional essential information. A set of landmark matchings of an image pair, for that the motion of the set of landmarks between the images can not be explained by a reasonable camera motion, is suspected. To find the set of correct matchings, i.e. to remove outlier matchings, spatial consistency checks can be used. The spatial consistency can be evaluated in 3d if depth information is available [Moravec, 1980] or in 2d based on epipolar geometry [Hartley and Zisserman, 2003]. While epipolar geometry provides a solid theoretical foundation and showed to be beneficial for outlier rejection, there are two reasons why, for example, a Fundamental matrix estimation is not used in the here presented place recognition approach:

1. Taking advantage of constraints based on multi view or epipolar geometry requires repeated and *precise* localization of features in the image - they have to be at the

more or less *exact* corresponding position. In the presence of severe appearance changes of the environment, we cannot expect, for example, scale space extrema based on differences of Gaussians to be as precisely localized as in non changing environments. For instance, think of the scale space extremum provided by the crown of a tree - in summer and in winter. While both may be detected, their location is supposed to vary significantly. Moreover, the here presented matching scheme is intended for combination with very different types of local region detectors including superpixel segmentations and object proposals - both lack the repeated precise localization of for example corner points.

2. The second issue arises from the small number of features and the high expected rate of outliers due to the changing appearance. For robust estimation of spatial constraints based on epipolar geometry in the presence of outliers, random sample consensus (RANSAC) or least median of squared error (LMSE) methods are often used. However, both "[...] are known to perform poorly when the percent of inliers falls much below 50%" [Lowe, 2004, p. 21].

Fortunately, accumulating information in image space using the Hough transform [Hough, 1962] provides an alternative solution, that showed to perform well in the presence of small inlier rates [Ballard, 1987; Grimson, 1990; Lowe, 2004].

7.3.2. The novel Star-Hough approach

For the application in the context of place recognition, in [Neubert and Protzel, 2015b] we propose to use a simple two dimensional Hough accumulator on the shift of the star graph model induced by the landmarks. The star graph model is similar to the implicit shape model [Leibe et al., 2004]. To represent the mutual relative position of landmarks with only few connections in the resulting dependency graph, all landmarks are anchored at the centre of the whole landmark set. The visualization of the resulting graph has a star-like appearance - hence the name.

Computing the star graph model for a set of landmarks can be implemented as simple transformation into a normalized image coordinate system with origin at the centre of the landmark set. Given a set of landmarks at positions $[x_i, y_i]$, the star graph representation $[x_i^s, y_i^s]$ is computed as follows:

$$x_i^s = \frac{x_i - \frac{1}{2} \cdot (\max_k x_k + \min_k x_k)}{\max_k x_k - \min_k x_k}$$

$$y_i^s = \frac{y_i - \frac{1}{2} \cdot (\max_k y_k + \min_k y_k)}{\max_k y_k - \min_k y_k} \tag{7.2}$$

$[x_i^s, y_i^s]$ is the normalized, relative position of a landmark to the centre of all landmarks. For application to place recognition, we make the assumption, that this relative position of corresponding landmarks to the centre of all landmarks is similar in two images showing the same scene. We cannot expect that this relative position is invariant, in particular since the location of the centre of the observed landmarks can vary significantly, e.g. for different viewpoints.

Algorithm 3: Outline of the algorithmic steps to compute the image similarity using the proposed Star Hough image matching.

Data: Two sets of Landmarks from images I^1, I^2 with descriptors d_i^1, d_j^2 and positions $[x_i^1, y_i^1]$ and $[x_j^2, y_j^2]$
Hough accumulator radius r
Result: Similarity s of the landmark sets

```
// Get left-right matchings, each triple [idx1, idx2, sim]
// is a matching of d^1_idx1, d^2_idx2 with similarity sim
```
1 {idx1, idx2, sim} = getLeftRigthMatchings(d_i^1, d_j^2);

```
// get Star model coordinates
```
2 $xs^1 = (x^1 - \frac{1}{2} \cdot (max(x^1) + min(x^1)))/(max(x^1) - min(x^1))$;
3 $ys^1 = (y^1 - \frac{1}{2} \cdot (max(y^1) + min(y^1)))/(max(y^1) - min(y^1))$;
4 $xs^2 = (x^2 - \frac{1}{2} \cdot (max(x^2) + min(x^2)))/(max(x^2) - min(x^2))$;
5 $ys^2 = (y^2 - \frac{1}{2} \cdot (max(y^2) + min(y^2)))/(max(y^2) - min(y^2))$;

```
// initialize Hough accumulator
```
6 H = zeros($2 \cdot r + 1$, $2 \cdot r + 1$) ;

7 **foreach** *matching [idx1, idx2, sim]* **do**

```
    // Get coordinates of the (x,y) shift in the Hough space
```
8 $xh = r + 1 + (xs^2_{idx2} - xs^1_{idx1}) \cdot r$;
9 $yh = r + 1 + (ys^2_{idx2} - ys^1_{idx1}) \cdot r$;

```
    // update accumulator
```
10 $H(yh, xh) = H(yh, xh) + sim$;

11 **end**

```
// Combine neighboured bins
```
12 H = integrate(H, Gaussian $\sigma = 0.1 \cdot r$);

```
// Image similarity is the maximum value of the accumulator
```
13 s = max(H);

However, similar issues are arise for object detection, where similar Hough voting approaches have been applied successfully. Lowe [2004] uses a Hough accumulator based on the 2d location, scale and orientation of keypoint image features. Each of the image features votes for the location of an object model. As Lowe points out, "the similarity transform implied by these 4 parameters is only an approximation to the full 6 degree-of-freedom pose space for a 3d object and also does not account for any non-rigid deformations" [Lowe, 2004, p. 21]. We incorporate two approaches to deal with this problem:

1. We use a coarse accumulator, e.g. of size (41×41) for images of size (854×480). Thus, each bin includes a wide range of shifts.
2. We use soft binning to distribute votes over several bins. According to Szeliski [2010, p. 223], "[...] softly distributing values to adjacent histogram bins is generally a good idea in any application where histograms are being computed, e.g., for Hough transforms". In our implementation this is done by a simple smoothing of the resulting accumulator image with a Gaussian kernel. For the (41×41) accumulator, we use a Gaussian with $\sigma = 2$.

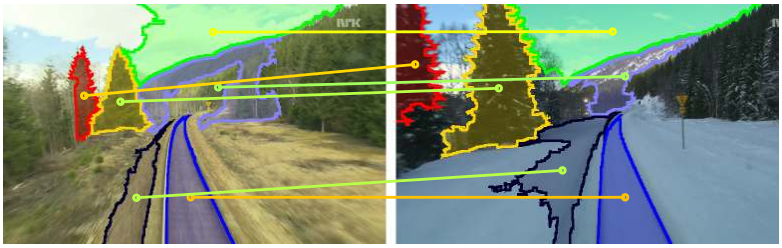


Fig. 7.2.: Isn't it an appealing idea to have segmentation based landmarks? This example image pair from the Nordland Spring-Winter datasets (with horizontal shift and time delay) has been successfully matched based on landmarks from a superpixel segment soup.

The algorithmic steps are listed in algorithm 3. Given a set of landmarks, each with a descriptor and a position in the image, a set of initial matchings using a left-right check is computed in line 1. This results in triples *[idx1, idx2, sim]* indicating that landmark *idx1* from I^1 has been matched to landmark *idx2* from I^2 and their similarity is *sim*.

Lines 2-5 implement the coordinate system transformation to the star graph given by equation 7.2. The two dimensional Hough accumulator is initialized with zero (line 6) and iteratively updated with each landmark matching (lines 7-11). Each shift between matched landmarks votes for a shift of the star model centres. This shift is converted to coordinates of the Hough accumulator (lines 8 and 9) and the value of the according bin is increased by the similarity value of the matching. To implement soft binning, the accumulator is smoothed with a Gaussian in line 12. Since no non-linearities are involved in the iterative update of the accumulator, this smoothing can be done in a single smoothing step for all matchings. The resulting image similarity is the maximum value of the accumulator. For landmark sets of strongly varying size, this value has to be normalized according to the normalization term in equation 7.1.

The benefit of this star graph model and Hough based image matching procedure will be evaluated in section 7.7. There, it will be combined with a broad range of local region detectors.

Contribution 7.2 *This section proposed Star-Hough, a novel image matching scheme based on star graph models and Hough voting for place recognition in changing environments. It is designed to deal with varying landmark positions and high numbers of outlier matchings.*

7.4. Novel superpixel-based local region detectors

The previous section presented methods for *description* (CNN-based descriptor) and *image matching* (the Star-Hough approach). Topic of this section is the first step of Fig. 7.1 - it will present novel local region *detectors* related to superpixel segmentations. There are various algorithmic approaches to provide the spatial image support for place recognition landmarks. Keypoint detectors are a common choice for local region detectors. An established example is the difference of Gaussian (DoG) based scale space

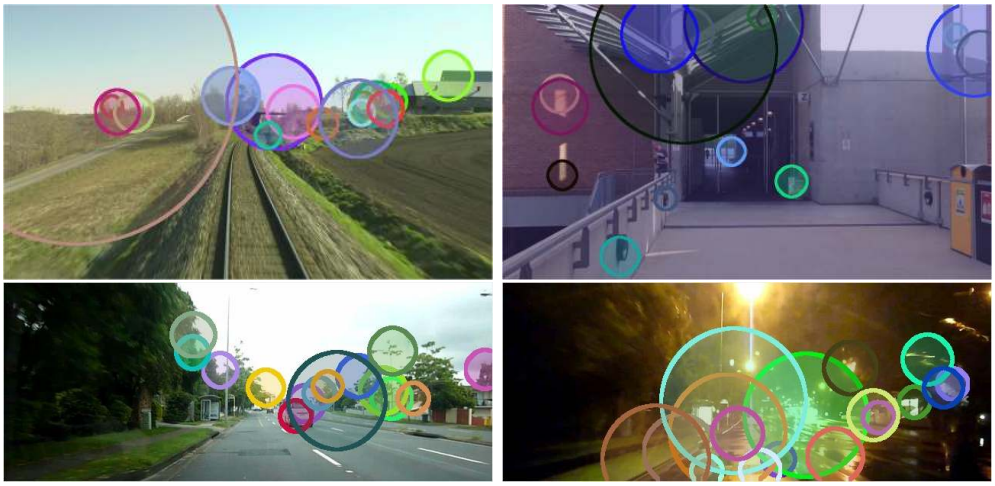


Fig. 7.3.: Example SIFT-DoG-6 regions for images from Nordland spring, GardensPoint day, Alderley day and the same place at Alderley night (subset of 20 regions per image shown).

extrema detector used for SIFT [Lowe, 2004], to which I will refer as SIFT-DoG in the following. Example regions can be seen in Fig. 7.3. The results, referenced in the related work section 5.2, indicate that the established combinations of local feature detectors and descriptors have problems with changing environments, e.g. [Milford and Wyeth, 2012; Valgren and Lilienthal, 2010]. Our own results in [Neubert et al., 2015b] showed that, for example, the SURF based FAB-MAP [Cummins and Newman, 2008] system has severe problems with seasonal changes. The results sections 7.6 and 7.7 will provide an evaluation of both, the detection and the description step of SIFT.

The there presented comparison of local region detectors will also include interest points from Maximally Stable Extremal Regions (MSER). MSER is an established region detector that builds upon segmentations. It provides regions that remain stable over a certain number of thresholds during a region growing segmentation.

Similar to the region growing step in MSER keypoints, superpixel segmentation algorithms provide a broad spectrum of methods to provide spatial image support. In the remainder of this section I want to present the following approaches that incorporate superpixel segmentations in local region detection for place recognition in changing environments:

1. *Stable* superpixel segmentation algorithms can be used directly as local regions detectors. Multiple segmentations of an image can be combined to a segment soup. Some details on this straight forward approach can be found in section 7.4.1.
2. Section 7.4.2 presents a novel approach to use *compact* superpixel segmentations at multiple scales to create a set of local regions that will show to be particularly suited in case of severe appearance changes.

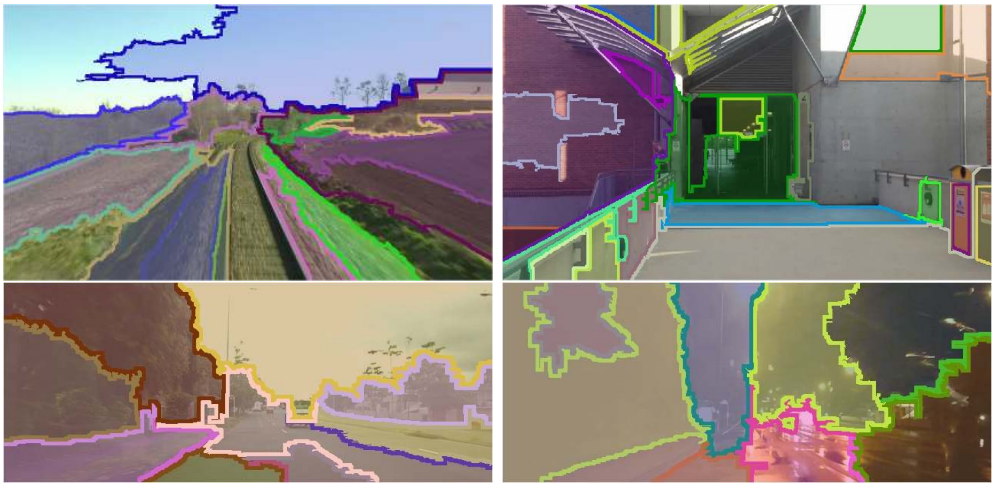


Fig. 7.4.: Example SP-Soup regions for images from Nordland spring, GardensPoint day, Alderley day and the same place at Alderley night (subset of 20 regions per image shown).

3. Navarro et al. [2014] propose to use superpixel segmentations to subdivide SIFT-DoG regions. Section 7.4.3 presents an extension of this approach that uses a segment soup to weight the importance of each part of the local region.
4. Although they are not designed to detect landmarks, object proposal algorithms are related to superpixel segmentation algorithms and included in the set of promising landmark detectors. Details can be found in section 7.4.4.

7.4.1. Using stable superpixel algorithms: Plain superpixels and segment soups

Superpixels can be used straight forward as local regions for place recognition. The superpixel comparison of the first part of this thesis showed that FH, EAMS and QS create segmentations that are stable in the presence of affine image transformations, noise and complex combinations of image changes. Oversegmention algorithms were not specifically designed to select salient features. However, the detected regions are somewhere located between small, local keypoints and the spatial image support for holistic image descriptors. During superpixel segmentation, there is no criteria involved to foster the creation of *descriptive* segments that are repeatedly detected across severe appearance changes.

Besides the direct usage of superpixel segmentations, a superpixel soup (SP-Soup) can be used. The SP-Soup is the combination of multiple segmentations of the same image. These segmentations are created using different parameter sets of the same segmentation algorithm or also different algorithms. The SP-Soup is the union of the sets of superpixels created by each segmentation setup. Example regions can be seen in Fig. 7.4. An example image matching based on a segment soup is illustrated in Fig. 7.2. The parameters that are later used in the experimental comparison are listed in table 7.1.

Algorithm	Parameters
SIFT-DoG	octaves=5, firstOctave=1, peakThresh is varied to obtain about 50 features
MSER	minDiversity=0.8, maxVariation=0.25, delta=5, minArea=200
FH	k=1, minSize=1500
EAMS	hs=8, hr=4, minSize=2000
QS	Ratio=0.5, kernelsize=2, maxDist=40
SP-Soup	3 FH segmentations: k=1, minSize ∈{2500, 5000, 10000}

Table 7.1.: Parameters that are different from defaults for the keypoint, plain superpixel and segment soup approaches. For both keypoint approaches we use implementations provided by `http://www.vlfeat.org/` [Last accessed: 2015-08-14].

7.4.2. Using compact superpixel algorithms: The multi scale superpixel grid

The overview of place recognition approaches in Fig. 5.1 distinguishes in the field of *Local feature based* methods between *Local region detectors* or *Fixedly arranged patches*. For place recognition in changing environments, grid- (or patch-) based methods like the visual front-end of SeqSLAM showed impressive performance in the presence of severe appearance changes as they appear for example between "sunny summer days and stormy winter nights" [Milford and Wyeth, 2012]. The potential benefit is obvious: If no local region detector is involved in the place recognition, it cannot fail to detect corresponding regions.

However, the decoupling of the region detection from the image content by using a fixed grid of image patches, comes at the cost of reduced robustness to viewpoint changes. Dependent on the arrangement of the grid there are *critical* cases - for example if the image content shifts horizontally half the distance of two neighboured patch centres. In this case, both neighboured patches are maximally different from the new patch position.

Is there a solution that combines the advantages of both approaches? A class of algorithms that create patch-like regions that are adapted to the image content has been introduced in the first part of this thesis - *compact* superpixel segmentation algorithms. These algorithms are located somewhere between fixed patches and regions solely detected from visual cues (like the outcome of a typical interest point detector, e.g. the scale space extrema detector used in SIFT). In particular, the evaluation of the compactness parameter of the proposed Compact Watershed superpixel algorithm in section 4.4 showed that there is a continuum of grid-like segmentations ranging from arbitrary shaped segments to a perfectly regular grid segmentation.

Therefore, I want to propose a novel local region detector, named superpixel-grid (SP-Grid), that uses compact superpixel segmentations to provide image regions that are arranged according to a given grid-like structure and adapt their shape to the image content to increase the robustness to viewpoint changes compared to fixed patches.

To implement the SP-Grid approach we have to answer two questions: (1) What is a good initial grid configuration (independent from the subsequent superpixel segmentation)? And (2) How can compact superpixel segmentation algorithms be used to adapt this grid to the image content?

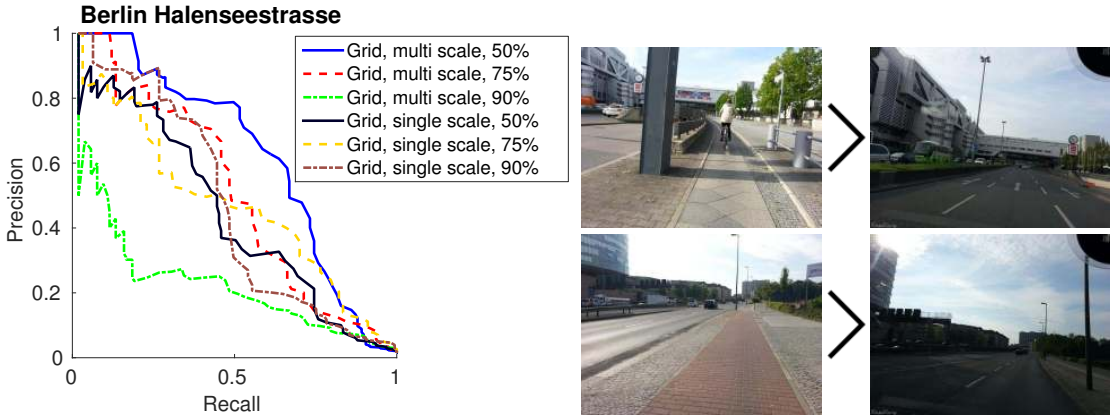


Fig. 7.5.: Comparison of different grid configurations on the Berlin Halenseestrasse dataset (this is the fixed grid, not the SP-Grid). The Berlin Halenseestrasse dataset contains severe viewpoint changes as can be seen by the example image pairs on the right.

The initial grid configuration

I want to clearly point out that finding an *optimal* grid configuration is beyond the scope of this thesis. More information about potential viewpoint changes between images that should be matched and robustness of the used descriptor (in this thesis the CNN-based descriptor) towards viewpoint and scale changes should be involved. However, I want to use a *reasonable* configuration that can deal with the viewpoint changes that typically occur in the place recognition datasets used in this thesis.

To separate the *tuning* of this parameter from the *test* results of the following experiments on place recognition in changing environments, this experiment is conducted on a different dataset, the Berlin Halenseestrasse dataset [Sünderhauf et al., 2015b]. There, the viewpoint changes between the camera mounted behind the windscreen of a car driving on a street and a camera on a bicycle on the cycle lane, more or less alongside the road. To select the grid configuration, I evaluate a small set of promising configurations on this dataset - a grid configuration that works for these severe viewpoint changes is supposed to be a reasonable choice. Examples for corresponding images of the Berlin Halenseestrasse dataset and results of the different grid configurations can be seen in Fig. 7.5.

For reasons discussed in section 7.2, the number of local regions should be about 50. Apart from the individual position of the image patches, the questions whether disjunct patches or overlapping patches should be used and whether all patches should share the same scale are interesting. The set of compared grid configurations comprises single and multi scale grids. For a single scale grid, all patches share the same size. To obtain about 50 features, a (7×7) grid is used. The patches at the image corners are aligned with the image boundaries. The patch size and the positions of the remaining patches can be computed from the image dimensions, the number of features and the overlap in percent of the patch length. To obtain the multi scale grid, the single scale grids of size

(1×1), (2×2), (3×3), (4×4) and (5×5) are combined. This results in a total of 55 regions. Fig. 7.5 shows results of the place recognition experiment using these different grid configurations in combination with the CNN-based descriptor and the Star-Hough image matching. The evaluation reveals a general benefit of the multi scale grid and the best performance for the combination of a multi scale grid and overlap of 50 % of the patch width and height. While this may not be the optimal configuration, it is supposed to be reasonable and will be used in the following.

Adapting an initial grid using compact superpixel segmentations

The proposed SP-Grid region detector builds upon a given grid configuration and a compact superpixel segmentation algorithm. The used compact superpixel algorithm is required to create superpixels in a grid like arrangement, in particular with a defined 4-neighbourhood. However, some of the compact superpixel algorithms provide uniformly shaped segments but lack this neighbourhood. For example, SLIC and the proposed CWS are good superpixel choices for a SP-Grid.

For non-overlapping grid patches, a compact superpixel segmentation can be used directly. To generate $(k \times k)$ regions, the image can be segmented into $(k \times k)$ superpixels and each superpixel becomes an output region. To allow for overlapping patches, a higher resolution of superpixels is computed and superpixels are subsequently grouped into regions. This is illustrated in Fig. 7.6. Starting from a (4×4) segmentation, all (2×2) groups of neighboured superpixels are combined to obtain (3×3) regions.

To create SP-Grid regions at multiple scales, an individual superpixel segmentation is computed for each scale. While it is also possible to create regions at different scales from a single fine grained superpixel segmentation and group different numbers of superpixels (e.g. (2×2), (3×3) and so on) subsequently, using coarse segmentations is supposed to better compensate larger image displacements. Fig. 7.7 shows example images together with the (3×3), (4×4), (5×5), and (6×6) segmentations used to create the overlapping regions as is illustrated in Fig. 7.6.

Although the algorithmic concept is quite intuitive, details on the necessary algorithmic steps tor extracting a set of SP-Grid regions from an image are given in algorithm 4. The first step in line 1 is an initial rescaling of the image. To create segmentations of as few as (6×6) superpixels, the image resolution can e.g. be halved without negative effects. The main loop in line 3 iterates over all scales of the grid. For each scale, the required number of superpixels is computed from the width and height of a region and the overlap of regions, both measured in number of superpixels (lines 5 and 6). For the example regions in Fig. 7.6, the width and height of a region is 2 superpixels and the overlap is 1 superpixel. Changing these numbers can be used to vary the amount of overlap and to generate regions of different sizes from a single superpixel segmentation.

In line 7 the superpixel segmentation is computed. The remaining lines 8-20 of the main loop are dedicated to the computation of the regions using the superpixel label image. For the conducted index arithmetic, it is assumed that the superpixels are arranged grid-like and the labels are in column major order. For each region (x, y), the superpixel label index at the top left corner is computed (lines 10 and 11) and subsequently all

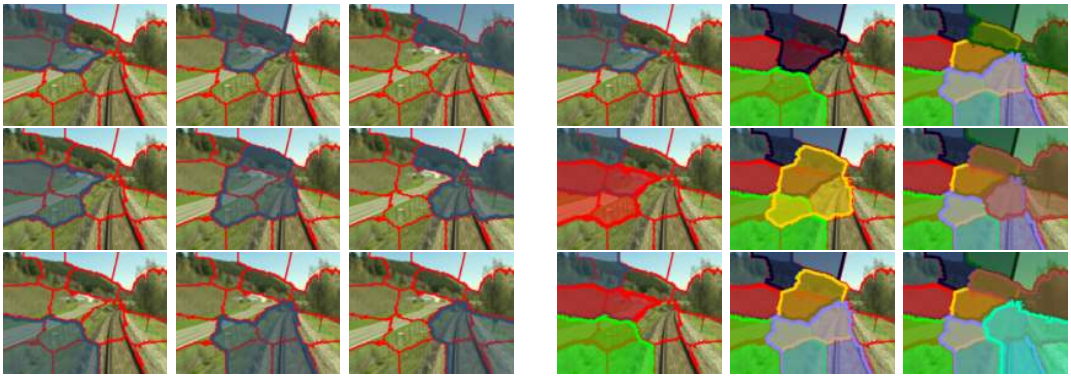


Fig. 7.6.: *(left)* Illustration of the superpixel groupings to obtain the SP-Grid regions for the (3×3) layer. All (2×2) groups of neighboured superpixels are combined and result in 9 regions. *(right)* Visualization of the region overlap: the same regions are added iteratively to a common visualization - each image shows the corresponding region from the left visualization and also all prior regions.

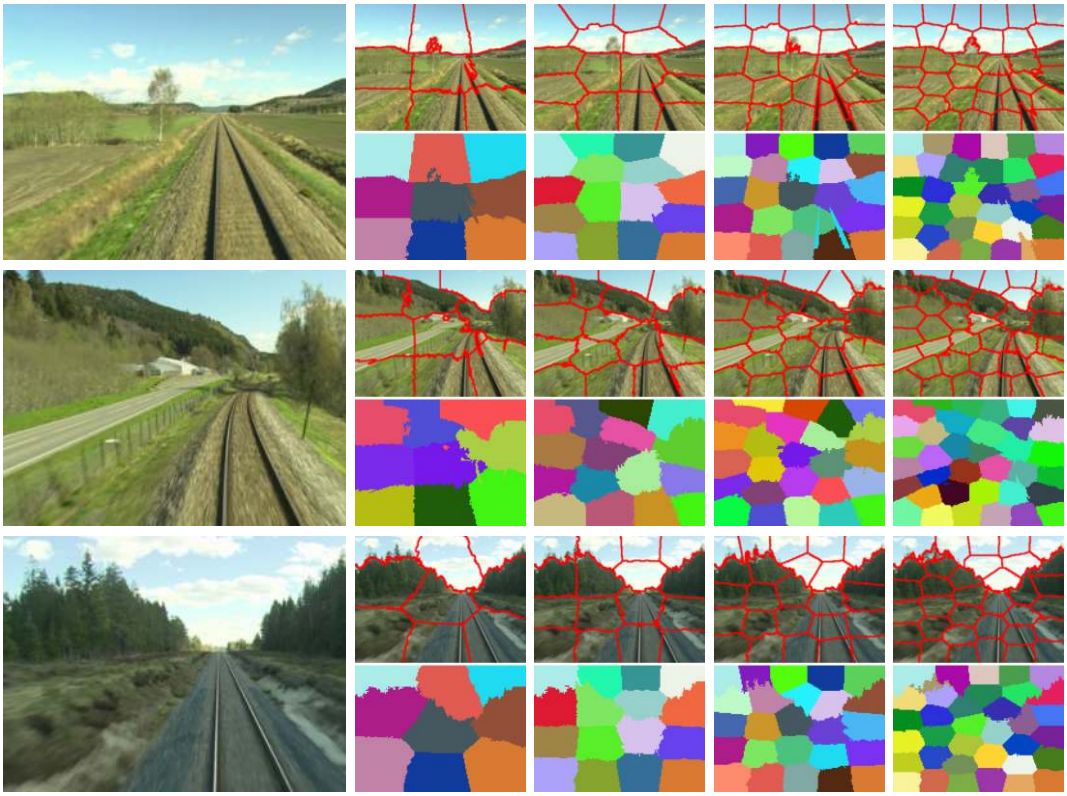


Fig. 7.7.: Example images from Nordland and the resulting SP-Grid superpixel layers.

