

Filtering of segmentation hierarchies for improved region-to-region matching

by

Oliver Walzer

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.Sc. degree in
Computer Science

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

© Oliver Walzer, Ottawa, Canada, 2011

Abstract