Algorithm 4: Outline of the algorithmic steps for creating the set of SP-Grid regions.

```
Data: Image
      Grid resolution for each scale: Grid
      Width and height of a region in superpixels: nSpPerRegion
      Overlap of regions in superpixels: spOverlap
Result: Set of regions: R
(Optionally) Resize the image;
Initialize empty set of regions R=∅;
foreach Scale level s of the grid configuration do
    // Get grid resolution for the current scale level
    (nx, ny) = getGridSize(Grid, s);
    // Get size of the superpixel segmentation
    nSpX = nx · nSpPerRegion - (nx-1) · spOverlap;
    nSpY = ny · nSpPerRegion - (ny-1) · spOverlap;
    // Compute the superpixel label image, superpixel labels
    // have to be arranged column major order
    L = performSuperpixelSegmentation(I, nSpX, nSpY);
    // Create regions by collecting (nSpPerRegion × nSpPerRegion)
    // groups of segments
    for y=1:ny do
        for x=1:nx do
            // Get superpixel label coordinates of top left corner
            xTL = (x-1) · (nSPperRegion-spOverlap)+1;
            yTL = (y-1) · (nSPperRegion-spOverlap))+1;
            // Collect all superpixel labels for this region
            labels = ∅;
            for yy=yTL:yTL+nSpPerRegion-1 do
                for xx=xTL:xTL+nSpPerRegion-1 do
                    labels = labels ∪ {(xx-1) · nSpY + yy};
                end
            end
            // Compute region from superpixel labels and include in R
            R = R ∪ {getRegionFromSuperpixels(L,labels)};
        end
    end
end
```

Parameter	Value
Grid resolutions	(1×1), (2×2), (3×3), (4×4), (5×5)
Resulting number of regions	55
Superpixel algorithm	vlSLIC (compactness=30k)
Width and height of a region	2 superpixels
Region overlap	1 superpixel
Image rescaling factor	0.5

Table 7.2.: Default parameters for the SP-Grid

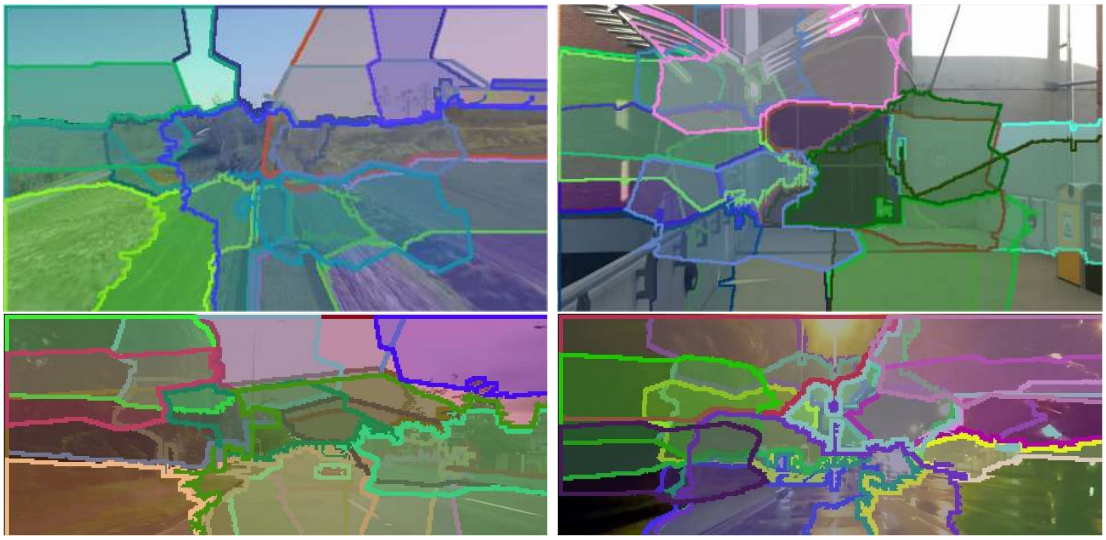


Fig. 7.8.: Example SP-Grid regions for images from Nordland spring, GardensPoint day, Alderley day and the same place at Alderley night (subset of 20 regions per image shown).

($nSpPerRegion \times nSpPerRegion$) labels are collected (lines 12-17). Line 18 collects the resulting regions. They can be obtained from a merging of the assigned superpixel areas or by any other combination (e.g. simply the bounding box containing all assigned superpixels).

The proposed idea of a superpixel-grid-based local region detector can be realized in various ways. While the presented algorithmic listing provides detailed computation steps, there are also some parameters and other important decisions involved, e.g. the choice of the superpixel segmentation algorithm. An extensive evaluation of these degrees of freedom is beyond the scope of this thesis. The choice of the grid configuration is reasoned by a first place recognition experiment in the previous section.

Suitable superpixel algorithms have been evaluated in the first part of the thesis, e.g. SLIC and Compact Watershed are suitable choices. In particular, the vlSLIC implementation showed to be the best grid-like segmentation in terms of segmentation quality and stability. While the proposed Preemptive SLIC implementation is more than 10 times faster than vlSLIC, if runtime really matters, than the novel Compact Watershed algorithm (that runs at about twice the speed of Preemptive SLIC) should be considered. A comparison can be found in the Appendix A.4.

So far, no special efforts have been made to optimize or evaluate the SP-Grid parameters, they are obtained empirically from a very small set of basic experiments. Example regions can be seen in Fig. 7.8. Section 7.7 will include extensive evaluation results of SP-Grid using the default configuration given in table 7.2. An important question in the evaluation will be whether the proposed SP-Grid outperforms the *fixedly arranged grid* at place recognition in changing environments.

7.4.3. Superpixel based weighting of scale space extrema

Superpixels can also be used to post-process keypoint regions to identify important parts of the region. The SIFT *descriptor* incorporates a somehow similar mechanism by Gaussian weighting of the gradient magnitudes to assign higher importance to the inner part of the described region. For CNN-based descriptors, dependent on the used network, the descriptor computations also inherently assign higher importance to the centre of the input image patch - a mechanism that arises from the nature of the training data where the important image content is often in the middle of the training patches. However, for both approaches, the weighting does not incorporate the particular image content or the region that is described.

To reduce the described region to the particularly interesting part, [Navarro et al., 2014] propose to combine SIFT keypoints with superpixels. They use the SIFT-DoG detector to extract local regions from an image and the SLIC superpixel algorithm to segment the image. Each local region detected by SIFT-DoG is subsequently divided into the k superpixels that cover at least 25 % of the region area. An individual SIFT descriptor is computed for each of these k parts by removing all pixels from the patch of the SIFT-DoG region that are not associated to the current superpixel and computing a SIFT descriptor for the resulting patch. This results in k SIFT descriptors for each SIFT-DoG region (even more if the descriptor is oriented and multiple principal orientations occur). According to [Navarro et al., 2014], the descriptor distance of two SIFT-DoG regions with k_1 and k_2 such parts is determined by the minimum distance between any pairing of the $k_1 \cdot k_2$ parts.

This increases the computational complexity of the region comparison and is supposed to increase the visual ambiguity by increasing the number of descriptors and reducing the information that is contained in each descriptor. While the idea of removing distracting information caused by the inclusion of the background in the detected local region is appealing, relying on a single segmentation to separate foreground and background is supposed to be prone to errors.

Therefore, we propose an alternative, straight forward approach on post-processing local regions obtained from scale space extrema in [Neubert and Protzel, 2015b]. Starting from a set of scale space extrema detected using SIFT-DoG, we compute an importance weight matrix of the size of the image patch that is used to compute the descriptor for this region. Examples are illustrated on the right part of Fig. 7.9. Each entry in this weight matrix is intended to represent the probability that the corresponding pixel of the local region belongs to the *foreground* of this region.

There is no clear distinction between foreground and background of a SIFT-DoG region. In particular, the scale of the used inner Gaussian of the difference-of-Gaussians can not directly be applied since the image patches typically involved in the descriptor computation are much larger. As a reminder, for a typical SIFT descriptor, the radius of the described image patch is six times the scale of the inner Gaussian [Lowe, 2004]. However, an intuitive assumption is that if there are homogeneous image regions that intersect with a SIFT-DoG region to a small fraction but largely lay outside the region, then they more likely belong to the background than homogeneous image regions that

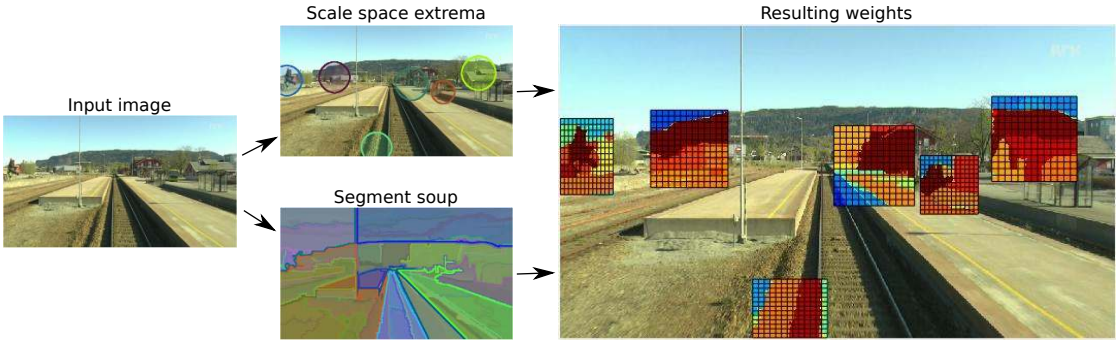


Fig. 7.9.: Superpixel-based reweighting. From the input image, scale space extrema (e.g. SIFT-DoG) and a segment soup are computed. From the overlap of the segments and the scale space extrema, a weighting of each cell of the rectangular image patch that is input for the CNN can be computed. The intention is to get high weight (hot colour) for cells that belong to many superpixels that have high overlap with the region.

are located completely inside the SIFT-DoG region.

In [Neubert and Protzel, 2015b], we propose to evaluate this simple criterion based on superpixels. Instead of relying on a single segmentation, we compute a segment soup that combines multiple segmentations of the same image as described in section 7.4.1. For each segment of the soup we compute the intersection over union (IoU) as described in section 6.2.3. As a reminder, given a set A comprising the pixel of the SIFT-DoG region and a set B with the pixels of the superpixel, the IoU is computed as

$$IoU(A, B) = \frac{|A \cap B|}{|B \cup A|} \tag{7.3}$$

The values of the weight matrix are computed as the mean IoU of all superpixels that comprise the corresponding pixels of the local region patch. If a pixel is contained in many superpixels that overflow the local region boundaries, the resulting weight is smaller than for pixels that belong to superpixels that are in accordance with the local region.

These weights can later be used to adapt the importance of descriptor dimensions - the part of the descriptor that describes the foreground should be weighted higher than the parts that describe the background. As a reminder, the feature vector obtained from the used network layer is a tensor with a spatial structure. Each dimension of the descriptor has a receptive field in the described image patch. The theoretical size of the receptive field can be computed from the network parameters. However, Zhou et al. [2014] found the practically relevant receptive fields to be much smaller. For a $(k \times k)$ spatial layer of the tensor representing the input patch, we assume each of the values to represent the corresponding area of a non-overlapping $(k \times k)$ grid tessellation of the input patch. This tessellation is shown as black grid in Fig. 7.9. Given this mapping of superpixel-based weights to dimensions of each layer of the CNN tensor and the particular interesting case of comparing CNN based descriptors using the *cosine*

distance (which is basically a scalar product), these weights can simply be multiplied element-wise with the corresponding dimensions of the CNN descriptor. An evaluation of this approach will be topic of section 7.7.3.

7.4.4. Object proposals

The generation of object proposals (or object candidates) is an established application of superpixel segmentations. As already mentioned in the introduction to superpixel segmentations in section 2, object proposals recently replaced the exhaustive search based on the sliding window approach in many successful object detection pipelines. For example the winning and second place object detection entries of the ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC13) [Russakovsky et al., 2015] both build on object proposals [van de Sande et al., 2014; Wang et al., 2013], as well as the 2014 winning entry, Region-based Convolutional Network (R-CNN) [Girshick et al., 2014].

Many of the available object proposal algorithms build directly on superpixel segmentations or use related techniques. Although they were not designed for usage as landmarks for place recognition, they can somehow be seen as the next higher level above superpixels in the hierarchy from raw pixels to objects. Therefore, we included them in the set of promising local region detectors for place recognition in changing environments in [Neubert and Protzel, 2015b]. Very recently, [Sünderhauf et al., 2015b] also used an object proposal algorithm, EdgeBoxes [Zitnick and Dollar, 2014], for place recognition in changing environments. The following object proposal algorithms are included in the set of potential landmarks related to superpixel segmentations:

1. Randomized Prim's (RP) [Manen et al., 2013] uses a greedy algorithm for computing sets of FH superpixels that are likely to occur together.
2. Objectness [Alexe et al., 2012] is one of the earliest approaches and samples and ranks a large number of windows per image according to their likelihood of containing an object. This likelihood is based on multiple cues derived from saliency, edges, superpixels, colour and location.
3. EdgeBoxes [Zitnick and Dollar, 2014] use the set of edges also used for superpixel computation based on [Dollár and Zitnick, 2013] to compute a simple box objectness score. Edges are grouped into contours and the objectness score is the difference between edges on contours completely inside a box and those crossing the box boundary.
4. Multiscale combinatorial grouping (MCG) Arbelaez et al. [2014] uses a fast implementation of normalized cuts superpixels and computes a bottom-up hierarchical image segmentation to obtain object proposals. Similar to EdgeBoxes, MCG also incorporates the fast edge computation presented in [Dollár and Zitnick, 2013].

Object proposal algorithms were designed as preprocessing steps for object detection to reduce the number of sliding windows that are classified to a much smaller set of promising candidates. However, the number of typically used object proposals is in the

Algorithm	Parameters	Implementation
Objectness	n=50	`http://groups.inf.ed.ac.uk/calvin/objectness/`
RP	q=200	`https://github.com/smanenfr/rp#rp`
EdgeBoxes	BSDS model	`https://github.com/pdollar/edges`
MCG	"fast" version, maxOverlap=0.9	`https://github.com/jponttuset/mcg`

Table 7.3.: Source of the used implementations and parameters that are different from defaults for the object proposal algorithms. All websites last accessed 2015-08-15.

Fig. 7.10.: Example MCG regions for images from Nordland spring, GardensPoint day, Alderley day and the same place at Alderley night (subset of 20 regions per image shown). Each region is a combination of a set of superpixels.

order of several hundreds or thousands. For the application as local landmarks in the proposed place recognition system, a smaller number of about 50 regions is used. Either by using the object proposals with the highest score of the proposal approach or, if there is no such score available, we greedily select regions with low pairwise overlap. The used implementations and the parameters are listed in table 7.3. Example regions for MCG and EdgeBoxes object proposals can be seen in Fig. 7.10 and 7.11. Sections 7.6 and 7.7 will present results on region repeatability and place recognition performance.

Contribution 7.3 *This section proposed the SP-Soup, SP-Grid and superpixel based reweighting as novel approaches to incorporate superpixel segmentations in local region detection. As an additional approach, related to superpixel segmentations, it further proposed the application of readily available object proposal algorithms for landmark detection.*

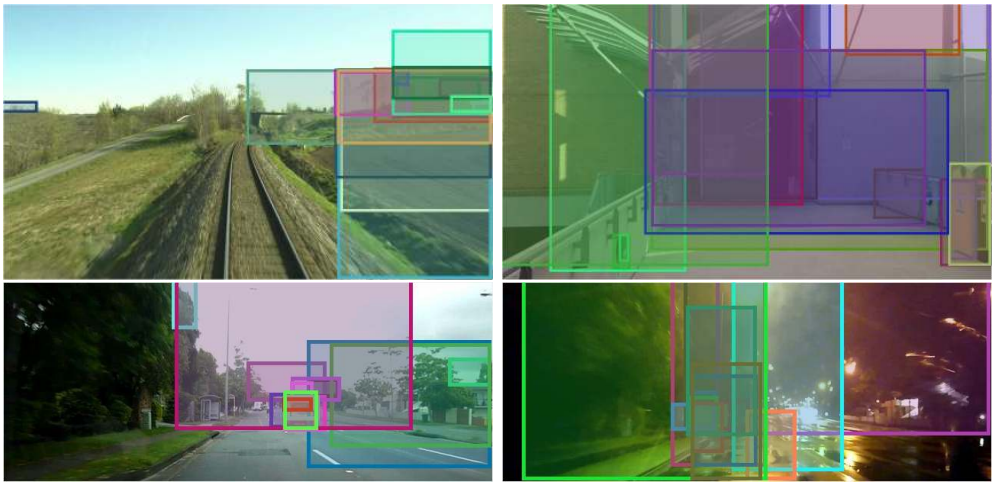


Fig. 7.11.: Example EdgeBoxes regions for images from Nordland spring, GardensPoint day, Alderley day and the same place at Alderley night (subset of 20 regions per image shown). Although obtained from edge cues, the EdgeBoxes are rectangular regions - hence the name.

7.5. How to measure the repeatability of local region detectors

The detection of local regions is the first step in the processing chain illustrated in the initial Fig. 7.1 of this chapter. If this step fails, subsequent processing steps cannot be expected to remedy the resulting problems in this simple feed forward pipeline. Therefore, region detectors that can repeatedly detect well localized image features are necessary. For the particular application of place recognition, the following criteria are supposed to be crucial:

1. location repeatability and accuracy
2. selection of salient regions in the context of the dataset
3. a distribution in the image that supports place recognition
4. reasonable size for the subsequent descriptor computation

Before evaluation of the proposed superpixel based local region detectors in place recognition experiments, I want to evaluate them independently of any descriptor or matching step. The idea is to evaluate an upper bound of the place recognition performance that could be achieved with a particular detector - if it would be combined with a perfect descriptor and matching scheme. However, from the above enumeration of requirements on the local region detectors for application on place recognition, the selection of salient regions and the suitable distribution of regions in the image are rather hard to evaluate independently of a particular place recognition task. For the first criterion, the localization repeatability and accuracy, Mikolajczyk et al. [2005] provide a suitable and established metric. In [Neubert and Protzel, 2015b], we propose a novel criterion that builds upon this metric but further allows for incorporation of the size of

the detected regions. These two metrics will be presented in the following two sections.[2]

7.5.1. Size-normalized IoU

In their influential work, Mikolajczyk et al. [2005] provide a methodology for evaluating the repeatability of feature detectors in terms of the *localization accuracy*. In principle, their approach is the following: Given two images I_A, I_B of the same scene and the transformation T_{AB} between these images (i.e. the ground truth optical flow), they detect features in both images and evaluate their overlap. Given the sets of pixels P_A, P_B constituting the spatial image support of two features f_A^i from I_A and f_B^j from I_B, the overlap is computed as the intersection over union (IoU):

$$IoU(f_A^i, f_B^j) = \frac{|T_{AB}(P_A^i) \cap P_B^j|}{|T_{AB}(P_A^i) \cup P_B^j|} \tag{7.4}$$

For example, f_A^i, f_B^j may be SIFT features, P_A^i, P_B^j are all pixels of the corresponding ellipses, and $T_{AB}(P_A^i)$ are the pixels of the SIFT ellipse of f_A^i transformed to the image space of I_B.

An important step in their evaluation is a normalization with respect to the features' sizes. They show that the performance of a region detector in their measure can be improved by simply increasing the size of the detected regions. Therefore, they compute for each feature comparison $IoU(f_A^i, f_B^j)$ a rescaling factor that normalizes the size of f_A^i to a given diameter and applies this rescaling to both features. This makes different region detectors comparable. However, the authors clearly point out that such a normalization should not be applied for real applications of these feature detectors.

7.5.2. Size-sensitive IoU

Our experiments with image features for place recognition in changing environments support the hypothesis of a real dependency between the features' sizes and the resulting place recognition performance. For example, typical seasonal changes induce severe appearance differences at small scales: leaves change their shape and colour, snow and ice cover meadows, rain modifies the reflective properties of surfaces and so on. In contrast, the appearance of coarser structures is supposed to be more persistent: mountains, buildings, streets and trees change their detailed appearance but are likely to remain at the same global configuration.

To evaluate the fourth criterion of the above listing, the size of the provided local regions, we present a modified criterion in [Neubert and Protzel, 2015b]. There, we propose to use the same IoU criterion as Mikolajczyk et al. [2005], but in combination with another normalization procedure. The objective is to measure the feature matching performance that can be achieved with a given feature detector incorporating the size of the regions. Given two sets of detections $F_A = \{f_A^1, f_A^2, ...), F_B = \{f_B^1, f_B^2, ...)$, from two images with known optical flow T_{AB}, we transfer all features to the same image space

[2]Sections 7.5.1 and 7.5.2 are published in very similar form in [Neubert and Protzel, 2015b].

and assign to each feature of F_A the maximum IoU between this feature and a feature from F_B:

$$IoU_{max}(f_A^i, F_B) = \max_{f_B^j \in F_B} IoU(f_A^i, f_B^j) \tag{7.5}$$

These pairs are the combinations with the highest rate of common pixels and thus (approximately) the highest rate of common world points. This is supposed to represent an upper bound for what could be found by matching based on real image descriptors. If there are no feature pairs with sufficient overlap detected, the best feature descriptor will fail to match the images.

As a measure for the repeatability, we can then simply count for a set of image pairs the average rate of features whose IoU_{max} exceeds a threshold t. More formally the average of:

$$R_t(I_A, I_B) = \frac{1}{|F_A|} \cdot |\{f_A^i \in F_A : IoU_{max}(f_A^i, F_B) > t\}| \tag{7.6}$$

To evaluate the repeatability of a feature detector, we compute a curve showing the average rate of detections R_t for a set of thresholds t.

Since we do not apply the normalization of [Mikolajczyk et al., 2005] on the feature size, there is a bias towards large regions in the IoU. On the one hand, this effect is intended to obtain features whose appearance is less affected by the environmental changes, on the other hand we need to prevent that feature detectors gain significant performance just due to the region size. Therefore, we evaluate the amount of overlap of features between *corresponding* scenes relative to the average overlap when comparing *random pairs* of scenes. The resulting curves presented in the results section show the average rate of detections per corresponding image exceeding the expected rate of "detections" in random images. Thus, artificial enhancement of the pure overlap performance that is not related to real image features also enhances the overlap on random scene pairs and does not affect the measure. And in contrast, if larger regions really increase the rate on common world points compared to random image pairs, then this results in an improved performance measure. For this normalization, all regions have to be described in normalized image coordinates if the images have different dimensions.

Contribution 7.4 *This section presented a novel, size sensitive measure for the evaluation of local region detectors.*

7.5.3. Benefit of the novel metric - Evaluating the scale of SIFT-DoG

I want to clearly point out, that the second proposed metric is not intended to replace the measure by Mikolajczyk et al. [2005]. The localization accuracy, as measured by Mikolajczyk et al. [2005], can be regarded as a *necessary* property of a local region detector. Although localization accuracy alone is not a *sufficient* condition (e.g. think of the discriminative power of the regions), we can not expect a local region detector to perform well in a place recognition task without detecting repetitive regions.

Good performance on the proposed criterion is not a *necessary* property of a region detector - in particular, since it measures the repeatability of regions from corresponding images compared to random images. All region detectors that use fixedly arranged patches will not exceed a value of zero in this measure due to this normalization. However, the proposed adapted measure is intended to provide additional insights on the influence of the size of the regions on the overlap of the detected regions that are hidden by the normalized size in the original metric.

I want to illustrate the difference of the IoU metric of Mikolajczyk et al. [2005] and the novel size-sensitive variant on results of differently rescaled versions of the SIFT-DoG local region detector.

Why are there differently scaled versions? For the typically used combination of the SIFT DoG-based detector and the SIFT descriptor, the size of the image patch that is used for the descriptor has to be determined. The DoG based scale space extrema detector provides a radius r that is obtained from the standard deviation of the smaller (or inner) Gaussian. In the here presented experiments, the resulting region with this radius r is called SIFT-DoG-1. In [Lowe, 2004], the spatial support for the SIFT descriptor is rescaled by factor 6, in particular it is the rectangular patch with edge length $2 \cdot r \cdot 6$. In the here used naming scheme, this "original" local region detector used in SIFT is called SIFT-DoG-6. Based on the assumption, that place recognition in changing environments in combination with sophisticated CNN descriptors may benefit from larger regions, SIFT-DoG-12 and SIFT-DoG-15 are interesting candidates for a comparison.

In this experiment, I use a 100 image subset of the Nordland spring-winter training dataset of section 5.1.1 as a basis for the landmark repeatability evaluation. To deal with the remaining synchronization problems (illustrated in the appendix Fig. A.12), the image pairs are manually synchronized by sampling the video at full framerate and selecting the best matching frame by manual, visual inspection of the overlayed edge images. The original dataset consists of image pairs showing the same scene, from the same viewpoint at different seasons. To prevent, in particular, superpixel segmentation algorithms from exploiting the image alignment, the first image is cropped on the left side and the second image on the right side. This is equivalent to a shift of about 12.5 % of the image width.

Fig. 7.12 illustrates the difference of the IoU metric of Mikolajczyk et al. [2005] and the novel size-sensitive variant. Since the two IoU based metrics differ in the normalization procedure, the non-normalized IoU rate of equation 7.6 is also included.

The left column shows the detection rate without normalization. There is a strong preference of larger regions for the whole spectrum - although we can not expect larger regions to be better in general. This is in accordance with the observations of Mikolajczyk et al. [2005]. Therefore, they propose the size-normalized metric whose results are shown in the middle of the figure. There, all SIFT-DoG variants perform identically, independently of the rescaling factor. However, there is of course an influence of the size of the region on the performance of the local region detector in combination with a *descriptor*. The results of the proposed size-sensitive measure of the previous section 7.5.2 can be seen in the right part of the figure. The proposed metric can identify the

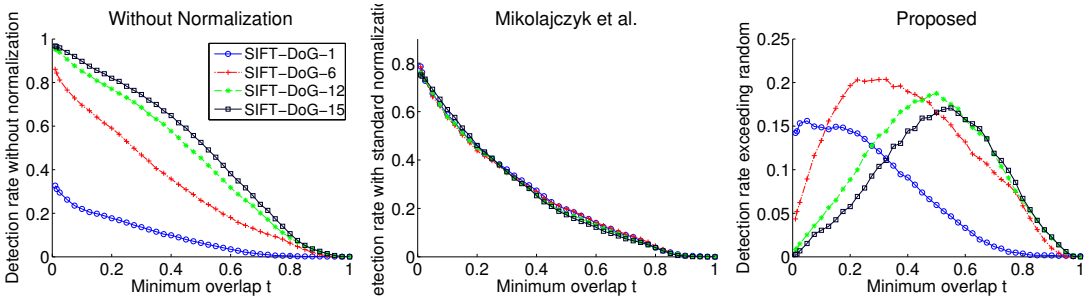


Fig. 7.12.: Results of SIFT-DoG using three overlap criteria. The overlap criterion without normalization prefers large segments. The normalized criterion presented by Mikolajczyk et al. [2005] solves this problem but hides the real existing amount of overlap. The proposed measure in the right column evaluates the real overlap that exceeds random by normalization with random image pair comparisons.

influence of the scale of the regions on different amounts of overlap. As a reminder, it provides for each overlap threshold the detection rate that is not caused by random but by the repeated selection of image regions. For the small SIFT-DoG-1 regions, there are almost no regions with high overlap. The default SIFT-DoG-6 provides the broadest range of overlaps for which high detection rates are provided. However, the further enlarged DoG regions, i.e. SIFT-DoG-12, provide even higher detection rates for large overlaps - at the cost of detections with low and mid amounts of overlap.

Simply rescaling regions is expected to result in decreased robustness to viewpoint changes - an issue that can be approached using the superpixel-based reweighting proposed in section 7.4.3. There is no dataset at hand providing seasonal changes, severe viewpoint changes *and* ground truth optical flow for evaluation including viewpoint changes based on these metrics. To evaluate these combined influences section 7.7.3 will evaluate the place recognition performance of these different SIFT-DoG variants

Result 7.2 *The proposed IoU-metric can provide insights on local region detectors at different scales.*

Result 7.3 *SIFT-DoG-12 is a promising alternative to the default SIFT-DoG-6 for the later experiments on place recognition in changing environments.*

7.6. Experimental results on the repeatability of local region detectors in changing environments

In this section I want to evaluate the presented local region detectors independently of a particular place recognition experiment. This is based on the size-normalized and size-sensitive overlap metrics presented in the previous section 7.5. In particular, the detectors are evaluated independently from a descriptor - purely based on the overlap of detected regions. The overall questions will be: What are good region detectors for

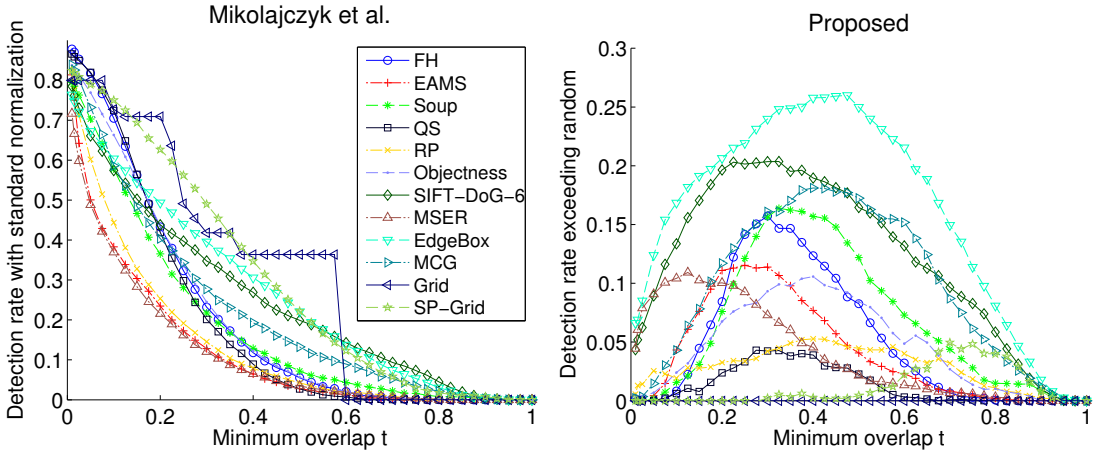


Fig. 7.13.: Results of the comparison using the two overlap criteria. *(left)* The size-normalized criterion by Mikolajczyk et al. [2005]. *(right)* The proposed size-sensitive measure.

changing environments? In particular, are the superpixel based approaches of section 7.4 competitive? The comparison includes the following local region detectors:

- the two keypoint based detectors SIFT-DoG-6 and MSER
- the plain superpixel algorithms FH, EAMS, and QS, and the superpixel soup SP-Soup
- the fixedly arranged Grid and the superpixel grid SP-Grid
- the object proposal algorithms RP, Objectness, EdgeBoxes and MCG

As a reminder, all algorithms are configured to create about 50 regions per image using the settings and parameters provided in section 7.4, i.e. tables 7.1, 7.2 and 7.3

Comparison of different local region detectors

Again, the used dataset are the shifted Spring-Winter images (the same that were already used in the first experiments using these measures in section 7.5). The resulting curves for the compared region detectors can be seen in Fig. 7.13. SIFT-DoG-6 is a region detector known to perform well in non changing environments. The place recognition results on the Nordland Spring-Winter dataset presented in the previous section 7.7.1 in combination with the CNN descriptor showed a reasonable performance of this detector for this kind of appearance change. Thus, SIFT-DoG-6 can be used as an indicator for a reasonable performance. For example, for this type of environmental change, MSER keypoints showed to be much less repetitive in this experiment.

From the plain superpixel segmentation based approaches, FH performs best and is thus a reasonable choice for inclusion in the segment soup - which is the combination of three FH segmentations as described in section 7.4.1. The SP-Soup provides more regions with high overlap than the plain superpixel segmentations. In particular, the

size-sensitive measure shows the benefit of the segment soup over single segmentations (to be clear, both have the same number of regions, the regions in the single segmentation are simply smaller).

From the comparison of the object proposal algorithms, EdgeBoxes and MCG perform comparable to SIFT-DoG-6 at the *necessary* size-normalized criterion. For Objectness we use the 50 features with the highest objectness-score. For RP we force the regions to be as different as possible to obtain the 50 features. This may prevent them from providing better repeatable features. EdgeBoxes clearly outperforms all other detectors at the size-sensitive measure. Presumably, this is to some extent due to the distribution of the EdgeBoxes in the image. They tend to group around a few salient image parts which results in high overlap for corresponding image pairs and low overlap for random image pairs where these few salient image parts may be located very differently. The influence on the resulting place recognition performance will be topic of section 7.7.4.

The Grid curve shows a characteristic shape. The uniform plateaus and sharp declines are due to the fixedly arranged boxes (at multiple scales) that provide the regions. Dependent on the shift between the compared images, this shape is expected to look quite different. In the absence of any shift, the grid creates perfect overlapping regions. However, this is not what is expected for real world application, including viewpoint changes. Dependent on the amount of shift, potentially, there will be no regions with high overlap as is shown in the experiment with 12.5 % shift.

The proposed adaptation of the fixed grid with a superpixel segmentation, SP-Grid, described in section 7.4.2 significantly smooths the curve of Grid - in particular we can hope for regions with high overlap for *all* reasonable amounts of shift. The resulting benefit for place recognition of using the SP-Grid instead of the fixed Grid will be evaluated in section 7.7.2. Before comparing Grid and SP-Grid, I want to evaluate the set of promising region detectors from this comparison on datasets with different amounts and types of environmental change.

Result 7.4 *From this comparison, SIFT-DoG-6, SP-Soup, SP-Grid, EdgeBoxes and MCG are promising candidates for local region detectors in changing environments.*

The influence of changing environments

The reasonable performance of the SIFT-DoG-6 detector in combination with a CNN descriptor shown in Fig. 7.17 illustrated that this *detector* can be used even in the presence of changing environments (as a reminder, the combination with the SIFT *descriptor* failed). However, this is supposed to depend on the severity of the appearance changes. In this section, I want to approach an answer to the question, how the performance of local region detectors depends on the type or severity of the environmental change.

The evaluation is based on datasets with varying amount of environmental change obtained from stationary webcams. A number of webcams from all over the world provides their imagery online. They observe and record the visual appearance of a variety of outdoor scenes over long periods of time. The used datasets are created from the Webcam Clip Art dataset presented in [Lalonde et al., 2009]. The data available

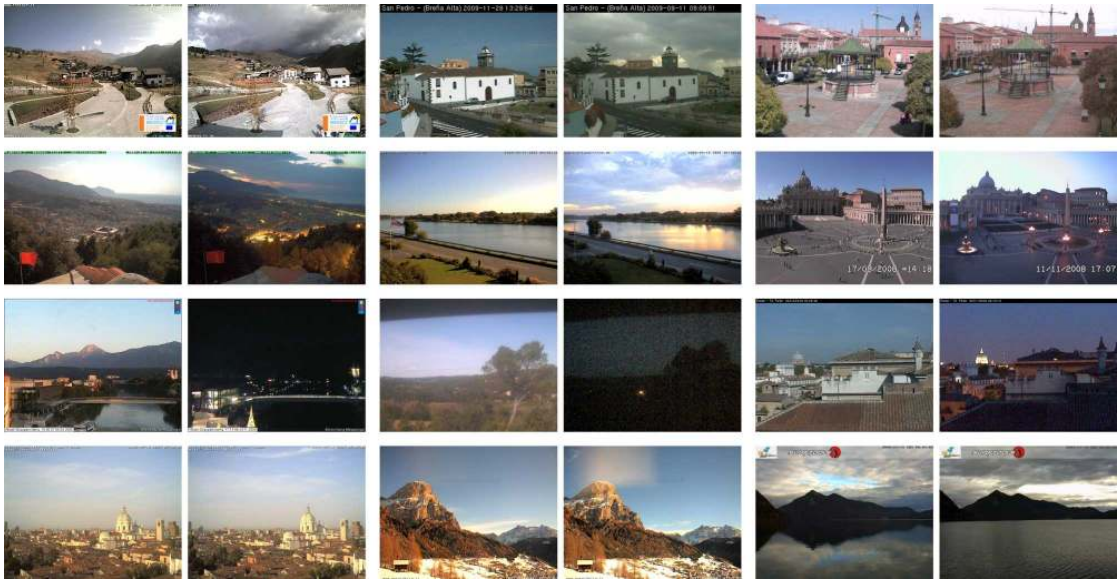


Fig. 7.14.: Example image pairs of the webcam dataset. The rows from top to bottom: *Sunny-Cloudy, Day-Nightfall, Day-Night* and *Non changing environment* datasets.

online[3] is a subset of the data used in the paper. It provides images from 54 webcams, on average captured about every 30 minutes over the course of more than six months, resulting in about 10,000 images per webcam.

To get image pairs that show the same place with a certain environmental change, we[4] selected example images according to the labels *sunny, cloudy, nightfall* and *night*. Since the IoU based evaluation requires information about the ground truth optical flow in the images, we can only use static webcams and apply synthetic image modification for which the resulting ground truth pixel wise associations are known. Thus, images for which the webcam moved from its initial position are removed (what happens occasionally over the course of several months). The evaluation is based on five datasets:

1. **Sunny-Cloudy** contains 50 places, each in sunny and cloudy weather conditions.
2. **Day-Nightfall** contains 50 places using the sunny condition and imagery from dusk or dawn.
3. **Day-Night** contains the 39 places for which there are night images. This dataset is smaller since some webcams are switched off at night or provide more or less useless black images during night.
4. **Spring-Winter** is the same Nordland validation data subset used above.
5. **Non changing environment** contains 50 places shown in consecutive images from the database. On average, the time delay between the images is 30 minutes. However, while each pair is taken from similar environmental conditions, the

[3] `http://graphics.cs.cmu.edu/projects/webcamdataset/` [Last accessed: 2015-08-14]
[4] Thanks to the two interns Max Becker and Lukas Voigtländer for their help on collecting the data.

conditions between different pairs vary.

Fig. 7.14 shows example image pairs of the webcam datasets. While the shown image pairs share the same viewpoints, for the conducted experiments the viewpoints are synthetically modified in the same way as the Norland images before (shifting the images horizontally followed by cropping). It can be seen, that the *Day-Nightfall* and *Day-Night* datasets share some characteristics. The *Non changing environment* contains image pairs with varying amounts of appearance change - for each place the change that happened in the 30 minutes between the two images were captured. While the position of the webcams is static, there are still some moving objects that are not regarded in the used ground truth. The amount of moving objects depends on the type of scene. Since many webcams show landscapes, there are only few dynamic objects of relevant size in the datasets.

Fig. 7.15 shows the performance of the set of promising local regions detectors on different datasets with increasing severity of the environmental change. It is obvious that there is a large influence of the type of environmental change on the performance of the local region detectors. In general, the performance of the evaluated detectors decreases from non changing environments over Sunny-Cloudy and Spring-Winter to Day-Nightfall and Day-Night. Fig. 7.16 shows the comparison of the results for all detectors on the simplest (Non changing environment) and the hardest (Day-Night) environmental change. It can be seen, that SIFT-DoG-6 performs superior in the non changing environment but it is strongly affected by the severe appearance changes occurring during day-night cycles. The regions detected by SP-Grid have a strong prior on their location and size in the image. Thus, they are much less affected by environmental changes. While SP-Grid is supposed to perform inferior for non changing environments, for place recognition in changing environments with reasonable assumptions about camera motion, it seems to be a promising alternative.

Result 7.5 *SIFT-DoG-6 is a reasonable detector for non changing environments. With increasing severity of the changes, its performance drops. For severe appearance changes, in particular the regions generated by the proposed SP-Grid are a promising alternative.*

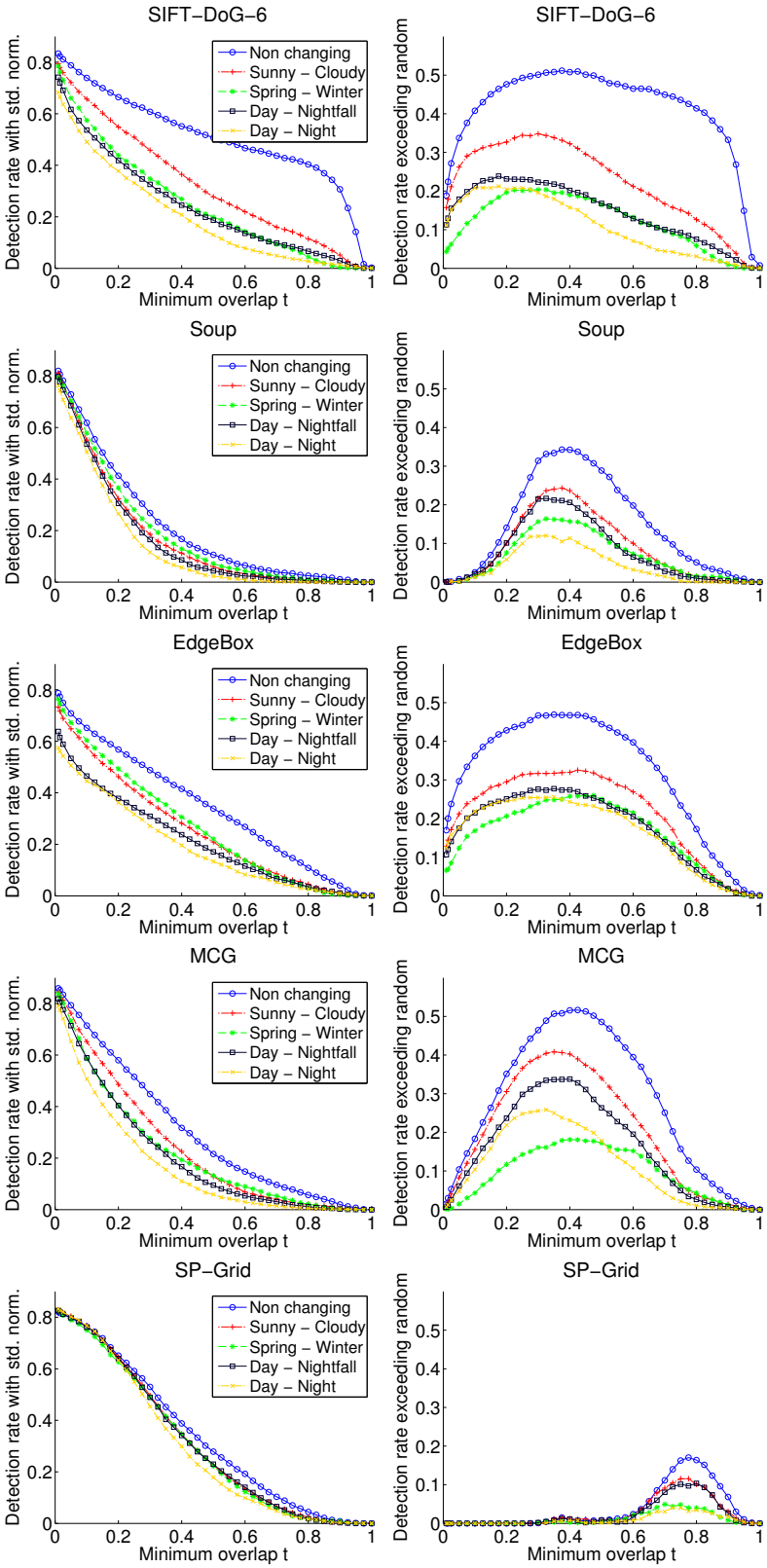


Fig. 7.15.: Detection rates for promising local region detectors and different types of environmental changes.

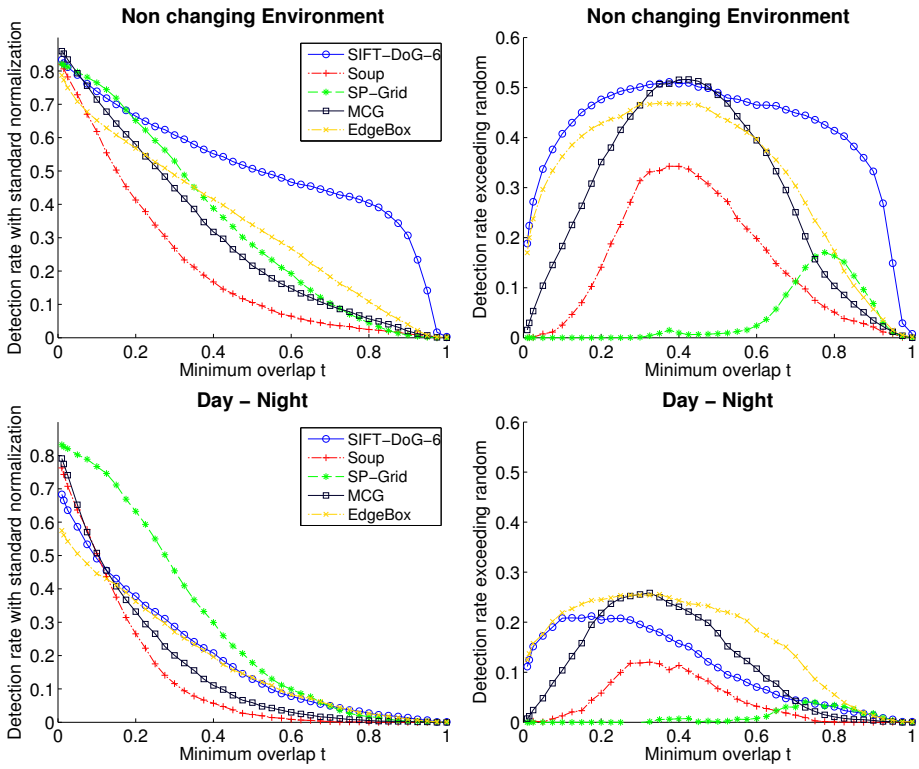


Fig. 7.16.: Comparison of different detectors on the same dataset. These are the same curves as in Fig. 7.15 but grouped with respect to the datasets.

7.7. Experimental results on place recognition in changing environments

The experimental results presented in the following are intended to answer the following questions:

Section 7.7.1	Is there a benefit of the proposed place recognition scheme based on local region detectors, CNN based descriptors and the Star-Hough image matching scheme?
Section 7.7.2	Is there a benefit from using the proposed superpixel grid over the fixed grid?
Section 7.7.3	What is the influence of the proposed superpixel based importance weighting of regions obtained from scale space extrema?
Section 7.7.4	What is the overall place recognition performance of the proposed system on different datasets showing changing environments?

The evaluation of place recognition experiments is based on precision-recall curves similar to the SP-ACP evaluation from the previous chapter 6. Details on the computation of the precision-recall curves can be found in section 5.3. However, there are some no-

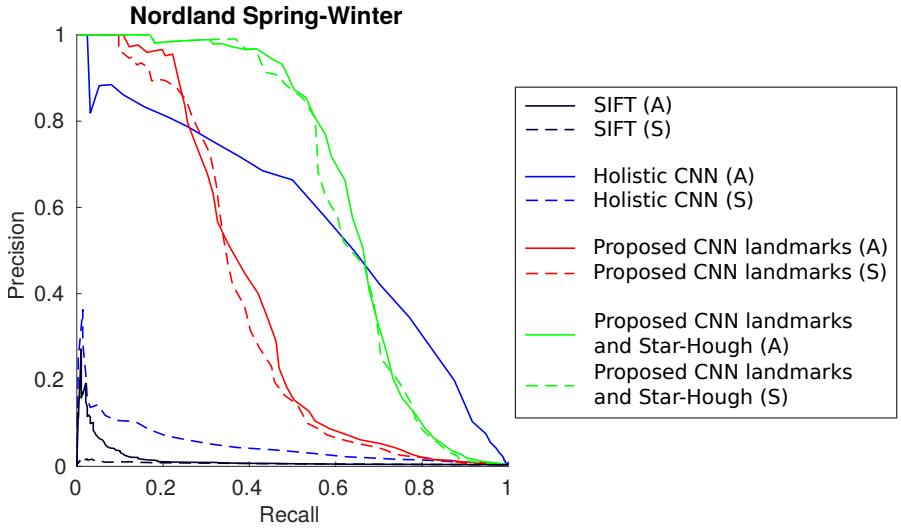


Fig. 7.17.: The place recognition performance improvements achieved by using two of the contributions of this chapter: The local region detector+CNN based landmarks and the proposed Star-Hough matching scheme. The local region detector for the landmarks is SIFT-DoG-6, the same as for SIFT. (A) original, aligned images, (S) images shifted by 10 % of the image width.

table details on the evaluation based on these curves and the comparability of results that will be mentioned in form of additional remarks in this section (i.e. notes 7.1 and 7.2).

7.7.1. Influence of the proposed landmark matching scheme

In the first set of place recognition experiments, I want to demonstrate the benefit from two rather general contributions presented in this chapter: the local region detector + CNN based landmarks and the Star-Hough image matching scheme. Fig. 7.17 shows results on the Nordland Spring-Winter *test* dataset. Similar to the final results of SP-ACP on the full Nordland journey presented in the previous chapter 6, a novel, unique place is expected each 10 frames of the test sequence - this way, matching of more nearby frames, that may be found by scale invariant region detectors, is not penalised. The influence of this parameter will later be seen in experiments in section 7.7.2. Further, frames where the train stopped or passed through a tunnel were removed.

The solid black curve in Fig. 7.17 shows the performance of image matching based on SIFT keypoints - the combination of SIFT-DoG-6 regions and SIFT *descriptors*. As repeatedly reported in the literature, this approach has severe problems with changing environments. The solid blue curve shows the performance of a single holistic CNN descriptor as described in [Sünderhauf et al., 2015a] - the performance is clearly superior to the SIFT keypoint matching. However, the performance of the holistic approach drops significantly in the presence of viewpoint changes. The dashed curves show the performance on shifted images of the Norldand dataset. Shifting and cropping each image by 10 % of the image width creates an artificial viewpoint change (more exactly

a rotation). This viewpoint change causes the holistic CNN descriptor to fail.

Therefore, I proposed to use local region detectors together with CNN-based descriptors to create local landmarks in this chapter. This idea is also presented in [Neubert and Protzel, 2015b] and has congruently and independently been developed by Sünderhauf et al. [2015b]. The red curves show the performance of an established local region detector (SIFT-DoG-6) with the CNN descriptor using equation 7.1 to compute image similarities from descriptor distances without using the landmarks spatial arrangement (i.e. without Star-Hough). It can clearly be seen, that this approach is much more robust towards the artificial viewpoint change compared to the holistic approach.

Result 7.6 *The proposed combination of a local region detector and CNN based descriptors provides reasonable results in the presence of severe appearance changes and (artificial) viewpoint changes.*

The computation of the image similarity from the novel landmarks used for the red curves is very similar to what is done in Sünderhauf et al. [2015b]. Additionally incorporation of the proposed Star-Hough image matching scheme presented in section 7.3 yields a further significant performance increase shown in the green curves.

Result 7.7 *The proposed Star-Hough image matching scheme can further significantly increase the performance.*

Since the green (CNN descriptor) and the black (SIFT descriptor) curves are computed based on the same local regions (SIFT-DoG-6) and the same matching scheme, the comparison shows that place recognition between spring and winter based on SIFT keypoints (detector and descriptor) fails at the *description* step. Before we proceed with further place recognition experiments in section 7.7.2, the following section will evaluate the robustness of the *detection* step of SIFT (SIFT-DoG-6) and other local region detectors (i.e. the superpixel based) towards environmental changes.

7.7.2. Is the superpixel grid better than a fixed grid?

Section 7.4 presented and motivated the superpixel grid (SP-Grid) as a novel local region detector that creates regularly arranged regions that are aligned with image gradients. Here, I want to investigate whether the superpixel grid provides a real existent benefit compared to the default uniform grid (named Grid in the experiments).

From a more general (or theoretical) point of view: The first part of this thesis evaluated superpixel segmentation algorithms and their properties. In particular, the benchmark of chapter 3 also included the regular BOX segmentation which is related to the fixedly arranged Grid of this chapter. Superpixels showed to be better aligned with object contours than the regular grid BOX segmentation in the quality benchmark in section 3.4. In terms of repeatability (or stability) in the presence of image changes, the performance showed to be dependent on the type of changes in section 3.5. For example, superpixel segmentations are much more stable in the presence of image shifts, rotation

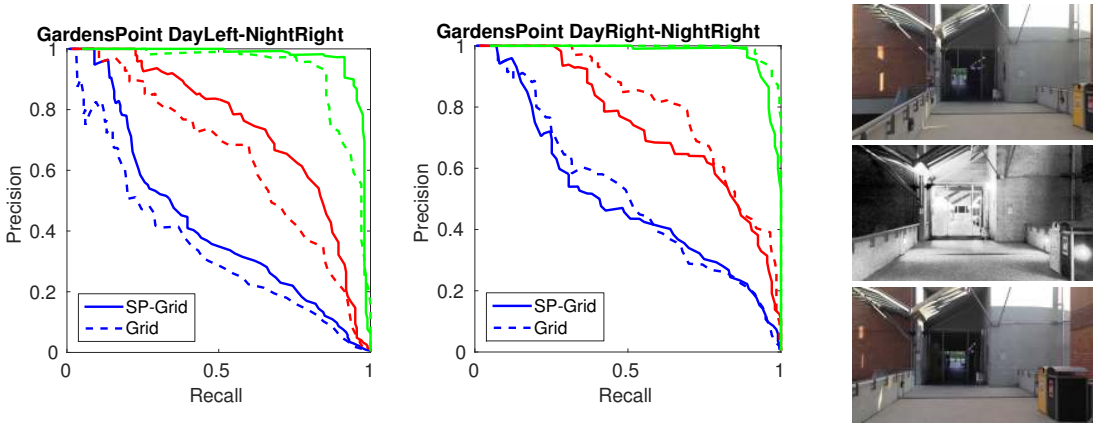


Fig. 7.18.: Comparison of Grid and SP-Grid on the Gardens Point datasets, Day*Left*-NightRight and Day*Right*-NightRight. The solid lines show the result of the proposed SP-Grid and the dashed lines the results of the fixed Grid. The different colours indicate how restrictively image matchings are accepted, the maximum image distance is: blue$\hat{=}$1, red$\hat{=}$3, and green$\hat{=}$10. Please see text for details.

or shearing while the regular grid BOX segmentation is for example unaffected by image noise. In terms of complex combinations of image changes, as they are measured by the proposed MUSE metric in combinations with datasets with ground truth optical flow, the performance of BOX depends on the amount of motion. For example for the KITTI images obtained from a driving car, BOX segmentations are more affected than for the Sintel data, with a more static camera and fewer moving objects. However, for the small motion between consecutive video frames, as they are used in KITTI, and small segment numbers, BOX can compete with some of the compact superpixel segmentation algorithms. So the question is, whether the theoretical advantage of a compact superpixel segmentation is large enough to result in a considerable benefit for place recognition.

I want to clearly point out that the intention of these experiments is not to evaluate the chosen grid arrangement - it is considered sufficient to know that the choice is reasonable (cf. section 7.4.2). To evaluate whether there is a benefit when using the SP-Grid instead of the fixed Grid, Fig. 7.18 shows the results of both algorithms on place recognition on the GardensPoint datasets presented in section 5.1.2. This dataset provides images from the same scene at day time and during night with an optional lateral shift. An example image triple showing the appearance change and the lateral shift can be seen on the right part of Fig. 7.18. The resulting curves for the DayLeft-NightRight comparison show a clear benefit from the usage of a SP-Grid.

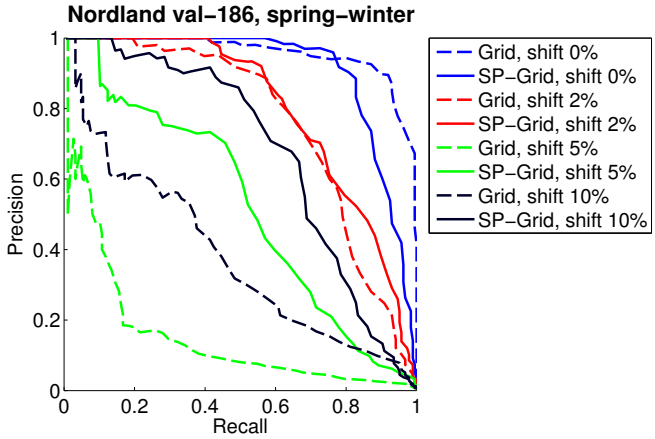


Fig. 7.19.: Comparison of the robustness of Grid and SP-Grid towards different amounts of horizontal shifts in corresponding images. For shifts $> 0\%$ SP-Grid performs superior to Grid in this place recognition experiment.

Marginal note 7.1 (Comparability of place recognition results I) *The different colours in Fig. 7.18 indicate the maximum distance of matched frames in the sequence that are accepted as showing the same place. GardensPoint and the other used datasets comprise image sequences, not images of disjunct places. An example sequence can e.g. be seen in Fig. 5.6. Typically this is handled by subsampling a set of disjunct places or by allowing matching images up to a certain distance in the original sequence - the effect of this distance can be seen by the differently coloured curves in Fig. 7.18. Such variations in the evaluation hamper the comparability of results from different papers - similar to the different results shown for the NC superpixel algorithm in Fig. 3.1.*

For this comparison of Grid and SP-Grid, the overall place recognition performance (of course) increases with less restrictive distances. However, SP-Grid always outperforms Grid in the presence of viewpoint changes. The right plot shows the results on the GardensPoint dataset without viewpoint change - there, the fixedly arranged Grid performs somehow better. To more precisely evaluate the influence of viewpoint changes, we can use the aligned Nordland images in combination with artificial viewpoint changes. Therefore, a second set of experiments is conducted using 186 uniformly sampled places from the Nordland Spring Winter validation dataset. The place recognition results for different amounts of horizontal shift can be seen in Fig. 7.19. It is obvious that the SP-Grid clearly outperforms the fixed Grid for all non zero amounts of shift. For the chosen (and presumably not optimal) choice of the initial grid arrangement described in section 7.4.2, a shift of 5 % of the image width seems to constitute a *critical* case. The regions of the fixedly arranged Grid show only small overlap for this shift, while for larger shifts, the overlap increases since regions now overlap with the regions corresponding to their neighbours in the grid. This is a somehow periodic behaviour with different frequency for each layer of the grid. The evaluation of the region overlap in Fig. 7.13 showed that the SP-Grid smooths the average overlap compared to the fixed Grid since

the superpixels adapt the initial grid to the image content. While this may decrease the performance for perfectly aligned images, place recognition based on the SP-Grid shows to be more robust against viewpoint changes - in particular for the critical shifts.

The following section 7.7.4 will include a comparison of the performance of SP-Grid with the other local region detectors. It will, in particular, refer to these critical shifts using data from the full Nordland journey.

Result 7.8 *The proposed SP-Grid showed to be less sensitive to viewpoint changes than the fixed Grid. In particular, critical shifts that badly correspond to the region distribution in the fixed Grid can be better handled by the adaptive regions of SP-Grid.*

7.7.3. Is there a benefit from superpixel based reweighting of scale space extrema?

In this section, I want to show results on the importance weighting of regions obtained from scale space extrema based on a segment soup. This approach was proposed in section 7.4.3. The evaluation of different scales of SIFT-DoG regions based on the proposed size-sensitive measure in section 7.5 showed that larger regions than SIFT-DoG-6, i.e. the regions with twice the radius SIFT-DoG-12, may provide higher detection rates of features with very high overlap (cf. Fig. 7.12). In particular in combination with powerful CNN descriptors and the assumption that many appearance changes occur at small scales (e.g. a tree may loose its leaves but its overall shape is supposed to be more stable) an evaluation of the potential benefit of the larger SIFT-DoG-12 regions is interesting.

Larger regions are supposed to be more sensitive to viewpoint changes and partial occlusions - exactly the issue that is in the focus of the proposed superpixel based reweighting of descriptor dimensions proposed in section 7.4.3. As a reminder, the idea is to assign high importance to those parts of the region, that are associated with superpixels that have high overlap with the region. In contrast, parts of the regions that are assigned only to superpixels that strongly overflow the region boundaries (e.g. 90 % of their area is outside the region) are considered to be more likely to belong to the less important background. Example region weights were illustrated in Fig. 7.9.

Fig. 7.20 shows results for the Alderley, GardensPoint and Nordland datasets. The comparison comprises the default SIFT-DoG-6, the regions located at the same positions but with twice the diameter SIFT-DoG-12, and the results of additional incorporation of the superpixel-based reweighting SP-SIFT-DoG-6 and SP-SIFT-DoG-12. On the original and shifted version of the Nordland Spring-Winter dataset, all SIFT-DoG variants perform similar. In particular, the simple artificial viewpoint change on the shifted datasets does not cause partial occlusions or other effects of real viewpoint changes that could hamper the enlarged regions. However, it can be seen, that the superpixel-based reweighting does not negatively affect the place recognition performance.

The evaluation of the repeatability of local region detectors in changing environments presented in section 7.6 revealed problems of SIFT-DoG in the presence of severe appearance changes. This can also be seen by the overall performance on place recognition

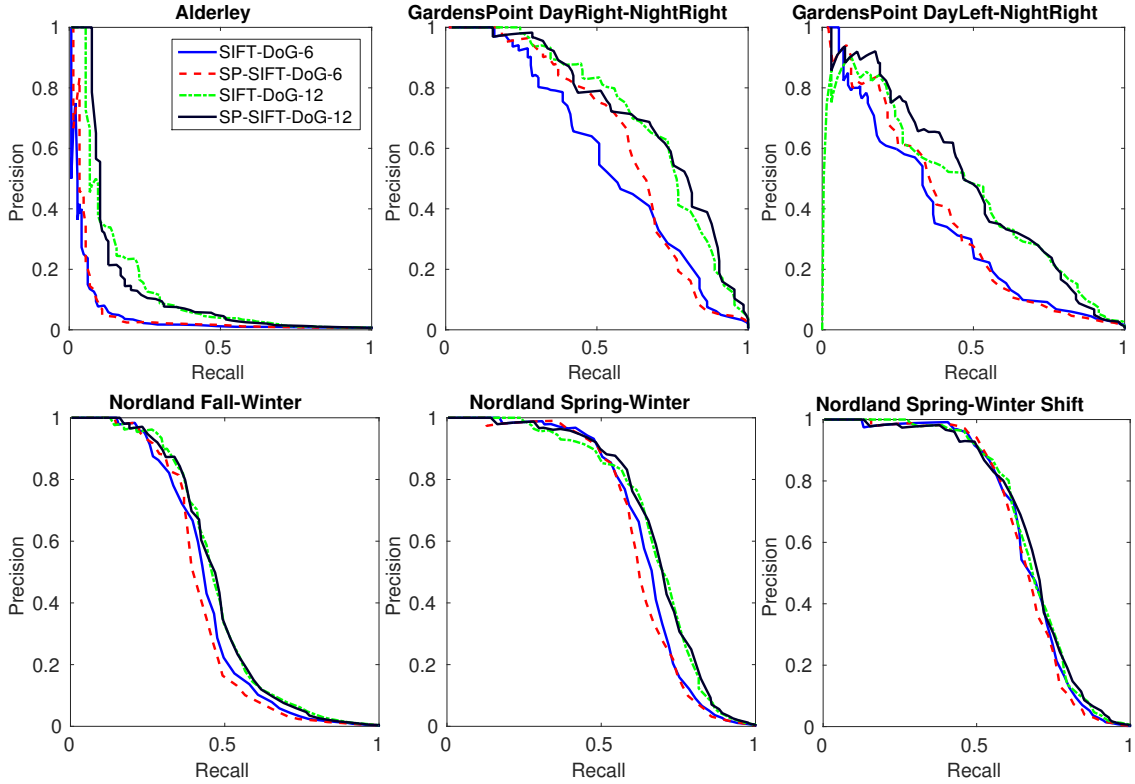


Fig. 7.20.: Results for different SIFT-DoG variants: Scale factors 6 and 12, with and without superpixel based importance weighting (SP-SIFT-DoG).

on the Alderley and GardensPoint day-night datasets in Fig. 7.20. The comparison of the performance to other local region detectors and in particular the other superpixel related approaches is topic of the following section 7.7.4.

However, in terms of the here compared SIFT-DoG variants and the severe appearance changes of the Alderley dataset (sunny day and rainy night), there is a clear benefit from using larger regions and a smaller additional benefit from the reweighting. The results on the GardensPoint dataset are similar. The presence of an additional viewpoint change (GardensPoint Day*Left*-NightRight) causes a performance decrease at high recall for the larger regions - this decrease can be compensated by the superpixel-based reweighting. However, the GardensPoint dataset is a quite small and later experiments will reveal that results on this dataset vary considerably (cf. Fig. 7.25). Thus, these results are considered preliminary and more work will be necessary to investigate the influence of the resizing and the superpixel-based reweighting on the robustness towards viewpoint changes. However, a promising observation is:

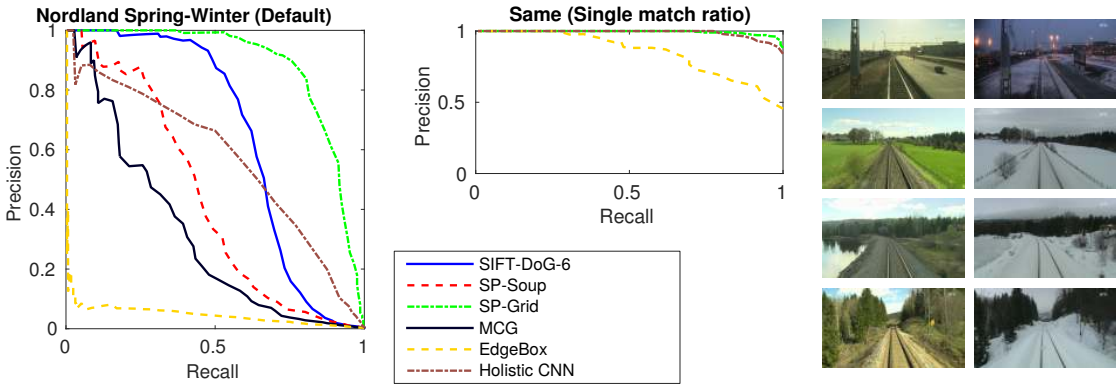


Fig. 7.21.: Results on Nordland Spring-Winter. To allow a comparison with results from the literature (in particular [Sünderhauf et al., 2015b]), the plot in the middle shows the *same results* with a different method to compute the precision recall curves, see note 7.2 for details. Nordland image credits: NRKbeta.no, cf. section 5.1.1.

Result 7.9 *The combination of the larger DoG regions with the superpixel-based reweighting (SP-SIFT-DoG-12) improves place recognition in the presence of severe appearance changes (e.g. the Alderley dataset) and showed to always perform at least on par with the default SIFT-DoG-6 in all other experiments. However, more evaluation on robustness to viewpoint changes is necessary.*

7.7.4. Comparison of local region detectors for place recognition in changing environments

So far, we know from the previous sections:

1. The proposed local region+CNN based landmarks and the novel Star-Hough image matching scheme perform superior to existing approaches (cf. section 7.7.1).
2. SIFT-DoG-6, SP-Grid, SP-Soup, EdgeBoxes and MCG are promising candidates to provide repetitive local regions for place recognition in changing environments (cf. local region detector comparison in section 7.6).

In this section, I want to compare these local region detectors in combination with the CNN based descriptor and the Star-Hough matching scheme by means of place recognition experiments. Although the insights obtained from visual inspection of individual image matchings are somehow limited, example landmark matchings based on the different regions detectors can be seen in Fig. 7.23, 7.27 and 7.28.

Fig. 7.21 shows the resulting curves for comparison on the aligned Nordland Spring-Winter test dataset (the same setup as in section 7.7.1). The right part of the figure shows example images from this sequence. The curves in the left plot show a large benefit from using the proposed SP-Grid regions compared to the established SIFT-DoG-6. These two approaches perform superior to the compared methods, in particular EdgeBoxes reveals problems.

Marginal note 7.2 (Comparability of place recognition results II) *The left plot in Fig. 7.21 shows precision-recall curves of a place recognition experiment. It can be seen that the image similarities provided by EdgeBoxes (the yellow curve) should not be used on this dataset. The plot in the middle of this figure shows another precision recall curve using the* ***same*** *image similarities - however, this curve indicates a reasonable performance. This second curve is generated with the evaluation method used in [Sünderhauf et al., 2015b] and similar to the there presented results of EdgeBoxes on this dataset. The difference between the two methods is not in the computation of image similarities - but how true and false matchings obtained from the image similarities are counted.*

In the left plot, the procedure described in section 5.3 is used: Given the similarities between all possible image matchings, they are divided into matchings and non-matchings based on a threshold t. All matchings that correspond to a ground truth matching are counted as true positives, all matchings that do not correspond to a ground truth matching are false positives, false negatives are all image pairings from the ground truth that are not in the set of matchings. From these three values, a point on the precision-recall curve is computed, the threshold t is varied to obtain a curve.

In contrast, Sünderhauf et al. [2015b] allow only a single matching for each place and evaluate the ratio of the similarity between the best and the second best matching for each place. While none of the two evaluation methods is in general better than the other, they measure different properties and result in significantly different curves. The here used evaluation method penalises algorithms that match single places to multiple false places. This can not happen in the evaluation of [Sünderhauf et al., 2015b] since they allow only matchings to a single previously seen place - however, in practical scenarios multiple revisits of the same place are realistic. For comparison, I also included the Holistic CNN and the proposed SP-Grid in the single-matching plot. Both clearly outperform EdgeBoxes in this measure as well.

However, this example confirms the implication of Fig. 3.1 in chapter 3: it is not possible to use a published curve directly for comparison to the own results.

Fig. 7.22 demonstrates the influence of an artificial viewpoint change by shifting the images 5 % of the image width - this showed to be the most critical case for SP-Grid in the previous experiments shown in Fig. 7.19. The performance of SP-Grid clearly drops. However, it shows to be significantly more stable than the holistic approach even in this particular challenging configuration for SP-Grid. The performance of the other region detectors drops slightly while keeping the order from the aligned setup. SIFT-DoG-6 is almost not affected by this shift and performs superior to SP-Grid in this setup.

So far, all place recognition experiments have been done on the Nordland spring-winter dataset. The evaluation of the sensitivity of local region detectors to appearance changes in section 7.6 showed that there are more challenging scenarios, e.g. day-night cycles. From the possible seasonal combinations of the Nordland sequences, fall-winter showed to be the most challenging and summer-fall the easiest in our previous evaluation in [Sünderhauf et al., 2013]. Fig. 7.24 shows the results on these two seasonal combinations using the original, aligned images. In accordance to the previous results [Sünderhauf

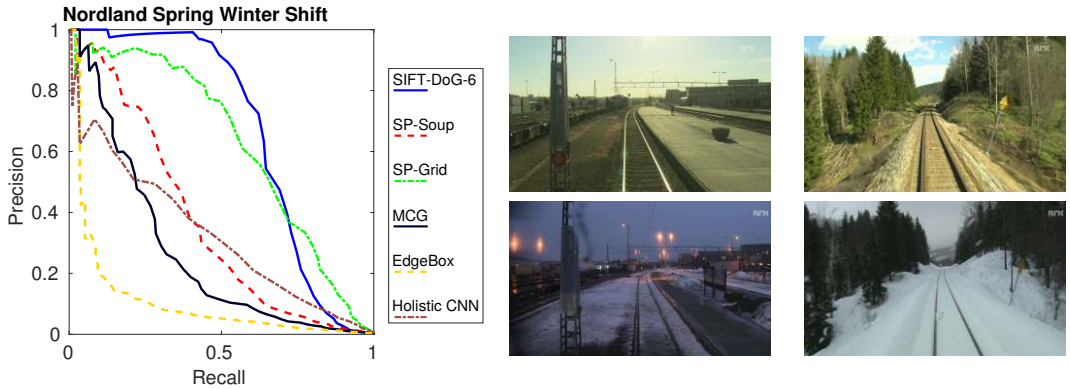


Fig. 7.22.: Results on the Nordland Spring-Winter dataset with artificial viewpoint changes. The images on the right show this rather small shift of 5 % of the image width. However, this showed to be the most challenging case for the SP-Grid in the previous experiments and is sufficient to significantly decrease the performance of the holistic approach. Nordland image credits: NRKbeta.no, cf. section 5.1.1

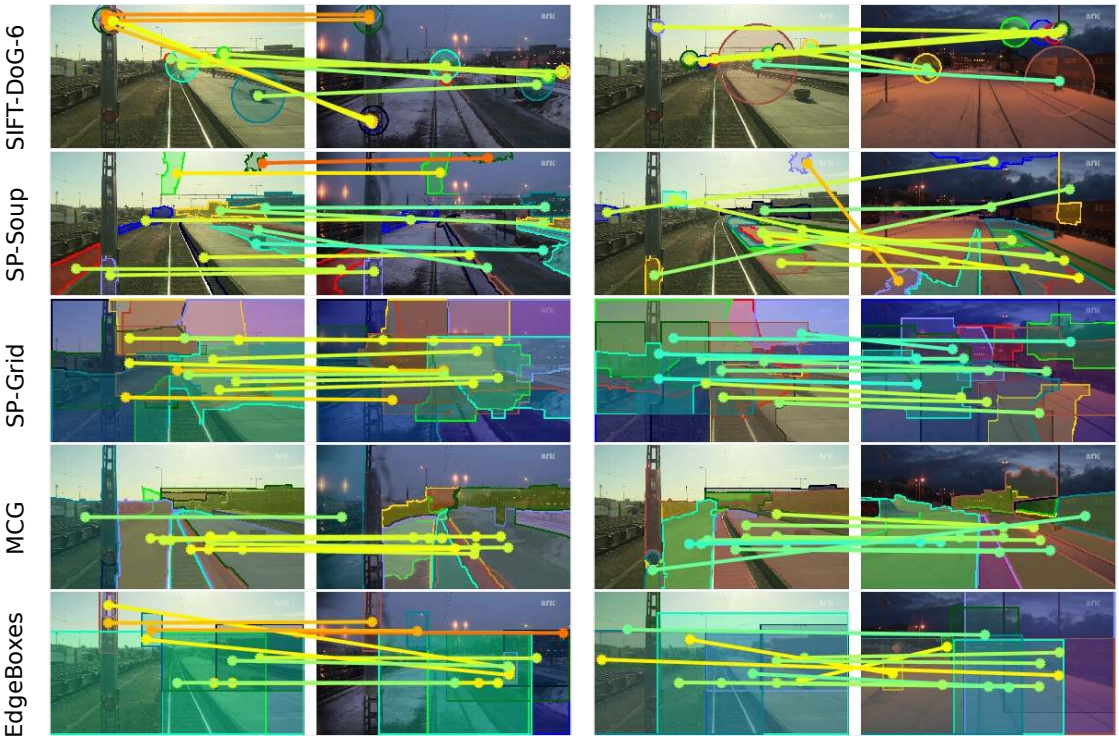


Fig. 7.23.: Example landmark matchings on Nordland Spring-Winter. The left image pairs show the (up to) ten most similar left-right matchings for two images of the *same* place, the right images show two *different* places.

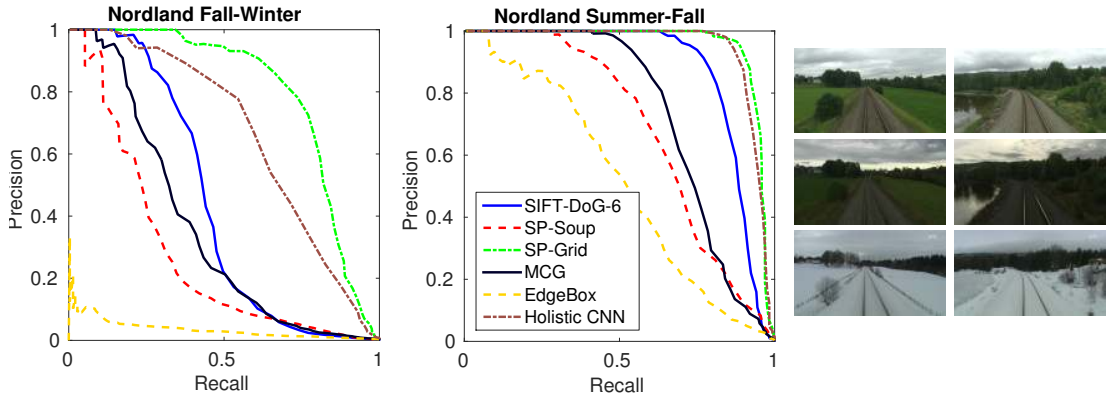


Fig. 7.24.: Results on the Nordland fall-winter and summer-fall combinations. The images on the right show two example places at different seasons. From top to bottom: summer, fall, winter. Nordland image credits: NRKbeta.no, cf. section 5.1.1

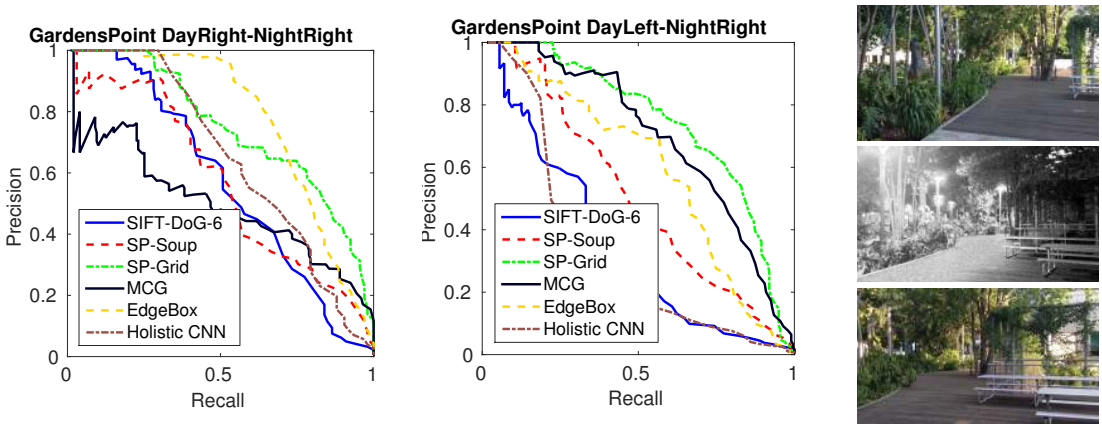


Fig. 7.25.: Results on GardensPoint day-night datasets. GardensPoint image credits: Arren Glover, cf. section 5.1.2

et al., 2013] and the results of SeqSLAM and SP-ACP presented in section 6.5, summer-fall is the much easier case than fall-winter. However, it can be seen that SP-Grid not only shows the best performance but the benefit for using the SP-Grid further increases for the more challenging dataset.

The next level of changing environments are day-night cycles. The left side of Fig. 7.25 shows the place recognition results on the GardensPoint DayRight-NightRight dataset (with traversals on the same side of the path, image matchings are allowed up to distance three in the sequence). For this type of appearance change, the SIFT-DoG-6 revealed problems in the region repeatability evaluation in section 7.7.1. In accordance with this observation, the severe appearance changes between the day and night images of the GardensPoint dataset results in inferior performance of SIFT-DoG-6. The SP-Grid shows reasonable performance but is clearly outperformed by EdgeBoxes.

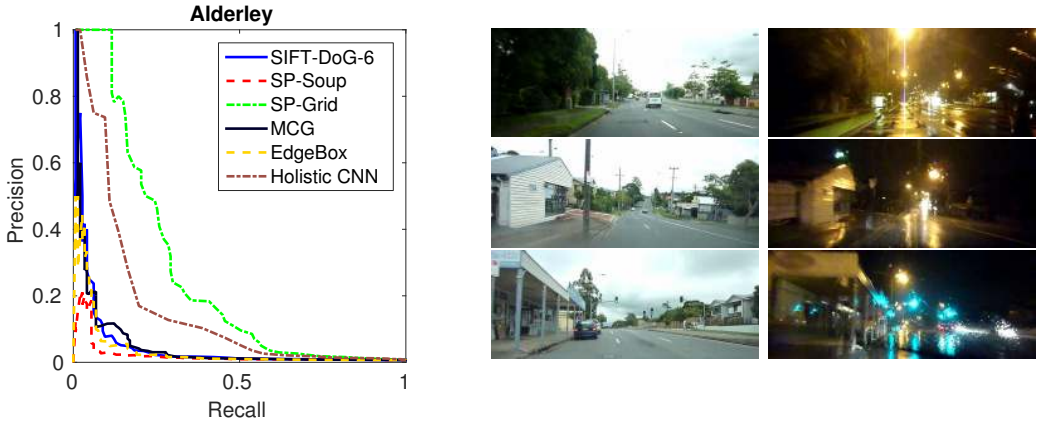


Fig. 7.26.: Results on Alderley - the most challenging place recognition experiment. Alderley image credits: [Milford and Wyeth, 2012], cf. section 5.1.3

However, the same experiment with an additional viewpoint change induced by traversing the path on the other side, reveals a somehow surprising performance drop of EdgeBoxes. This can be seen in the middle part of Fig. 7.25 showing results of the GardensPoint DayLeft-NightRight dataset. In contrast, the performance of the second object proposal algorithm MCG considerably increases. The object proposal algorithms are intended to find regions that are likely to contain an object in an image, they are not designed to find repetitive landmarks. The example image triple shown on the right side of Fig. 7.25 shows that due to the viewpoint change, salient objects like the bench and the table on the very right can disappear - however, this can not explain why the performance of MCG improves. Presumably, further visual inspection of individual matchings, as they are shown in Fig. 7.27, could help to find an explanation. However, the route of the GardensPoint dataset is rather small compared to the other datasets, which makes it more sensitive to individual effects on few images.

A significantly larger track is provided by the Alderley dataset presented in section 5.1.3. This dataset comprises two drives through a suburb, one at a sunny day and the other during a rainy night with low visibility. Example images of these severe appearance changes can be seen in the right part of Fig. 7.26, example landmark matchings can be seen in Fig. 7.28. The resulting curves shown in Fig. 7.26 reveal two insights: (1) This is by far the most challenging dataset in this evaluation. (2) The benefit from using the SP-Grid regions increases. For these low visibility conditions, the SP-Grid regions converge to a regular grid in the absence of considerable image gradients due to the compactness constraints of the used superpixel algorithms - this constitutes a reasonable default set of regions. In particular, this is in contrast to the SP-Soup that adapts to even very small image gradients due to the absence of compactness constraints.

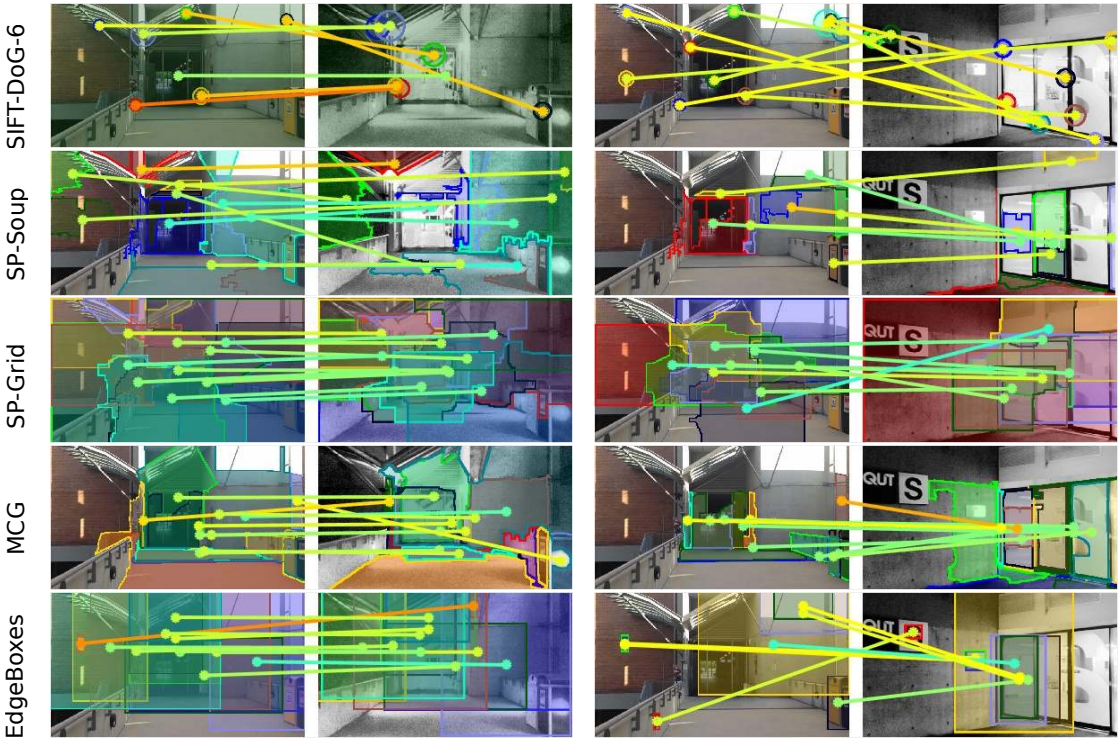


Fig. 7.27.: Example landmark matchings on the GardensPoint DayLeft-NightRight dataset. Again, the left image pairs show the (up to) ten most similar left-right matchings for two images of the *same* place, the right images show two *different* places.

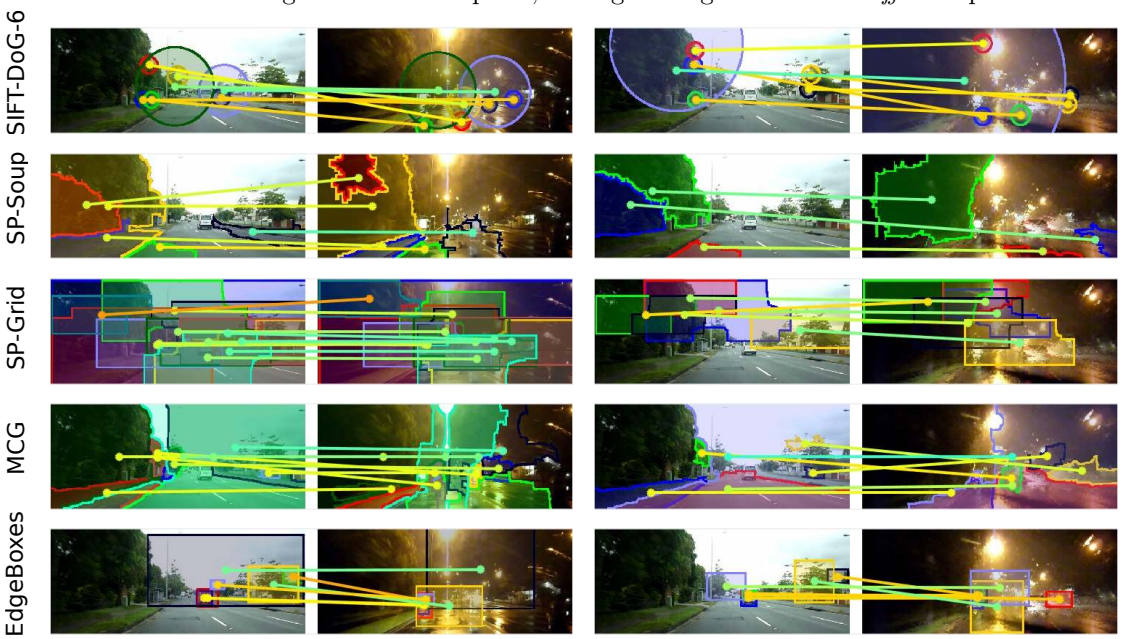


Fig. 7.28.: Example landmark matchings on the Alderley dataset. Again, the left image pairs show the (up to) ten most similar left-right matchings for two images of the *same* place, the right images show two *different* places.

Result 7.10

1. *For low amounts of appearance change but considerable viewpoint changes, SIFT-DoG-6 is a reasonable region detector for combination with a CNN based descriptor.*
2. *The performance of the object proposal algorithms and the SP-Soup vary considerably dependent on the dataset.*
3. *SP-Grid is the best choice for well aligned images. In presence of viewpoint changes it showed to be more stable than the fixedly arranged Grid but more sensitive than e.g. SIFT-DoG-6.*
4. *For severe appearance changes, as they occur e.g. in Nordland Fall-Winter or in the Alderley dataset, the proposed SP-Grid is the best choice.*

7.8. Summary of achieved results and directions for future work

Topic of this chapter was place recognition in changing environments based on *local* image features. Before the concluding chapter 8 of this thesis will discuss the achieved results of this chapter in the overall context of place recognition in changing environments and in particular their relation to the *holistic* SP-ACP approach of the previous chapter 6, I want to summarize the results obtained in this chapter.

This chapter presented the novel ideas of CNN based landmarks and the Star-Hough image matching scheme. It further described a set of novel approaches to incorporate superpixel segmentations in landmark detection: using *stable* segmentations directly in form of a segment soup (SP-Soup), using *compact* segmentations to create a set of adaptive and overlapping regions from an initial grid tessellation (SP-Grid), an importance weighting based on a superpixel soup (SP-SIFT-DoG) and the usage of object proposal methods.

The combination of local region detectors + CNN based descriptors + Star-Hough showed promising place recognition performance. In particular, it enables to use established region detectors like SIFT-DoG-6 for a wide range of appearance changes where they could not be used before (cf. the black curves in Fig. 7.17).

From the set of superpixel related approaches, the novel superpixel grid approach SP-Grid presented in section 7.4.2 provided a significant improvement in many experiments, in particular for severe appearance changes. The key idea was to adapt a regular grid with a superpixel segmentation to increase robustness against viewpoint changes. This benefit, compared to a fixed grid, has been shown in section 7.7.2.

This approach has some similarities with the patch verification step proposed by Milford et al. [2013]. However, they use a holistic approach to select candidates and subsequently validate the candidates based on comparison of rectangular patches. Although the patches are tested for small position deviations, the holistic candidate selection step is supposed to be sensitive for viewpoint changes. However, this method is one of the very few for which reasonable results were presented for single image similarity on the Alderley dataset. Unfortunately, to the best of my knowledge, there is no implementation available and, as has e.g. been shown in Fig. 7.22, the plain curves from different

experimental setups cannot be compared.

In the here presented experiments including critical viewpoint changes, i.e. the shifted Nordland dataset, local regions provided by SIFT-DoG-6 perform superior to the SP-Grid - this is supposed to happen for other datasets with severe viewpoint changes and only moderate appearance changes as well. However, the chosen initial grid arrangement for the superpixel grid is reasonable, but no special efforts have been made on optimization - in particular not for the test datasets. For example, if we expect considerable lateral shift but only small longitudinal offset, a higher horizontal than vertical resolution of the grid and accordingly higher region overlap may be preferable (while retaining the total number of landmarks). Also, the number and distance of the different scales in the grid may be improved. If additional rotations in the image plane are expected, an orientation could be assigned to each grid region similar to the method used for SIFT descriptors.

Regarding the usage of SIFT-DoG regions for place recognition in changing environments, the presented experiments indicate a benefit in case of larger scaling factors than are typically used (i.e. by the originator of SIFT [Lowe, 2004]). This result was obtained based on the novel size-sensitive measure for the repeatability of local region detectors presented in section 7.5 and validated on place recognition experiments in section 7.7.3. In particular, the combination with the proposed superpixel-based reweighting of section 7.4.3 showed promising results. However, the results in combination with viewpoint changes are somehow preliminary and should be subject to future research.

The other superpixel related approaches showed reasonable performance, however, with considerable variations. None of these approaches, neither the segment soup nor the object proposal algorithms, was designed for creation of repetitive landmarks. However, in particular the available object proposal algorithms may be a promising foundation for "landmark proposal" algorithms (a term coined by Niko Sünderhauf). Regarding the superpixel segment soup, a saliency measure could be used to select distinctive regions from the soup.

A lot of parameters of the system were kept fixed to reduce the complexity of the evaluation, e.g. the number of landmarks. With respect to the remarks on the computational efforts of CNN features in section 7.2, also higher number of landmarks can be used and may further improve the performance. For all presented approaches to incorporating superpixel segmentations in local region detection, reasonable but presumably not optimal parameter settings were used. A more detailed investigation of the approaches and their parameters is expected to yield further improvements. This also holds for the proposed usage of CNN descriptors and the novel Star-Hough image matching scheme. Regarding the fact that the used CNN has been trained for a completely different task (image classification), a considerable improvement can be anticipated from a network particularly trained for recognizing landmarks across severe appearance changes.

So far, the place recognitions are obtained from single image comparisons. In particular for severe appearance changes (e.g. the sunny day and rainy night on Alderely), exploiting even short *sequences* of images may considerably reduce the number of false positive matchings. The similarities obtained from the proposed system can directly by

fed into a sequence matching approach, e.g. the SeqSLAM back-end.

Moreover, SP-Grid showed the best performance in the presence of severe appearance changes, SIFT-DoG-6 showed the best robustness to viewpoint changes - there is no reason that prevents the usage of a combination of both types of landmarks in the proposed system. However, the best approach how to combine these different landmarks is a interesting direction for future work.

And it's not just a list of hog wash
That we've done in the past
It's a chronicling of our rise to power

Tenacious D, from the song "History", 2006

8 Conclusions

This final chapter summarizes the achieved insights and discusses open questions of both parts of this thesis. A special focus will be on practical considerations on place recognition in changing environments. In particular, I want to compare the achievements obtained from the two novel place recognition approaches presented in chapters 6 and 7 - to show the relation between both and to summarize the improvements towards existing methods. This will demonstrate how close we are to a solution for some types of environmental changes and what are open problems.

8.1. What has been achieved - Summary of the main contributions

At first glance, the topics of the two parts of this thesis, superpixel segmentations and place recognition in changing environments, appear quite unrelated. However, the novel ideas presented in the second part revealed a benefit from using the tool *superpixel segmentation* for a solution to the problem *place recognition in changing environments*. From a rather abstract point of view, the nature of contributions to the two related fields also show some commonalities - in both parts there are contributions on four levels:

1. Algorithmic approaches to solve a particular problem

In the first part of this thesis, chapter 4 presented two novel superpixel segmentation algorithms: Preemptive SLIC and Compact Watershed. Preemptive SLIC uses a local preemption criterion and code optimization to speed up the computations and save about 80 % of the runtime compared to the original SLIC algorithm (a speed up of factor 5). The second algorithm, Compact Watershed uses compactness constraints to turn

seeded Watershed segmentation from an oversegmentation algorithm into a compact superpixel approach while remaining the fast runtime - the proposed novel algorithm showed to be the fastest compact superpixel algorithm available ($\sim$ 100 Hz on images of size 481 $\times$ 321) and can, for example, be used in the SP-Grid of chapter 7.

In the second part, chapter 6 presented the novel concept of appearance change prediction for place recognition in changing environments and an algorithmic implementation based on superpixels, SP-ACP. The image is split into superpixels which can be associated to words of a learned vocabulary. Using training data of image pairs showing the same scenes at different environmental conditions and pixel level associations between the images, a dictionary can be learned. The dictionary is used to translate words between superpixel vocabularies. From these translations, a predicted image can be generated. The incorporation of this additional prediction step showed to improve the place recognition performance in changing environments in combination with existing holistic place recognition approaches (BRIEF-Gist and SeqSLAM).

In terms of using local features for this particular place recognition task, chapter 7 introduced the novel combination of local region detectors with powerful CNN based descriptors. Moreover, the proposed Star-Hough image matching scheme can compute image similarities based on landmarks with low spatial position precision and high number of outlier matchings. It showed to further significantly improve place recognition performance. This novel place recognition system (CNN and Star-Hough) can be used with established region detectors, e.g. the detector used in SIFT keypoints. This combination shows reasonable performance for moderate appearance changes and provides the highest robustness to viewpoint changes. Finally, a set of novel, superpixel based local region detectors has been proposed - most notable a multiscale superpixel grid, SP-Grid; but also a segment soup, superpixel based importance weighting and the usage of object proposals for landmark detection. Particularly in the presence of severe appearance changes, using SP-Grid in the novel place recognition system, outperforms all other compared methods.

2. Novel metrics to evaluate properties of algorithms

Besides novel algorithms, this thesis also worked on the evaluation of the properties of these algorithms and how these properties can be measured. The first part of this thesis showed the need for a standardized benchmark for superpixel segmentation algorithms. To overcome this issue, in a first step, a set of important properties of these algorithms has been identified: segmentation quality, runtime, stability and compactness. In particular for detailed analysis of the latter two properties, there were no evaluation methodologies available. Therefore, chapter 3 presented novel metrics to measure these properties, most notably the motion undersegmentation error MUSE and the motion discontinuity error MDE. These measures exploit datasets with known ground truth optical flow to evaluate stability and compactness of superpixels. Other proposed measures for the segmentation stability were metrics for the robustness towards affine image transformations and image noise. In terms of compactness, it is further proposed to use the weighted isodiametric quotient and the standard deviation of the superpixels' size.

In terms of metrics, the second part of this thesis used the established evaluation of place recognition experiments based on precision-recall curves. However, chapter 6 presented novel evaluation methodologies to obtain insights on the properties of superpixel descriptors and vocabularies for their usage in the proposed SP-ACP approach. In particular, section 6.3 proposed a method to evaluate the amount of semantic meaning provided by a superpixel word based on conditional entropy. This was used to justify the assumptions made in the proposed appearance change prediction approach. In terms of local image features, the novel size-sensitive region overlap measure, derived from the established method by Mikolajczyk et al. [2005] and presented in section 7.5, revealed novel insights on the influence of the scale factor in local region detectors - i.e. it indicated a potential benefit from using larger scale factors in SIFT-DoG for changing environments.

3. Experimental comparison

According to Richard P. Feynman, "[t]he principle of science, the definition, almost, is the following: The test of all knowledge is experiment. Experiment is the sole judge of scientific truth." [Feynman et al., 1963, p. 1] Both parts of this thesis provided extensive experimental evaluation. The superpixel benchmark in the first part of this thesis compared the segmentation quality, stability, compactness and runtime of 14 implementations of existing and novel superpixel algorithms - using existing and novel metrics. This is intended to be a useful source of information for users to select a superpixel algorithm for the task at hand and for researches to compare new algorithms to the state of the art.

In terms of place recognition in changing environments, of course, a lot of place recognition experiments using novel and existing methods have been conducted. Moreover, chapter 6 also included experimental comparisons of superpixel descriptors and vocabularies. Further, chapter 7 provided an experimental evaluations of the robustness of local region detectors towards appearance changes, independently of a particular descriptor or place recognition experiment. For place recognition in changing environments, only a few ready to use solutions are available. To the best of my knowledge these are SeqSLAM, FAB-MAP and holistic CNN descriptors. The results sections of the chapters 6 and 7 provided extensive experimental evaluation of the proposed novel algorithms and included a comparison to these available methods. The section 8.3 will summarize the achievements by bringing the results of the two novel approaches and the existing methods in a joint picture.

4. Sustainability - Open source implementations

An additional important insight is the fact that results from different papers often cannot be compared directly - even if they seem to use the same dataset and metric. This became obvious in both parts of the thesis:

- The curves from different papers showing the performance of the NC superpixel

segmentation algorithm, using the BSDS dataset and the boundary recall metric in Fig. 3.1.

- The evaluation of EdgeBoxes landmarks on the Nordland dataset based on precision recall curves in Fig. 7.21.

One way to approach this problem is to provide *all* the necessary details to reproduce the same results - for example in form of an available implementation of the own work. In particular, I provide open source Matlab toolboxes for the superpixel benchmarks. They include implementations of the used metrics and additional functions to run the benchmarks on new algorithms. Newly presented algorithms can be compared to the existing superpixel methods using these implementations - download the toolbox and the datasets, configure a YAML file to incorporate the new algorithm and wait for the resulting curves to appear amongst those of the existing algorithms.

Further, I provide an open source implementation of the novel place recognition system including the CNN based descriptors, the Star-Hough image matching and the most promising local region detectors. This is intended to serve as novel indicator of the state-of-the-art performance on place recognition in changing environments and as baseline method for comparison of novel approaches. Moreover, higher level routines that require a module for computation of image similarities in the presence of severe appearance changes can integrate this implementation. Details on the availability of the implementations can be found in section 1.3.

8.2. What are open problems in terms of superpixel segmentations?

The standardized superpixel benchmark, presented in the first part of this thesis, addressed the problem of contradicting results on superpixel comparison in the literature. It was used to compare available implementations of algorithms. Although the presented comparison included various superpixel algorithm implementation, the comparison is not exhaustive. Most importantly, this is due to the limitation to the comparison of *fast* algorithms (and NC due to its historical relevance) with an *available implementation*. However, there are other algorithms that might be interesting candidates for a further extension of this comparison - they can be integrated easily using the available implementation of the benchmark.

The experimental comparison conducted in chapter 3 was used to identify two directions for improvements: a faster version of SLIC and compact segments from the very fast Watershed - novel algorithms for both were proposed in chapter 4. In terms of the development of further novel superpixel algorithms, an interesting observation is that most existing algorithms are optimized for segmentation quality and/or runtime. Although new algorithms that achieve even higher boundary recall at lower runtime will still be valuable, in my opinion, novel algorithms should better address the more diverse requirements of the particular applications, e.g. segmentation stability and compactness.

Pushing superpixel algorithms towards these properties is supposed to impair other

properties like segmentation quality - a question of finding the best trade-off. Chapter 3 showed a trade-off between segmentation stability and compactness for the available algorithms. Stable algorithms like FH are less compact and segmentations from compact algorithms like SLIC are less stable. The application of superpixels for landmark detection in chapter 7 revealed a somehow surprising benefit of using *compact* instead of *stable* segmentation algorithms. This is an example of an application that is supposed to benefit from new algorithms with a better adjusted trade-off to be considered compact *and* stable - a possible direction for future work. For example: What about a superpixel algorithm particularly designed for creation of a SP-Grid, e.g. a compact-FH?

From a rather abstract point of view, I see three general issues of superpixel algorithms:

1. They still make hard decisions about possible object boundaries.
2. Superpixels are created purely bottom-up.
3. The algorithms are almost always purely hand engineered.

As stated in the introduction to segmentation in section 2.1, segmenting the single foreground object from an image is an ill-posed problem. This is in particular due to the fact that the correct solution depends on higher-level knowledge about the task. Superpixel segmentations deal with this problem by providing an intermediate level image representation that can be used to create many different segmentations - however, these are not *all* possible segmentations that could be generated from the input image. The number and size of the superpixels somehow define a minimum scale at which details of the image are represented and at which objects can be distinguished in later processing steps. This is both, benefit and drawback of superpixels - it reduces the complexity for subsequent processing steps, but makes hard decisions prematurely. Dealing with this trade-off is a major challenge, in particular for those who just want to use superpixels for a certain task and have to chose an algorithm and its parameters.

For this and other reasons, a better interaction between the superpixel segmentation process and the higher level task seems valuable. Superpixels are typically created purely bottom up. The incorporation of top-down information in the segmentation process may help to create better superpixels for the particular task. Object proposal algorithms are an example for a processing step that uses superpixels, but may also feed useful information back to the superpixel creation process, e.g. about the expected size or shape of the expected objects. However, this requires novel superpixel algorithms that can incorporate such high level information. Currently, almost all superpixel segmentation algorithms are purely hand engineered. In particular the incorporation of such top down information could be approached by learning mechanisms - e.g. similar to the very recently presented end-to-end learning of semantic segmentation using *fully* convolutional neural networks [Long et al., 2015].

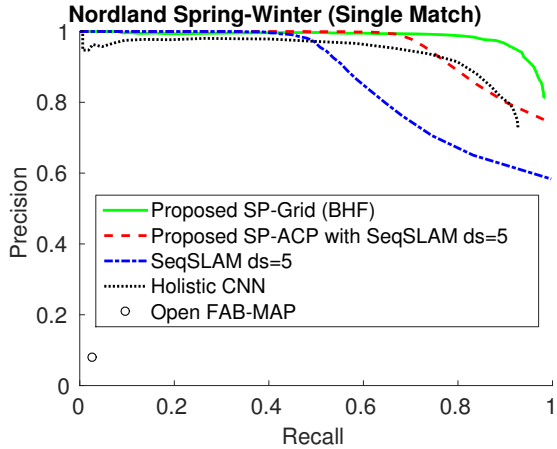


Fig. 8.1.: This figure summarizes the achievements with respect to place recognition in changing environments. The two proposed approaches perform superior to the state-of-the-art methods. Moreover, in contrast to the proposed landmark based approach, the holistic CNN and SeqSLAM are known to **fail** in the additional presence of viewpoint changes. This was, for example, shown in Fig. 7.22 and Fig. 7.25 - there, the proposed local landmark based approach showed to be much more robust to viewpoint changes.

8.3. Place recognition in changing environments - Where are we at the question where are we?

Finally, I want to compare the tow novel approaches to place recognition in changing environments, discuss open questions and directions for future work.

8.3.1. How do the two novel approaches compete?

The chapters 6 and 7 presented two novel approaches to the same problem. An obvious question is: How do the two approaches compete to each other and to the state-of-the-art methods? Fig. 8.1 shows the result of this comparison. From chapter 6 the combination of the proposed appearance change prediction system SP-ACP with SeqSLAM and from chapter 7 the SP-Grid showed the best performances. For the comparison of these two approaches, one has to keep in mind that SeqSLAM matches *sequences* of images - its performance on the Nordland dataset improves with increasing sequence length. For this comparison a reasonable sequence length of ds=5 is used. Further, the SeqSLAM implementation already selects a *single* best matching for each image. To allow comparable results, all approaches have been run on the full Nordland Spring-Winter test dataset and the same *single* image matching scheme has been applied to all methods - hence, the curve for the SP-Grid shows better precision-recall than in chapter 7 (for details, please refer to the description of precision-recall curves in section 5.3). For speedup, the CNN descriptor used for SP-Grid is approximated based on binary hash functions as described in section 7.2. This will be further discussed in section 8.3.2.

The results presented in Fig. 8.1 show that FAB-MAP cannot handle the appearance

changes induced by the different seasons. SeqSLAM and the holistic CNN are available methods that provide reasonable performance on this datasets - however, both are holistic approaches and known to fail in the presence of additional viewpoint changes (cf. Fig. 7.22 and 7.25). In accordance to the results presented in section 6.5, the additional appearance change prediction step using the proposed SP-ACP implementation improves the performance of SeqSLAM. The novel landmark based approach using the SP-Grid regions of chapter 7 performs superior to all other methods. Moreover, this local landmark based approach showed to be robust against viewpoint changes.

Regarding the practical applicability, the two presented approaches are quite different. The proposed idea of appearance change prediction of chapter 6 is a rather abstract or theoretical approach. Although the presented superpixel based implementation showed to improve the performance of existing place recognition approaches, the strong requirements on the training data and the unclear generalization capabilities prevent a direct practical application. Moreover, the artefacts introduced during creation of the predicted images prevent the combination with local region detectors and can even decrease place recognition performance in case of non changing environments. The final section of chapter 6 discussed possible approaches to overcome these limitations.

In contrast, the novel landmark based system presented in chapter 7 is a ready-to-use approach. It showed very promising results on various datasets, showed to be robust to viewpoint changes and provided reasonable results even in case of severe viewpoint changes. It outperformed the available methods and thus improved the state of the art in this field. Moreover, the final section of chapter 7 presented several promising directions for further improvement based on the proposed ideas.

8.3.2. Is the problem of place recognition in changing environments solved?

No - however, in terms of computing image similarities, we got really far. The proposed landmark based system showed very promising place recognition performance in the experiments. Moreover, chapter 7 provided directions for future work including particularly trained CNN descriptors, exploiting image sequences and the combination of contrasting local region detectors. This may further improve the presented precision recall curves.

However, this is just the basis for a practical solution to the problem of place recognition in changing environments. One has to keep in mind that place recognition is more than the computation of image similarities. And long term navigation is more than recognizing places despite severe appearance changes.

From image similarities to place recognition

Fig. 8.2 shows examples for places that could be successfully matched despite severe appearance changes and also failure cases. On the Nordland dataset, the missed matchings in the third row also provides a high similarity - however the images of the false matching of the second row are *even more* similar. For the really severe appearance changes of the Alderley dataset, the overall image similarities are smaller. There, it is not surprising

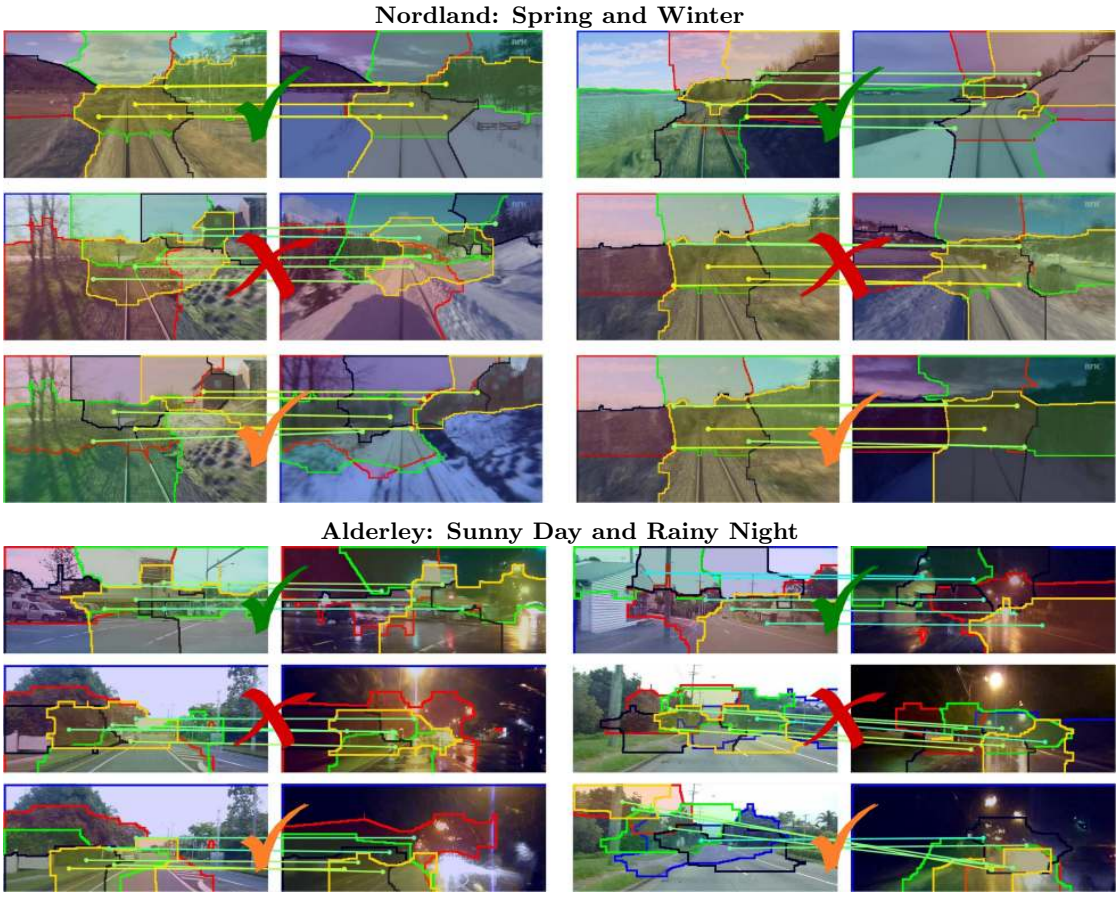


Fig. 8.2.: If we would define a place matching as the *single most similar image* from the database, these would be examples of good and bad place recognitions. For both datasets, the top row shows true positive matchings and the second row false positive matchings. The third row illustrates the true matching that should have been found instead of the false matching in the second row.

to miss image matchings. However, to create a high similarity between images that are not true matchings, individual false matchings of landmarks *and* a corresponding spatial arrangement of these landmarks have to happen in combinations. If this higher image similarity results in a false place recognition, depends on the subsequent processing steps.

The evaluation based on precision-recall curves enables a statistically profound comparison of the performance of place recognition algorithms. However, one has to keep in mind what exactly is measured. This was demonstrated in Fig. 7.21, showing two different precision-recall measures (cf. note 7.2). Moreover, both measures show a precision-recall *curve*. To obtain the curve, a threshold is varied. This threshold decides whether two images show the same place or not based on their similarity. In practice, it may be hard to actually select this threshold - or in other words: to *decide* whether two images show the same place or not. Presumably, a simple, fixed threshold will not be enough in changing environments.

For instance, think of the image similarity of two different Nordland places in summer, and the similarity of two images of the same place, one from summer and the other from winter. Recognizing a place based on a simple threshold poses a lower bound on the image similarities - all images with a higher similarity are considered to show the same place. However, the two summer images of different places might be much more similar than the images of the same place taken in summer and winter. Therefore, the image similarity should be evaluated in higher computational levels together with higher level knowledge. In particular, information about the expected appearance changes and the overall visual ambiguity in the database of priorly seen places (or at least the available part). This is related to one of the presumably most challenging questions for place recognition in changing environments: Is this a *new* place or an priorly seen place with a changed appearance? Related work in these directions can, for example, be found in [Cummins and Newman, 2011] and [Churchill and Newman, 2012]. However, there is plenty of space for future work.

From place recognition to long term localization

A motivation for place recognition in changing environments is long term localization and this can be expected to imply *large* datasets. With respect to practical applications, a low runtime always sounds valuable. However, in contrast to, for example, tracking of a keypoint across an image sequence, relocalization based on place recognition can be expected to happen at a much lower frequency. To keep the computational efforts for the overall place recognition chain feasible, the required time for an image *comparison* is typically more important than that of the computation of the image features. For example, given two image sequences, each with n images, the number of image comparisons is n^2 for an exhaustive comparison of all images.

This can be approached on two levels: We can speed up the individual comparison or we can reduce the number of comparisons. Section 7.2 discussed the feasibility of using CNN-based descriptors for practical application by presenting an approach from the literature to speed up the comparisons based on binary hash functions. The resulting slight decrease in place recognition performance can be seen in the right part of Fig. A.13 in the Appendix. This procedure has also been used for the landmark descriptors in the final comparison in Fig. 8.1. However, in preliminary experiments, these binary hash functions showed to best approximate the distance of very distant CNN features. For points located closer in the high dimensional space, the quality of the approximation decreases. Since the binary hash functions are chosen randomly, an interesting question might be whether they can be better selected to improve the approximation of very similar CNN descriptors - and thus better distinguish very similar landmarks.

In terms of the overall runtime of the landmark based system, one could say that the current Matlab implementation allows for a relocalization on the Nordland test dataset in realtime - each 10 h journey was sampled at 0.1 Hz and the proposed landmark based approach requires about 10 seconds to match an image to the whole 10 h dataset. However, for application on larger datasets, this exhaustive comparison of all possible image pairings becomes infeasible.

Approaches to this problem can be inspired by techniques used in information retrieval, e.g. the visual vocabularies and inverted index used in FAB-MAP [Cummins and Newman, 2011]. However, there might be other approaches that incorporate the particular properties of changing environments. In very recent work presented in [Neubert et al., 2015a], we propose an approach that is particularly suited to reduce the number of comparisons for place recognition in these changing environments by exploiting intra database similarities. Typically the database images are not affected by environmental changes - the appearance changes happen between database and query images. We propose to split the place recognition in an *offline* phase (we already know the database images) and a *online* phase (we revisit the area where the database images have been captured and want to relocalize using a current query image). In the offline phase, the database images are preprocessed to prepare the selection of matching candidates in the subsequent online phase. This approach showed to save in many configurations about 90 % of the comparisons at the online stage without considerable place recognition performance loss.

Using higher level features and semantic information

Segmentation based landmarks can somehow be regarded as higher level features compared to a simple corner point or scale space extrema. Although the best performing approach, the SP-Grid, also deviates the most from this statement, it is reasonable at least for object proposals and also segment soups. The caption of Fig. 7.2 asked: Isn't it an appealing idea to have segmentation based landmarks? In my opinion, a main benefit of segmentation based landmarks is the potential semantic meaning that they provide. The evaluation in section 6.3 showed that superpixel words provide information about class labels. Object proposal showed to be a useful preprocessing step for object detection. Incorporating and exploiting this semantic meaning is presumably one of the most useful, interesting and challenging directions for future work.

In a nutshell, what is the recommended approach to compute image similarities in changing environments?

Use the CNN based landmarks and the Star-Hough approach. For local region detection: If there are only moderate appearance changes and severe viewpoint changes SIFT-DoG-6 is a reasonable choice. In all other cases (i.e. severe viewpoint changes) use SP-Grid. If runtime plays a role, use the binary hash functions (cf. section 7.2) and Compact Watershed for the SP-Grid. The influence of both choices on the place recognition quality can be seen in Fig. A.13 in the Appendix.

"This is the end. My only friend, the end."
Jim Morrison, 1967

"Every new beginning comes from some other beginning's end."
Seneca

A Appendix

A.1. Additional results Preemptive SLIC

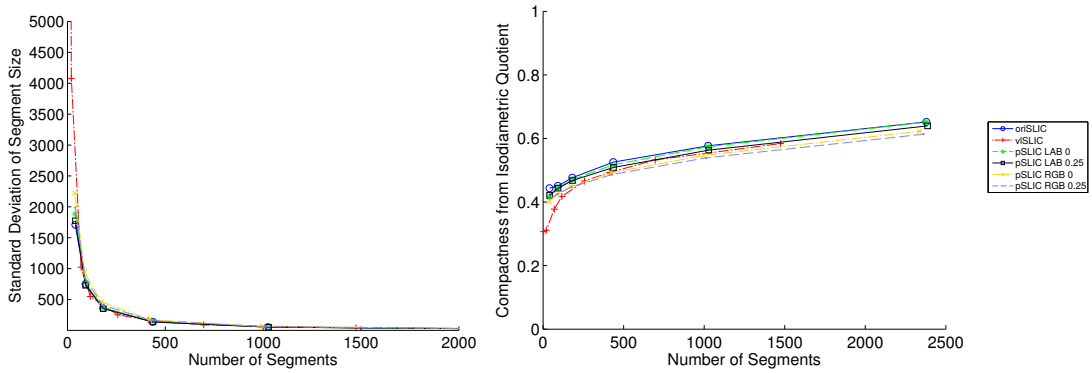


Fig. A.1.: Results of Watershed and Compact Watershed with varying compactness weights on variation of segment sizes and compactness from isodiametric quotient.

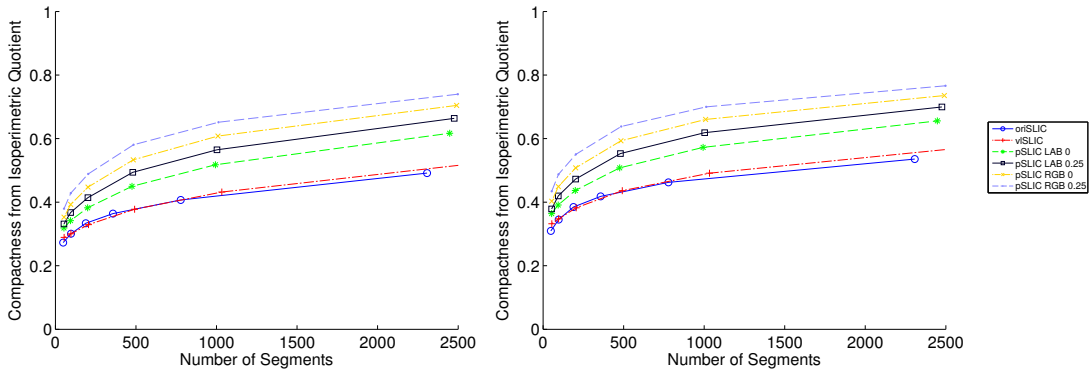


Fig. A.2.: Results of Preemptive SLIC on compactness from isoperimetric quotient. The left figure is for the original label images and the right figure for the smoothing with a morphological open of radius 5 as described in section 3.6.

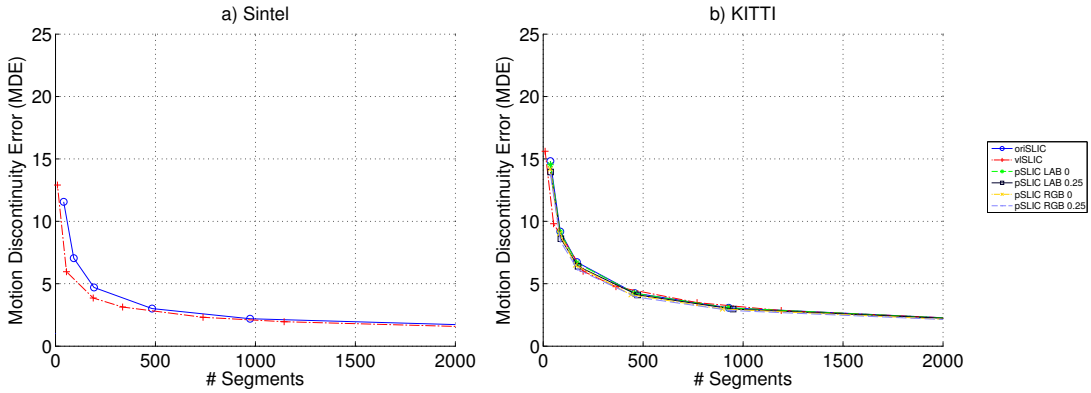


Fig. A.3.: Results of Preemptive SLIC on motion discontinuity error MDE.

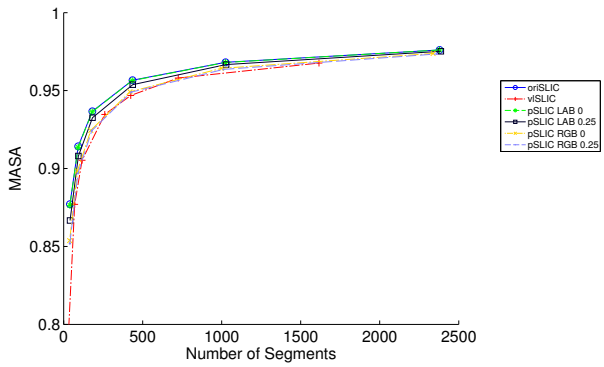


Fig. A.4.: Preemptive SLIC maximum achievable segmentation accuracy.

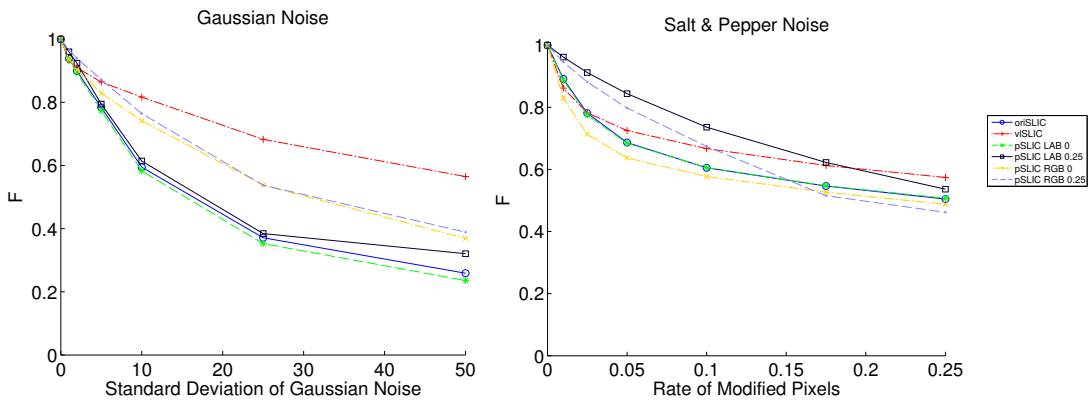


Fig. A.5.: Preemptive SLIC robustness to noise.

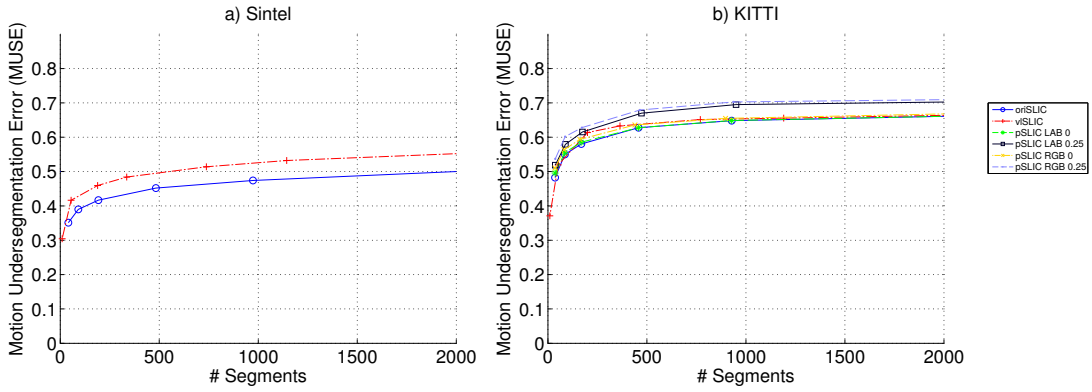


Fig. A.6.: Preemptive SLIC results on motion undersegmentation error.

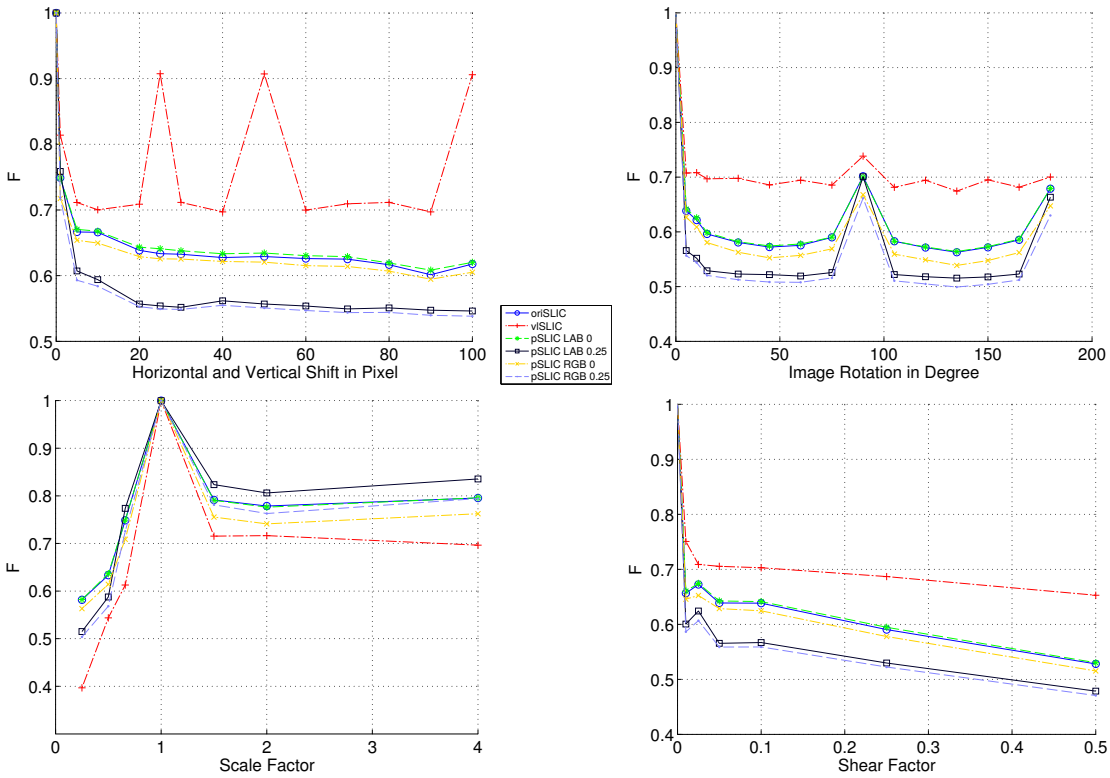


Fig. A.7.: Robustness of Preemptive SLIC towards shifting, rotation, scaling and shearing.

A.2. Additional results Compact Watershed

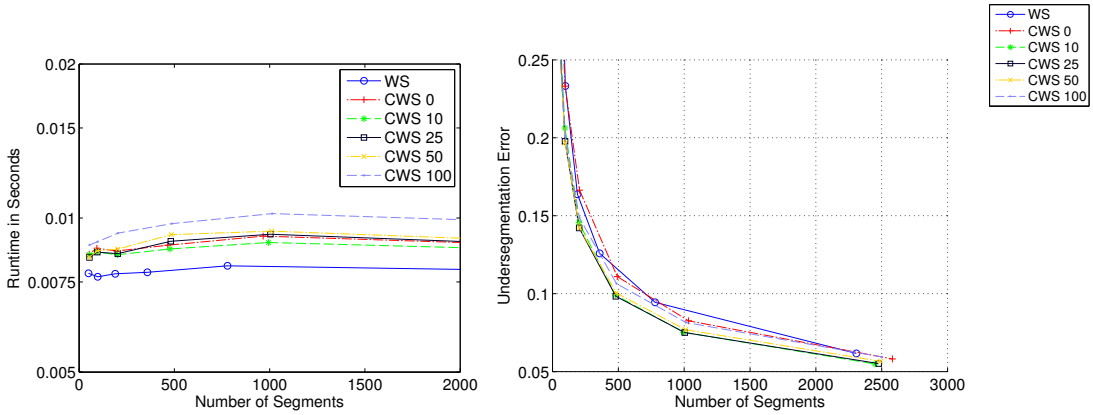


Fig. A.8.: Compact Watershed runtime and undersegmentation error. The runtime for WS does not include the time for computation of the seeds - including the detection of the local gradient minima that are used as seeds it runs even slower than the proposed CWS.

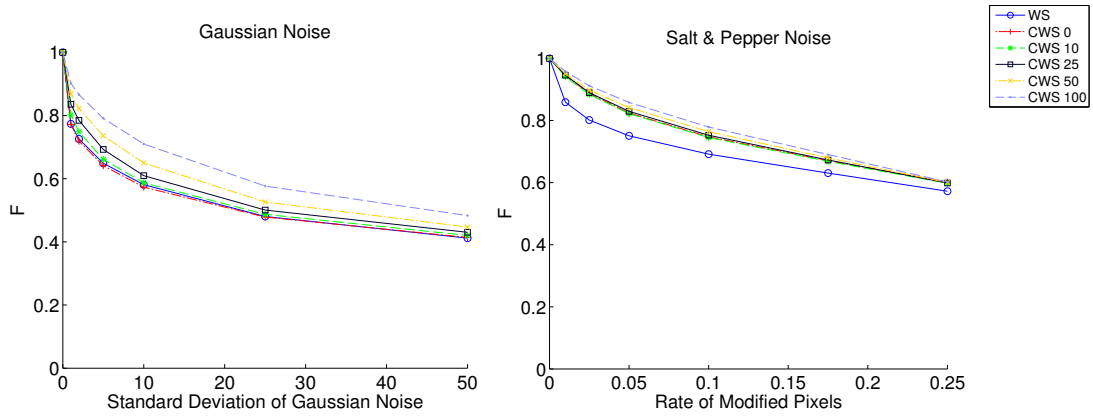


Fig. A.9.: Compact Watershed robustness to noise.

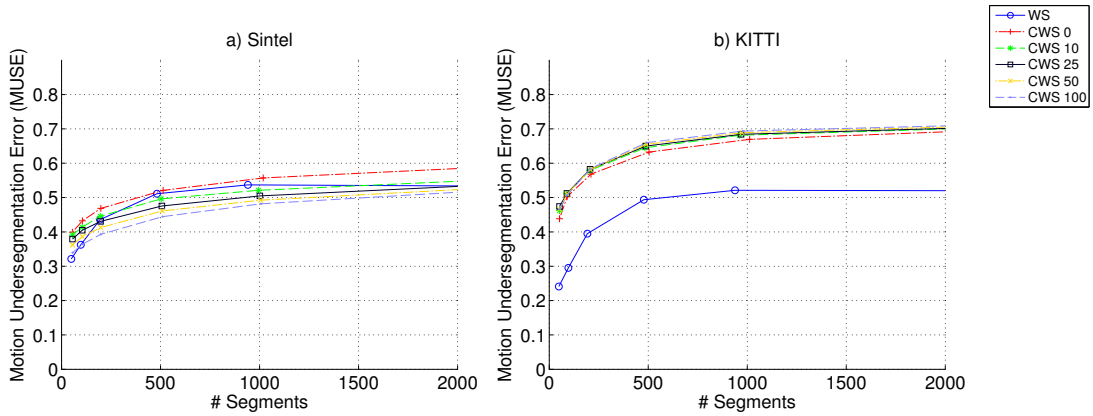


Fig. A.10.: Compact Watershed results on motion undersegmentation error.

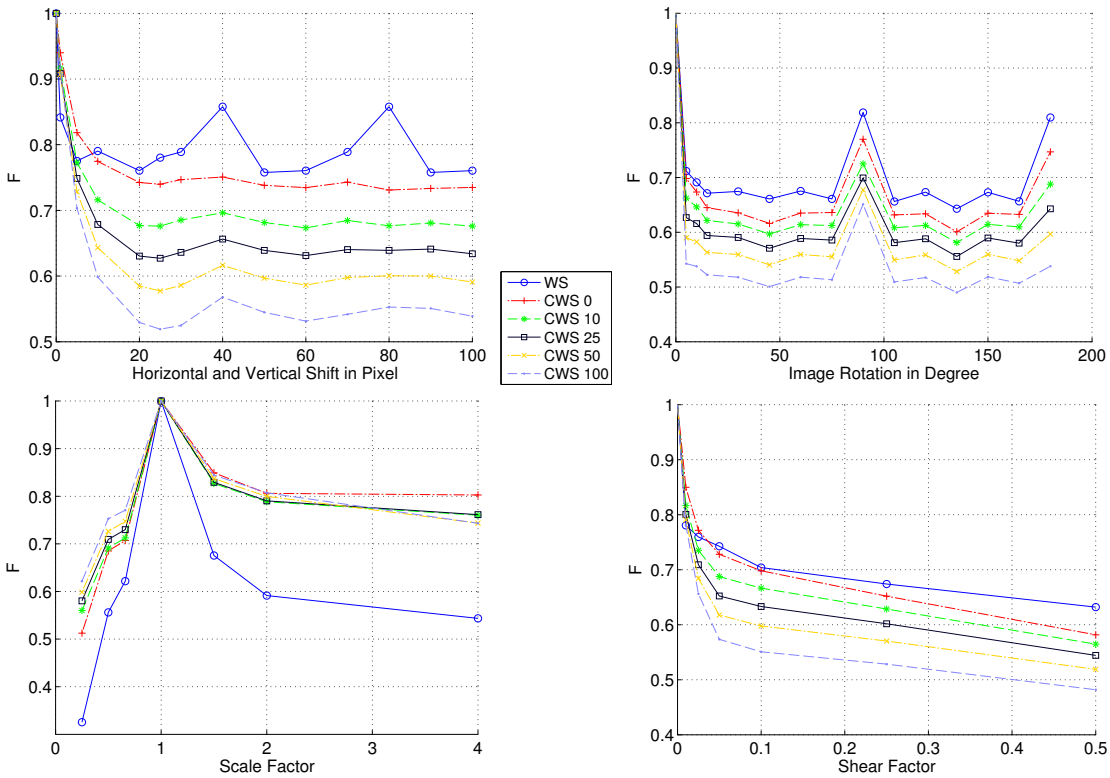


Fig. A.11.: Robustness of Compact Watershed towards shifting, rotation, scaling and shearing.

A.3. Additional details on the Nordland dataset

To extract the training and validation subset, we used the following *avconv* command on the Linux command line:

```
avconv -i nordlandsbanen.winter.sync.1920x1080.h264.nrk.mp4
-vsync vfr -r 2 -t 1800 -ss 02:02:00 -s 854x480 image-%05d.png
```

For testing we subsampled the 10 h videos at 0.1 frames per second using the following *avconv* command:

```
avconv -i nordlandsbanen.winter.sync.1920x1080.h264.nrk.mp4
-r 0.1 -vsync vfr -s 854x480 image-%05d.png
```

The quality of the image alignment is illustrated in Fig. A.12. While the alignment for distant objects is quit good, for objects close to the rails, there are some deviations.

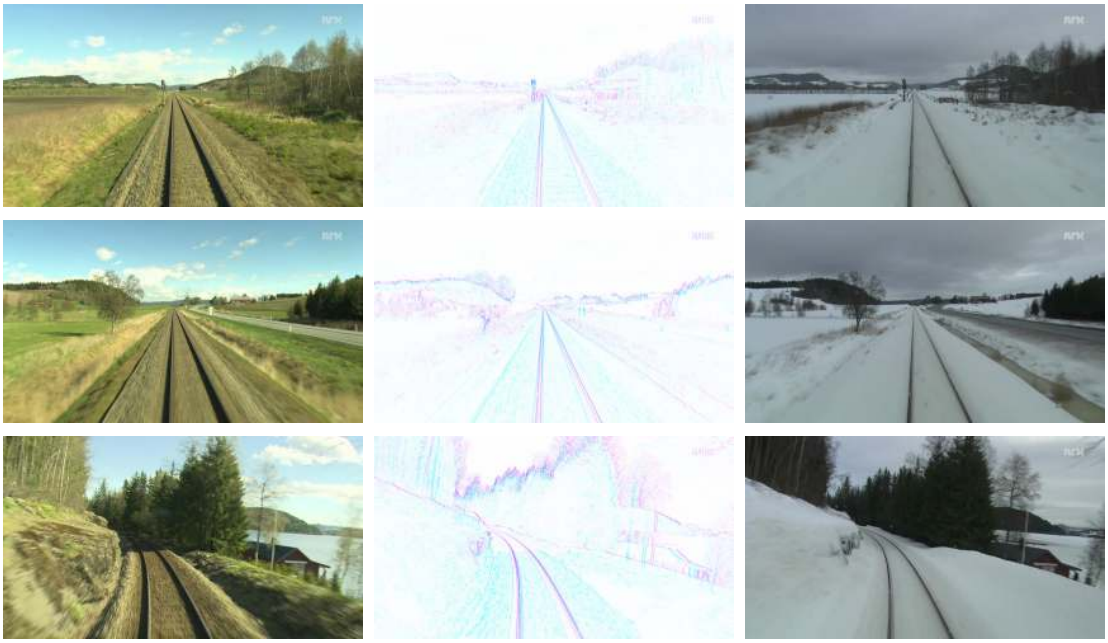


Fig. A.12.: Synchronization of Nordland images. The left and right images are the images from the spring and winter sequences. The overlayed edge maps in the middle illustrate the quality of the image alignment.

A.4. Using Compact Watershed for SP-Grid

Although the experimental results of the SP-Grid in chapter 7 have been obtained using SLIC, the Compact Watershed segmentation can also be used to create SP-Grid regions. The left part of Fig. A.13 compares the place recognition results when using SLIC or

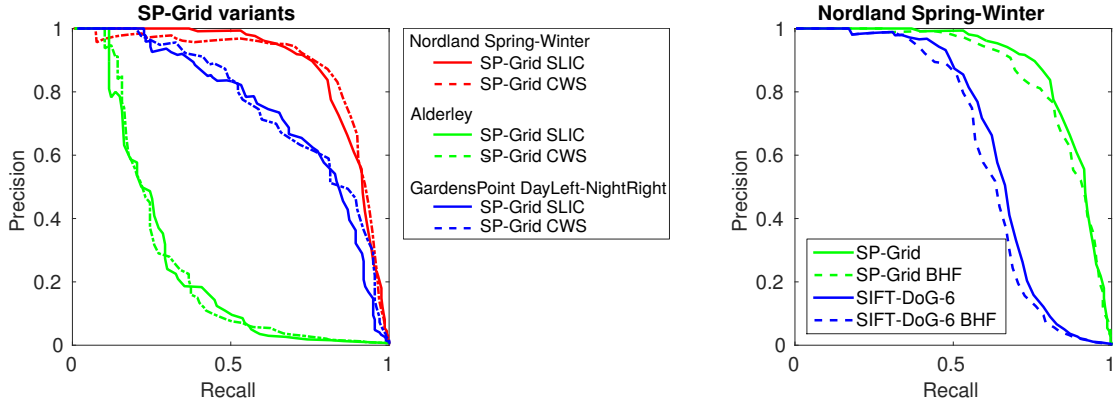


Fig. A.13.: *(left)* Influence of using SLIC or CWS for SP-Grid. *(right)* Influence of using binary hash functions (BHF) for approximation of the CNN based descriptor.

CWS. The superpixel benchmark of the first part of this thesis showed that SLIC creates segments that are better aligned with objects and that are more stable than those of CWS. This is in accordance with the place recognition results shown in Fig. A.13. Although, CWS provides a very similar overall performance, in particular at the high precision range of the Nordland curves, SLIC performs superior. Nevertheless, CWS is a reasonable choice and can be used if runtime is important.

A.5. Using binary hash functions in combination with the proposed local landmarks

CWS ca be used speedup the lcoal region detection of the SP-Grid. However, the detection of the regions is only done once per image and the *descriptor distance computation* is done for each image comparison. Typically, the number of image comparisons is *much* larger than the number of images, e.g. given two image sequences, each with n images, n^2 image comparisons are necessary for an exhaustive comparison. Moreover, in case of 50 landmarks per image, each image comparison includes 2,500 descriptor comparisons. Therefore, a speedup of the image comparison has a larger effect than of the region detection. Section 7.2 presented an approach from the literature to speed up the comparisons based on binary hash functions. The resulting slight decrease in place recognition performance can be seen in the right part of Fig. A.13.

Bibliography

R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(11):2274–2282, 2012. ISSN 0162-8828.

B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(11), 2012. ISSN 0162-8828.

P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 33, 2011.

P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marqués, and J. Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

H. Badino, D. F. Huber, and T. Kanade. Real-time topometric localization. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2012.

R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999. ISBN 020139829X.

S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92, 2011.

D. H. Ballard. Readings in computer vision: Issues, problems, principles, and paradigms. chapter Generalizing the Hough Transform to Detect Arbitrary Shapes, pages 714–725. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987. ISBN 0-934613-33-8. URL `http://dl.acm.org/citation.cfm?id=33517.33574`.

T.D. Barfoot. Online visual motion estimation using fastslam with sift features. In *Proceedings of the IEEE RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 579–585, 2005. doi: 10.1109/IROS.2005.1545444.

K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003. ISSN 1532-4435.

J. Bauer, N. Sünderhauf, and P. Protzel. Comparing several implementations of two recently published feature detectors. In *Proceedings of the International Conference on Intelligent and Autonomous Systems*, 2007.

H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110, 2008.

S. Belongie and J. Malik. Matching with shape contexts. In *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 20–26, 2000. doi: 10.1109/IVL.2000.853834.

P. Biber and T. Duckett. Experimental analysis of sample-based maps for long-term slam. *International Journal of Robotics Research*, 28(1):20–33, 2009. ISSN 0278-3649.

A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 12(1):55–73, 1990. ISSN 0162-8828. doi: 10.1109/34.41384. URL `http://dx.doi.org/10.1109/34.41384`.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23 (11):1222–1239, 2001. ISSN 0162-8828. doi: 10.1109/34.969114. URL `http://dx.doi.org/10.1109/34.969114`.

C. R. Brice and C. L. Fennema. Scene analysis using regions. *Artificial Intelligence*, 1(34):205 – 226, 1970. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/0004-3702(70)90008-1. URL `http://www.sciencedirect.com/science/article/pii/0004370270900081`.

D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *Proceedings of the 11th European Conference on Computer Vision: Part IV (ECCV)*, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15560-X, 978-3-642-15560-4. URL `http://dl.acm.org/citation.cfm?id=1888089.1888148`.

M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 380–388, New York, NY, USA, 2002. ACM. ISBN 1-58113-495-9. doi: 10.1145/509907.509965. URL `http://doi.acm.org/10.1145/509907.509965`.

K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *Computing Research Repository (CoRR)*, abs/1405.3531, 2014. URL `http://arxiv.org/abs/1405.3531`.

Z. Chen, O. Lam, A. Jacobson, and M. Milford. Convolutional neural network-based place recognition. In *In proceedings of the Australasian Conference on Robotics and Automation*, December 2014. URL `http://eprints.qut.edu.au/79662/`.

Y. Cheng. Mean shift, mode seeking, and clustering. *Transactions of IEEE Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995. ISSN 0162-8828. doi: 10.1109/34.400568. URL `http://dx.doi.org/10.1109/34.400568`.

C. M. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 150–155, 2002. URL `http://dblp.uni-trier.de/db/conf/icpr/icpr2002-4.html#ChristoudiasGM02`.

W. Churchill and P. M. Newman. Practice makes perfect? managing and leveraging visual experiences for lifelong navigation. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2012.

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24, 2002.

P. Corke, R. Paul, W. Churchill, and P. Newman. Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, November 2013.

M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, 27(6):647–665, 2008.

M. Cummins and P. Newman. Appearance-only slam at large scale with fab-map 2.0. *International Journal of Robotics Research*, 30(9):1100–1123, 2011. ISSN 0278-3649. doi: 10.1177/0278364910385483. URL `http://dx.doi.org/10.1177/0278364910385483`.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.177. URL `http://dx.doi.org/10.1109/CVPR.2005.177`.

F. Dayoub and T. Duckett. An adaptive appearance-based map for long-term topological localization of mobile robots. In *In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2008.

J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez. Revisiting the vlad image representation. In *Proceedings of the 21st ACM International Conference on Multimedia*, pages 653–656, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2404-5. doi: 10.1145/2502081.2502171. URL `http://doi.acm.org/10.1145/2502081.2502171`.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

R. Diestel. *Graph Theory*. Number 173 in Graduate Texts in Mathematics. Springer, 2000. ISBN 0-387-98976-5.

H. Digabel and C. Lantuéjoul. *Quantitative Analysis of Microstructures in Materials Sciences, Biology and Medicine*, chapter Iterative Algorithms. 1978.

P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2013.

R. C. Dubes and A. K. Jain. Random field models in image analysis. *Journal of Applied Statistics*, 16(5):131–164, 1989.

A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 341–346. ACM, 2001. ISBN 1-58113-374-X.

I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(2): 222–234, 2014. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2013.122.

M. Everingham, L.v. Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2), 2010.

M. Everingham, S.M.A. Eslami, L.v. Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge - a retrospective. *International Journal of Computer Vision*, 2014.

P. F. Felzenszwalb and D. P. Huttenlocher. Image Segmentation Using Local Variation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 98+, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 0-8186-8497-6. URL `http://portal.acm.org/citation.cfm?id=794679`.

P. F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 2004.

R. Feynman, R. Leighton, and M. Sands. *The Feynman Lectures on Physics*, volume 1. Addison-Wesley, Boston, second edition, 1963.

L. R. Ford and D. R. Fulkerson. Maximal Flow through a Network. *Canadian Journal of Mathematics*, 8:399–404, 56. URL `http://www.rand.org/pubs/papers/P605/`.

K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theor.*, 21(1):32–40, 2006. ISSN 0018-9448. doi: 10.1109/TIT.1975.1055330. URL `http://dx.doi.org/10.1109/TIT.1975.1055330`.

F. Galasso, N.S. Nagaraja, T.J. Cardenas, T. Brox, and B.Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *In proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec 2013.

F. Ge, S. Wang, and T. Liu. Image-segmentation evaluation from the perspective of salient object extraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

T. Gevers and A. W. M. Smeulders. A comparative study of several color models for color image invariant retrieval. In *In Proceedings of the First International Workshop on Image Databases and Multi-Media Search (IDB-MMS '96)*, pages 17–26, 1996.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

R. B. Girshick. Fast R-CNN. *Computing Research Repository (CoRR)*, abs/1504.08083, 2015. URL `http://arxiv.org/abs/1504.08083`.

A. Glover, W. Maddern, M. Milford, and G. Wyeth. FAB-MAP + RatSLAM : appearance-based slam for multiple times of day. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2010.

R. C. Gonzalez and R. E. Woods. *Digital Image Processing (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2002. ISBN 0130946508.

S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, 2008. ISSN 0920-5691.

W. E. L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, Cambridge, MA, USA, 1990. ISBN 0-262-07130-4.

C. Gu, J. Lim, P. Arbelez, and J Malik. Recognition using regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

R. Haralick, K. Shanmugam, and I. Dinstein. Texture features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6), 1973.

C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. ISBN 0521540518.

K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *Computing Research Repository (CoRR)*, abs/1406.4729, 2014. URL `http://arxiv.org/abs/1406.4729`.

X. He, R. S. Zemel, and V. Mnih. Topological map learning from outdoor image sequences. *Journal of Field Robotics*, 23(11-12):1091–1104, 2006.

A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 327–340. ACM, 2001. ISBN 1-58113-374-X.

D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Transactions on Graphics*, 24, 2005.

P. Hough. Method and means for recognizing complex patterns. U.S. Patent 3.069.654, 1962.

N. Jacobs, N. Roman, and R. Pless. Consistent Temporal Variations in Many Outdoor Scenes. In *In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. doi: 10.1109/CVPR.2007.383258.

H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the 10th European Conference on Computer Vision: Part I (ECCV)*, pages 304–317, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-88681-5. doi: 10.1007/978-3-540-88682-2_24. URL `http://dx.doi.org/10.1007/978-3-540-88682-2_24`.

H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(9):1704–1716, 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.235. URL `http://dx.doi.org/10.1109/TPAMI.2011.235`.

H. Johannsson, M. Kaess, M.F. Fallon, and J.J. Leonard. Temporally scalable visual SLAM using a reduced pose graph. In *RSS Workshop on Long-term Operation of Autonomous Robotic Systems in Changing Environments*, 2012.

E. Johns and G.-Z. Yang. Feature co-occurrence maps: Appearance-based localisation throughout the day. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3212–3218. IEEE, 2013. ISBN 978-1-4673-5641-1. URL `http://dblp.uni-trier.de/db/conf/icra/icra2013.html#JohnsY13a`.

A. Kae, B. Marlin, and E. Learned-Miller. The shape-time random field for semantic video labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 272–279, June 2014. doi: 10.1109/CVPR.2014.42.

Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 506–513, Washington, DC, USA, 2004. IEEE Computer Society. URL `http://dl.acm.org/citation.cfm?id=1896300.1896374`.

D. Kim, S. M. Oh, and J.M. Rehg. Traversability classification for ugv navigation: a comparison of patch and superpixel representations. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3166–3173, Oct 2007. doi: 10.1109/IROS.2007.4399610.

G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Nara, Japan, November 2007.

P. Koniusz and K. Mikolajczyk. Segmentation based interest points and evaluation of unsupervised image segmentation methods. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2009.

K. Konolige and J. Bowman. Towards lifelong visual maps. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2009.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`.

J. F. Lalonde, A. A. Efros, and S. G. Narasimhan. Webcam Clip Art: Appearance and illuminant transfer from time-lapse sequences. *ACM Transactions on Graphics (SIGGRAPH Asia 2009)*, 28(5), December 2009.

K. I. Laws. *Textured image segmentation.* PhD thesis, Dept. Electrical Engineering, University of Southern California, January 1980.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0. doi: 10.1109/CVPR.2006.68. URL `http://dx.doi.org/10.1109/CVPR.2006.68`.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. ISSN 0028-0836. doi: 10.1038/nature14539. URL `http://dx.doi.org/10.1038/nature14539`.

B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision (ECCV Workshop)*, 2004.

V. Lepetit, J. Pilet, and P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 244–250, Washington, DC, USA, 2004. IEEE Computer Society. URL `http://dl.acm.org/citation.cfm?id=1896300.1896335`.

V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 775–781, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.288. URL `http://dx.doi.org/10.1109/CVPR.2005.288`.

A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2009.

C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, (1):99–112. ISSN 0162-8828. doi: 10.1109/tpami.2013.107.

M.-Y. Liu, O. Tuzell, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. ISSN 0018-9448. doi: 10.1109/TIT.1982.1056489. URL `http://dx.doi.org/10.1109/TIT.1982.1056489`.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *In proceedings of the Computer Vision and Pattern Recognition (CVPR)*, 2015.

D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0164-8. URL `http://dl.acm.org/citation.cfm?id=850924.851523`.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60, 2004.

S. Lowry. *Visual place recognition for persistent robot navigation in changing environments*. PhD thesis, Queensland University of Technology, 2014. URL `http://eprints.qut.edu.au/79404/`.

S. Lowry, M. Milford, and G. Wyeth. Transforming morning to afternoon using linear regression techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3950–3955, Hong Kong Convention and Exhibition Center, Hong Kong, 2014a. IEEE. URL `http://eprints.qut.edu.au/68142/`.

S. M. Lowry, G. F. Wyeth, and M. Milford. Towards training-free appearance-based localization: Probabilistic models for whole-image descriptors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 711–717, 2014b. doi: 10.1109/ICRA.2014.6906932. URL `http://dx.doi.org/10.1109/ICRA.2014.6906932`.

V Machairas, Etienne Decencière, and Thomas Walter. Waterpixels: Superpixels based on the watershed transformation. In *In Proceedings of the IEEE International Conference On Image Processing*, Paris, France, October 2014. doi: 10.1109/ICIP.2014.7025882.

W. Maddern and S. Vidas. Towards robust night and day place recognition using visible and thermal imaging. In *Proceedings of the Robotics Science and Systems Conference (RSS)*, 2012.

W. Maddern, M. Milford, and G. Wyeth. Cat-slam: Probabilistic localisation and mapping using a continuous appearance-based trajectory. *International Journal of Robotics Research*, 31(4):429–451, 2012. ISSN 0278-3649. doi: 10.1177/0278364912438273. URL `http://dx.doi.org/10.1177/0278364912438273`.

J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001. ISSN 0920-5691. doi: 10.1023/A:1011174803800. URL `http://dx.doi.org/10.1023/A:1011174803800`.

T. Malisiewicz and A. A. Efros. Improving spatial support for objects via multiple segmentations. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2007.

S. Manen, M. Guillaumin, and L. Van Gool. Prime object proposals with randomized prim's algorithm. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2013.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2001.

D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(5):530–549, 2004. ISSN 0162-8828. doi: 10.1109/TPAMI.2004.1273918. URL `http://dx.doi.org/10.1109/TPAMI.2004.1273918`.

J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2002. ISBN 1-901725-19-7.

C. McManus, W. Churchill, W. Maddern, A. Stewart, and P. Newman. Shady dealings: Robust, long- term visual localisation using illumination invariance. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, May 2014.

P. Meer and B. Georgescu. Edge detection with embedded confidence. *Transactions of IEEE Pattern Analysis and Machine Intelligence*, 23, 2001.

S. Meister, B. Jähne, and D. Kondermann. Outdoor stereo camera system for the generation of real-world benchmark data sets. *Optical Engineering*.

F. Meyer. Color image segmentation. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 1992.

B. Miĉuŝik and J. Koŝecká. Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry. In *Proceedings of the IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 625–632. IEEE, 2009. ISBN 978-1-4244-4442-7. doi: 10.1109/iccvw.2009.5457645. URL `http://dx.doi.org/10.1109/iccvw.2009.5457645`.

B. Miĉuŝik and J. Koŝecká. Multi-view superpixel stereo in urban environments. *International Journal of Computer Vision*, 89(1):106–119, 2010. ISSN 0920-5691. doi: 10.1007/s11263-010-0327-9. URL `http://dx.doi.org/10.1007/s11263-010-0327-9`.

K. Mikolajczyk and J. Matas. Improving descriptors for fast tree matching by optimal linear projection. In *Proceedings of the 11th International Conference on Computer Vision (ICCV)*, pages 1–8, Oct 2007. doi: 10.1109/ICCV.2007.4408871.

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(10):1615–1630, 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.188. URL `http://dx.doi.org/10.1109/TPAMI.2005.188`.

K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005. ISSN 0920-5691.

A. Mikulik, M. Perdoch, O. Chum, and J. Matas. Learning vocabularies over a fine quantization. *International Journal of Computer Vision*, 103(1):163–175, 2013. ISSN 0920-5691. doi: 10.1007/s11263-012-0600-1. URL `http://dx.doi.org/10.1007/s11263-012-0600-1`.

M. Milford and G. Wyeth. Persistent navigation and mapping using a biologically inspired slam system. *International Journal of Robotics Research*, 29(9):1131–1153, 2010. ISSN 0278-3649.

M. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

M. Milford, E. Vig, W. Scheirer, and D. Cox. Towards condition-invariant, top-down visual place recognition. In J. Katupitiya, J. Guivant, and R. Eaton, editors, *Proceedings of the Australasian Conference on Robotics and Automation (ACRA)*, pages 1–10, University of New South Wales, Sydney, NSW, 2013. Australian Robotics & Automation Association. URL `http://eprints.qut.edu.au/66634/`.

M.J. Milford, G.F. Wyeth, and D. Prasser. Ratslam: a hippocampal model for simultaneous localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 403–408 Vol.1, April 2004. doi: 10.1109/ROBOT.2004.1307183.

A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel Lattices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

H. P. Moravec. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. PhD thesis, Stanford, CA, USA, 1980. AAI8024717.

R. Mur-Artal and J. Tardos. Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM. In *In proceedings of Robotics: Science and Systems (RSS)*, 2015.

G. L. Murphy. *The Big Book of Concepts*. MIT Press, Boston, Mass., 2002. ISBN 9780-262134095.

T. Naseer, L. Spinello, W. Burgard, and C. Stachniss. Robust visual robot localization across seasons using network flows. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014. URL `http://www.informatik.uni-freiburg.de/~naseer/publications/naseer14aaai.pdf`.

F. Navarro, M. Escudero-Violo, and J. Bescos. Sp-sift: enhancing sift discrimination via super-pixel-based foreground-background segregation. *Electronics Letters*, 50(4): 272–274, Feb. 2014. ISSN 0013-5194. doi: 10.1049/el.2013.3949.

P. Neubert and P. Protzel. Superpixel benchmark and comparison. In *Proceedings of the Forum Bildverarbeitung*, 2012.

P. Neubert and P. Protzel. Evaluating superpixels in video: Metrics beyond figure-ground segmentation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2013.

P. Neubert and P. Protzel. Compact Watershed and Preemptive SLIC: On improving trade-offs of superpixel segmentation algorithms. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2014.

P. Neubert and P. Protzel. Benchmarking superpixel descriptors. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2015a.

P. Neubert and P. Protzel. Local region detector + CNN based landmarks for practical place recognition in changing environments. In *Proceedings of the European Conference on Mobile Robotics (ECMR)*, 2015b.

P. Neubert, N. Sünderhauf, and P. Protzel. Appearance change prediction for long-term navigation across seasons. In *Proceedings of the European Conference on Mobile Robotics (ECMR)*, 2013.

P. Neubert, S. Schubert, and P. Protzel. Exploiting intra database similarities for selection of place recognition candidates in changing environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Visual Place Recognition in Changing Environments*, 2015a.

P. Neubert, N. Sünderhauf, and P. Protzel. Superpixel-based appearance change prediction for long-term navigation across seasons. *Journal of Robotics and Autonomous Systems*, 69(0):15 – 27, 2015b. ISSN 0921-8890.

D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0. doi: 10.1109/CVPR.2006.264. URL `http://dx.doi.org/10.1109/CVPR.2006.264`.

P. Nowozin, S.and Gehler and C. Lampert. On parameter learning in crf-based approaches to object class image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

J. Papon, T. Kulvicius, E.E. Aksoy, and F. Worgotter. Point cloud video object segmentation using a persistent supervoxel world-model. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3712–3718, Nov 2013. doi: 10.1109/IROS.2013.6696886.

Theodosios Pavlidis. Segmentation of pictures and maps through functional approximation. *Computer Graphics and Image Processing*, 1(4):360 – 372, 1972. ISSN 0146-664X. doi: http://dx.doi.org/10.1016/0146-664X(72)90021-4. URL `http://www.sciencedirect.com/science/article/pii/0146664X72900214`.

E. Pepperell, P.I. Corke, and M.J. Milford. All-environment visual place recognition with smart. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1612–1618, May 2014.

F. Perronnin. Universal and adapted vocabularies for generic visual categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(7): 1243–1256, 2008. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.70755. URL `http://dx.doi.org/10.1109/TPAMI.2007.70755`.

F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2007. doi: 10.1109/CVPR.2007.383266. URL `http://dx.doi.org/10.1109/CVPR.2007.383266`.

A. Ranganathan, S. Matsumoto, and D. Ilstrup. Towards illumination invariance for visual localization. In *In proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3791–3798, May 2013. doi: 10.1109/ICRA.2013.6631110.

D. Ravichandran, P. Pantel, and E. Hovy. Randomized algorithms and nlp: Using locality sensitive hash function for high speed noun clustering. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 622–629, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219917. URL `http://dx.doi.org/10.3115/1219840.1219917`.

A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *Computing Research Repository (CoRR)*, abs/1403.6382, 2014. URL `http://arxiv.org/abs/1403.6382`.

C.Y. Ren and I. Reid. gslic: a real-time implementation of slic superpixel segmentation. Technical report, University of Oxford, Department of Engineering Science, 2011.

X. Ren and J. Malik. Learning a classification model for segmentation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2003.

J.B.T.M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Journal of Fundamenta Informaticae*, 2000.

A. Rosenfeld and B. Lipkin. Texture synthesis. *Picture Processing and Psychopictories*, 1970.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pages 1–42, April 2015. doi: 10.1007/s11263-015-0816-y.

B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008.

A. Schick, M. Fischer, and R. Stiefelhagen. Measuring and evaluating the compactness of superpixels. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*.

Y. A. Sheikh, E. Khan, and T. Kanade. Mode-seeking by medoidshifts. In *Proceedings of the Eleventh IEEE International Conference on Computer Vision (ICCV)*, number 141, October 2007.

J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proceedings of the the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 731–737, 1997.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(8):888–905, 2000. ISSN 0162-8828. doi: 10.1109/34.868688. URL `http://dx.doi.org/10.1109/34.868688`.

J. Shi and C. Tomasi. Good features to track. Technical report, Ithaca, NY, USA, 1993.

J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1), 2009.

J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV)*, pages 1470–1477 vol.2, Oct 2003. doi: 10.1109/ICCV.2003.1238663.

H. Steinhaus. Sur la division des corp materiels en parties. *Bulletin de l'Acadmie Polonaise des Sciences*, 1:801–804, 1956.

R. Stolkin. *Scene Reconstruction, Pose Estimation and Tracking*. I-Tech Education and Publ., 2007. ISBN 9783902613066. URL `https://books.google.de/books?id=9Z8MHwAACAAJ`.

J Strassburg, R. Grzeszick, L. Rothacker, and G. Fink. On the influence of superpixel methods for image parsing. In *In proceedings of the International Conference on Computer Vision Theory and Applications (Visapp)*, 2015.

N. Sünderhauf and P. Protzel. Brief-gist - closing the loop by simple means. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2011.

N. Sünderhauf and P. Protzel. Switchable constraints vs. max-mixture models vs. rrr – a comparison of three approaches to robust pose graph slam. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2013.

N. Sünderhauf, P. Neubert, and P. Protzel. Are we there yet? challenging seqslam on a 3000 km journey across all four seasons. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Workshop on Long-Term Autonomy*, 2013.

N. Sünderhauf, P. Neubert, and P. Protzel. Predicting the change - a step towards lifelong operation in everyday environments. In *Proceedings of the Robotics: Science and Systems (RSS), Robotics Challenges and Vision Workshop*, 2013.

N. Sünderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. *Computing Research Repository (CoRR)*, abs/1501.04158, 2015a. URL `http://arxiv.org/abs/1501.04158`.

N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. In *Proceedings of the Robotics: Science and Systems*, Rome, Italy, July 2015b.

R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010. ISBN 1848829345, 9781848829343.

J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. Proceedings of the European Conference on Computer Vision (ECCV), 2010.

A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. Proceedings of the International Conference on Computer Vision (ICCV), 2003.

J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013. URL `https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013`.

B. Upcroft, C. McManus, W. Churchill, W. Maddern, and P. Newman. Lighting invariant urban street classification. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1712–1718, May 2014. doi: 10.1109/ICRA.2014.6907082.

C. Valgren and A. J. Lilienthal. SIFT, SURF and seasons: Long-term outdoor localization using local features. In *Proceedings of the European Conference on Mobile Robots (ECMR)*, pages 253–258, Freiburg, Germany, September 19–21 2007.

C. Valgren and A. J. Lilienthal. Sift, surf & seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems*, 58(2):149–156, 2010. ISSN 0921-8890.

K.E.A. van de Sande, C.G.M. Snoek, and A.W.M. Smeulders. Fisher and vlad with flair. In *Proceedings of the IEEE Conference onComputer Vision and Pattern Recognition (CVPR)*, pages 2377–2384, June 2014. doi: 10.1109/CVPR.2014.304.

M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 7578 of *Lecture Notes in Computer Science*, pages 13–26. Springer, 2012. ISBN 978-3-642-33785-7. URL `http://dblp.uni-trier.de/db/conf/eccv/eccv2012-7.html#BerghBRCG12`.

M. Van den Bergh, X Boix, G. Roig, and L. Van Gool. Seeds: Superpixels extracted via energy-driven sampling. *Computing Research Repository (CoRR)*, abs/1309.3848, 2013. URL `http://arxiv.org/abs/1309.3848`.

M. Varma and A. Zisserman. Texture classification: are filter banks necessary? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–691–8 vol.2, June 2003. doi: 10.1109/CVPR.2003.1211534.

A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision ECCV 2010*, volume 6315 of *Lecture Notes in Computer Science*, pages 268–281. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15554-3. doi: 10.1007/978-3-642-15555-0_20. URL `http://dx.doi.org/10.1007/978-3-642-15555-0_20`.

A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

O. Veksler, Y. Boykov, and P. Mehrani. Superpixels and supervoxels in an energy optimization framework. In *Proceedings of the the 11th European Conference on Computer Vision: Part V (ECCV)*, pages 211–224, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15554-5, 978-3-642-15554-3. URL `http://dl.acm.org/citation.cfm?id=1888150.1888167`.

S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1323–1330, Nov 2011. doi: 10.1109/ICCV.2011.6126385.

X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.

J. Wu. *Rotation Invariant Classification of 3D Surface Texture Using Photometric Stereo*. PhD thesis, Heriot-Watt University, Edinburgh, 2003.

Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 15(11):1101–1113, 1993. ISSN 0162-8828. doi: 10.1109/34.244673. URL `http://dx.doi.org/10.1109/34.244673`.

J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492, June 2010. doi: 10.1109/CVPR.2010.5539970.

C. Xu and J. Corso. Evaluation of super-voxel methods for early video processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

W. Zhang and J Kosecka. Localization based on building recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995. ISSN 0004-3702. doi: 10.1016/0004-3702(95)00022-4. URL `http://dx.doi.org/10.1016/0004-3702(95)00022-4`.

B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *Computing Research Repository (CoRR)*, abs/1412.6856, 2014. URL `http://arxiv.org/abs/1412.6856`.

C. L. Zitnick and P. Dollar. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision (ECCV)*. Proceedings of the European Conference on Computer Vision (ECCV), September 2014. URL `http://research.microsoft.com/apps/pubs/default.aspx?id=220569`.

List of Figures

1.1 Relation of the parts of this work . . . 12
1.2 Place recognition in changing environments . . . 13

2.1 Checkerboard illusion . . . 22
2.2 Segmentation is hard . . . 23
2.3 Different oversegmentation results . . . 25
2.4 Example NC segmentation . . . 32
2.5 Example FH segmentation . . . 34
2.6 Example EAMS segmentation . . . 35
2.7 Example QS segmentation . . . 36
2.8 Example WS segmentation . . . 37
2.9 Example ERS segmentation . . . 38
2.10 Example SLIC segmentation . . . 38
2.11 Example SEEDS segmentation . . . 39
2.12 Example VEK segmentation . . . 40
2.13 Example segmentations . . . 43

3.1 Why we need a standardized benchmark . . . 47
3.2 BSDS example data . . . 51
3.3 Grorund truth optical flow criteria . . . 53
3.4 KITTI examples . . . 54
3.5 Sintel examples . . . 55
3.6 Superpixel results: runtime . . . 58
3.7 Undersegmentation error visualization . . . 61
3.8 Superpixel results: boundary recall and undersegmentation error . . . 62
3.9 Superpixel results: maximum achievable segmentation accuracy . . . 63
3.10 Illustration of affine image transformations . . . 65
3.11 MUSE and MDE illustration . . . 68
3.12 Superpixel results: affine transformations . . . 69
3.13 Superpixel results: noise . . . 70
3.14 Superpixel results: MUSE . . . 70
3.15 Superpixel results: size and isodiametric quotient . . . 76
3.16 Superpixel results: isoperimetric quotient . . . 76
3.17 Superpixel results: MDE . . . 77
3.18 Superpixel results: concatenated flow fields . . . 77

4.1 Superpixel trade-offs . . . 82
4.2 Example segmentations Preemptive SLIC and Compact Watershed . . . 83
4.3 Examples SLIC and Preemptive SLIC . . . 88
4.4 Preemptive SLIC results: Runtime . . . 89
4.5 Preemptive SLIC results: Quality . . . 90
4.6 Compact Watershed priority queues . . . 91
4.7 CWS examples compactness . . . 93
4.8 CWS examples number of segments . . . 95
4.9 CWS examples image sizes . . . 96
4.10 Examples Watershed and Compact Watershed . . . 97
4.11 Compact Watershed results: size and isodiametric quotient . . . 98
4.12 Compact Watershed results: isoperimetric quotient . . . 98
4.13 Compact Watershed results: MDE . . . 99
4.14 Compact Watershed results: quality . . . 99
4.15 Example pSLIC segmentation . . . 101
4.16 Example CWS segmentation . . . 101

5.1 Localization overview . . . 105
5.2 Examples from changing environments . . . 106
5.3 Nordland examples images . . . 109
5.4 Nordland track . . . 109
5.5 Gardens Point example images . . . 111
5.6 Gardens Point sequence . . . 111
5.7 Gardens Point example images . . . 112
5.8 Timeline of approaches to place recognition in changing environments . . . 115
5.9 Creation of precision-recall curves . . . 121

6.1 ACP overview . . . 124
6.2 ACP components . . . 126
6.3 Overlap criterion . . . 130
6.4 Propagating segments . . . 133
6.5 Results on overlap criterion . . . 134
6.6 ACP components 2 . . . 135
6.7 SUN examples . . . 138
6.8 SUN results . . . 139
6.9 ACP training . . . 141
6.10 ACP prediction . . . 143
6.11 Example words and translations . . . 145
6.12 Example predictions . . . 146
6.13 ACP with BRIEF-GIST . . . 147
6.14 ACP with SeqSLAM validation . . . 149
6.15 ACP parameter evaluation . . . 151
6.16 ACP other seasons . . . 152
6.17 ACP prediction for other seasons . . . 153

6.18 ACP on Nordland test set 154

7.1 Place recognition based on local regions 158
7.2 Example superpixel landmarks 166
7.3 Example landmarks: SIFT-DoG-6 167
7.4 Example landmarks: SP-Soup 168
7.5 Grid evaluation 170
7.6 SP-Grid superpixel grouping 172
7.7 SP-Grid layer 172
7.8 Example landmarks: SP-Grid 174
7.9 Superpixel-based reweighting 176
7.10 Example landmarks: MCG 178
7.11 Example landmarks: EdgeBoxes 179
7.12 Different IoU -based metrics 183
7.13 Local region detector overlap results 184
7.14 Webcam example images 186
7.15 Detector sensitivity to changing environments 188
7.16 Detector sensitivity to changing environments 189
7.17 Landmark results: Proposed matching scheme 190
7.18 Comparison Grid and SP-Grid on Gardens Point 192
7.19 Grid vs. SP-Grid 193
7.20 Results on SIFT-DoG 195
7.21 Results on Nordland Spring-Winter 196
7.22 Results on Nordland spring-winter with shift 198
7.23 Example matchings Nordland Spring-Winter 198
7.24 Results on Nordland Fall-Winter and Summer-Fall 199
7.25 Results on Gardens Point DayRight-NightRight 199
7.26 Results on Alderley 200
7.27 Example matchings GardensPoint 201
7.28 Example matchings Alderley 201

8.1 Comparison to state of the art 210
8.2 Example matchings 212

A.1 Preemptive SLIC results: size and isodiametric quotient 215
A.2 Preemptive SLIC results: isoperimetric quotient 215
A.3 Preemptive SLIC results: MDE 216
A.4 Preemptive SLIC results: runtime and USE 216
A.5 Preemptive SLIC results: noise 216
A.6 Preemptive SLIC results: MUSE 217
A.7 Preemptive SLIC results: affine transformations 217
A.8 Compact Watershed results: runtime and USE 218
A.9 Compact Watershed results: noise 218
A.10 Compact Watershed results: MUSE 218

A.11 Compact Watershed results: affine transformations 219
A.12 Nordland synchronization 220
A.13 Influence of CWS and BHF 221

List of Tables

2.1 Overview of compares superpixel algorithms 42

3.1 Overview of conducted superpixel algorithm evaluations 50
3.2 Listing of the compared implementations 57
3.3 Parameters that are different from defaults. 58

6.1 ACP experiments 144
6.2 ACP default parameters 144

7.1 Parameters for keypoint, plain superpixel and segment soup approaches . 169
7.2 Parameters for SP-Grid 173
7.3 Parameters for object proposal algorithms 178

Versicherung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Bei der Auswahl und Auswertung des Materials sowie der Herstellung des Manuskripts habe ich Unterstützungsleistungen von folgenden Personen erhalten:

- In den Kapiteln 5 und 6 werden teilweise Inhalte verwendet, die im Rahmen gemeinsamer Veröffentlichungen mit Niko Sünderhauf erarbeitet und publiziert wurden. Die betroffenen Stellen sind in der Arbeit kenntlich gemacht.
- Sebastian Geißler, Niko Sünderhauf, Kathrin Baar und Thomas Wiener haben die Arbeit mit ihren Korrekturvorschlägen wesentlich verbessert.
- Ryan Wheeler von EnglischAlternativ hat mich bei Fragen der englischen Rechtschreibung und Grammatik unterstützt.

Weitere Personen waren an der Abfassung der vorliegenden Arbeit nicht beteiligt. Die Hilfe eines Promotionsberaters habe ich nicht in Anspruch genommen. Weitere Personen haben von mir keine geldwerten Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Chemnitz, 17. August 2015

Peer Neubert

Theses

1. Superpixels are an established and useful intermediate level image representation to deal with the ill-posed formulation of the image segmentation problem.
2. There are various approaches to generate superpixels, each with individual properties and different segmentation results. Important properties of superpixel segmentation algorithms are segmentation quality, runtime, stability and compactness.
3. For some of these properties, there are established evaluation methodologies available. In particular, for segmentation stability and compactness this is not the case. Datasets with known ground truth-optical flow can be used to evaluate the stability and compactness of superpixel segmentation algorithms.
4. There is a lack of standardized benchmarks for this class of algorithms. In particular, there are contradicting results in the literature. The presented existing and novel metrics can be used for profound evaluation of available algorithms.
5. The SLIC superpixel algorithm is based on a local k-means clustering with a holistic preemption criterion. The runtime can be significantly decreased by the incorporation of a local preemption criterion.
6. The seeded watershed superpixel algorithm is a very fast oversegmentation algorithm. It creates superpixel with strongly varying shape and size. The incorporation of compactness constraints in the watershed computation can be used to create regularly shaped, compact superpixels at the same high speed.
7. Vision is an established source of information for mobile robot navigation. Operating robots over the course of days, weeks or months based on visual navigation requires repeated recognition of places - despite severe appearance changes as they are for example induced by illumination changes, day-night cycles, changing weather or seasons. Existing visual place recognition methods show severe problems in the presence of such changes.
8. Place recognition in changing environments can benefit from learning systematic appearance changes between different environmental conditions. In particular from the novel idea of prediction of appearance changes.
9. The combination of superpixel segmentations and descriptors can be used to learn superpixel vocabularies. The words of these superpixel vocabularies provide valuable amounts of semantic meaning. It is possible to learn systematic appearance changes based on superpixel vocabularies and to predict images from one environmental condition to another.
10. For a wide range of configurations of the superpixel vocabularies and the appearance change prediction approach, existing holistic place recognition approaches can benefit from an additional prediction step. A limitation of the approach are strong requirements on the training data.
11. Holistic approaches to place recognition fail in presence of viewpoint changes. Using local image features is an alternative. This is commonly known and accepted in the literature.

12. The established interest point approach SIFT comprises a detection and a description step. With increasing severity of the appearance changes, the description step fails before the detection step. For severe appearance changes, the detection step is strongly affected as well.
13. Descriptors obtained from the lower levels of convolutional neural networks, that are known to be beneficial for holistic image description in changing environments, can also be used in combination with local region detectors. The computational effort for using these descriptors is feasible for practical applications.
14. The proposed combination of star graph models and Hough voting can be used to incorporate the spatial arrangement of local image features in the computation of the image similarity. This approach is particularly suited for features with low spatial position precision and high outlier rates as they are expected in the presence of appearance changes. Incorporation of this approach improves the performance of local region based approaches to place recognition in changing environments.
15. The superpixel segment soup, the multiscale superpixel grid, superpixel based importance reweigthing of scale space extrema, and the usage of object proposals are novel superpixel-related approaches to local region detection for place recognition in changing environments.
16. The proposed superpixel grid is more robust to viewpoint changes than a fixed grid.
17. The established approach to measure the repeatability of local region detectors incorporates a normalization of the region size. Replacing this normalization with a novel normalization based on the detections due to chance, reveals a dependency of the detector performance on the scale of the region.
18. This measure indicates a benefit from larger region sizes than those of the typically used regions obtained from difference of Gaussian based scale space extrema. The combination of these larger regions with superpixel based reweighting shows promising results.
19. The proposed combination of local region detectors, descriptors based on convolutional neural networks, and the star graph model and Hough voting based image matching performs superior to available approaches to place recognition in changing environments.
20. The choice of the used local region detectors depends on the expected amounts of environmental and viewpoint changes: Difference of Gaussian based scale space extrema are a reasonable choice for minor appearance changes and severe viewpoint changes. For any type of appearance change and somewhat aligned images, the multiscale superpixel grid is preferable. For severe appearance changes, e.g. between day and night, independent of the amount of viewpoint changes, the multiscale superpixel grid performs superior.

Curriculum Vitae

Personal Information

Name	Peer Neubert
Academic Degree	Dipl. Inf.
Date of Birth	November 10th, 1984

Research Experience

01/2010 to date	Research associate at Technische Universität Chemnitz, Germany
09/2007 to 02/2008	Research internship at Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS), Toulouse, France

Refereed Publications

16 refereed publications: 1 journal article, 12 conference papers, 3 workshop contributions

Education

10/2003 to 12/09	Computer science at Technische Universität Chemnitz, Germany Degree: Diplom Informatiker Grade: 1.1, Grade thesis: 1.0
08/1995 to 06/2003	Samuel-von-Pufendorf-Gymnasium Flöha, Germany Degree: Abitur (German University Entrance Qualification) Grade: 1.1

Awards and scholarships

06/2013	Robotics: Science and Systems (RSS) "Robotics Challenges and Vision Workshop", 3rd Best Paper Award
01/2010 to 12/2013	ESF Landesinnovationspromotionsstipendium
09/2007 to 02/2008	LEONARDO scholarship in the Leonardo da Vinci program
10/2003 to 12/2009	e-fellows.net scholarship

Titel (Übersetzung)
Superpixel und ihre Anwendung zur Wiedererkennung von Orten in sich verändernden Umgebungen

Inhaltsverzeichnis (Übersetzung)

1 **Einleitung und Motivation** 11
1.1 Forschungsfragen und Beiträge 14
1.2 Publikationen und Zusammenarbeit 16
1.3 Open Source Software Veröffentlichungen 17
1.4 Persönliche Anmerkungen 17

Teil 1: Superpixel Algorithmen

2 **Eine Einführung zu Superpixel Segmentierungen** 21
2.1 Das Problem der Bildsegmentierung 21
2.2 Was sind Superpixel? 24
2.3 Erstellen von Superpixel Segmentierungen 26

3 **Über den Vergleich von Superpixel Verfahren** 45
3.1 Einleitung 46
3.2 Beschreibung des Experimentalaufbaus 49
3.3 Laufzeit 58
3.4 Qualität 59
3.5 Stabilität 64
3.6 Kompaktheit 71
3.7 Zusammenfassung 78

4 **Zwei neue Superixel Algorithmen: Preemptive SLIC und Compact Watershed** 81
4.1 Trade-offs in Superpixel Algorithmen 81
4.2 Preemptive SLIC: Beschleunigung eines hoch qualitativen Algorithmus 84
4.3 Ergebnisse von Preemptive SLIC 87
4.4 Compact Watershed: Von Übersegmentierungen zu Superpixeln 89
4.5 Ergebnisse von Compact Watershed 96
4.6 Zusammenfassung 100

Teil 2: Anwendung von Superpixeln zur Wiedererkennung von Orten in sich verändernden Umgebungen

5 **Das Problem der Wiedererkennung von Orten in sich verändernden Umgebungen** 105
5.1 Datensätze 108
5.2 Existierende Verfahren 112
5.3 Auswertung basierend auf Precision-Recall Kurven 121

6 **Superpixel basierte Vorhersage von Veränderungen** 123
6.1 Konzept der Superpixel basierte Vorhersage von Veränderungen 123
6.2 Deskriptoren für Superpixel 126
6.3 Superpixel Vocabulare 135
6.4 SP-ACP: Implementierung des Konzepts 140
6.5 Ergebnisse auf dem Nordland Datensatz 143
6.6 Einschränkungen und mögliche Erweiterungen 155

7 **Landmarken basierte Wiedererkennung von Orten in sich verändernden Umgebungen** 157
7.1 Übersicht der Bestandteile 157
7.2 Deskriptoren für lokale Bildmerkmale aus Convolutional Neural Networks 159
7.3 Das neue Star-Hough Matching Schema 162
7.4 Lokale Bildmerkmale basierend auf Superpixeln 166
7.5 Über die Wiederholpräzision von lokalen Bildmerkmalen 179
7.6 Ergebnisse zur Wiederholpräzision in sich verändernden Umgebungen 183
7.7 Ergebnisse zur Wiedererkennung von Orten in sich verändernden Umgebungen 189

8 **Zusammenfassung** 205
8.1 Ergebnisse und Beiträge 205
8.2 Offene Fragen im Bereich Superpixel 208
8.3 Offene Fragen im Bereich der Wiederkennung von Orten in sich verändernden Umgebungen 210

Kurzzusammenfassung (Übersetzung)

Superpixel sind das Ergebnis einer Übersegmentierung von Bildern. Sie sind eine bewährte Bildrepräsentation und werden beispielsweise für Objekterkennung, 3d Rekonstruktion und semantische Segmentierung eingesetzt. Es existieren vielfältige Ansätze solche Segmentierungen zu erzeugen, allerdings fehlen Erkenntnisse über deren Eigenschaften. Insbesondere sind in der Literatur widersprüchliche Ergebnisse zu finden. In dieser Arbeit werden Qualität, Stabilität, Kompaktheit und Laufzeit als wichtige Eigenschaften identifiziert. Während für manche dieser Eigenschaften bewährte Methoden zur Auswertung existieren, ist das für Stabilität und Kompaktheit nicht der Fall. In dieser Arbeit werden zwei neue Metriken vorgestellt, die die beiden genannten Eigenschaften basierend auf Informationen über den Optischen Fluss auswerten. Diese und andere neue und existierende Maße werden verwendet, um einen standardisierten Benchmark für Superpixel Algorithmen bereit zu stellen. Die Ergebnisse eines umfangreichen Vergleiches existierender Algorithmen mit diesem Benchmark motivieren zwei neue Algorithmen, die die Eigenschaften existierender Algorithmen besser ausbalancieren: Preemptive SLIC erweitert den existierenden SLIC Algorithmus mit einem lokalen Abbruchkriterium und ermöglicht damit eine Beschleunigung um Faktor 5. Compact Watershed kombiniert Seeded Watershed mit Kompaktheitsbedingungen, um regelmäßige, kompakte Superpixel Segmentierungen mit der gleichen hohen Geschwindigkeit des nativen Watershed Algorithmus zu erzeugen.

Der Betrieb von autonomen Systemen mit visueller Navigation über die Dauer von Tagen, Wochen oder Monaten erfordert die Wiedererkennung von Orten trotz schwerwiegender Veränderungen durch wechselnde Beleuchtung, Tag-Nacht Zyklen, Wettereinflüsse oder verschiedene Jahreszeiten. Der zweite Teil dieser Arbeit stellt zwei neue Verfahren vor, um Superpixel für die Wiederkennung von Orten unter sich verändernden Bedingungen zu nutzen. Der erste Ansatz ist das Lernen von systematischen Veränderungen. Statt beispielsweise die Bilder aus Sommer und Winter direkt zu vergleichen, wird ein zusätzlicher Vorhersageschritt vorgeschlagen. Basierend auf Superpixel Vokabularen wird ein vorhergesagtes Bild erzeugt, wie die Sommerszene im Winter aussehen könnte oder umgekehrt. Die erzielten Ergebnisse zeigen, dass unter bestimmten Anforderungen an die Veränderungen und die Trainingsdaten, existierende holistische Vergleichsverfahren von dem zusätzlichen Vorhersageschritt profitieren können.

Es ist bekannt, dass holistische Vergleichsverfahren im Falle von Blickwinkeländerungen scheitern. Daher wird in dieser Arbeit ein weiteres Verfahren basierend auf lokalen Bildmerkmalen und Star-Hough vorgeschlagen. Star-Hough ist ein neuartiger Ansatz, die Anordnung der Landmarken im Bild in die Berechnung der Ähnlichkeit zwischen Bildern einzubeziehen. Er basiert auf Star Graph Models und Hough Voting und ist besonders geeignet für unpräzise lokalisierte Merkmale und hohe Ausreißerraten. Die neuartigen Landmarken basieren auf der Kombination von lokalen Merkmalsdetektoren und Deskriptoren aus Convolutional Neural Networks. In dieser Arbeit werden verschiedene Ansätze vorgestellt und ausgewertet, um Superpixel in die Merkmalsdetektion einzubeziehen. Das vorgeschlagene Verfahren kann mit verschiedenen Merkmalsdetektoren verwendet werden. Insbesondere die Kombination mit Merkmalen basierend auf einem Multiscale Superpixel Grid liefert bessere Ergebnisse als die existierenden Methoden und erscheint als vielversprechende Ausgangsbasis für praktische Anwendungen.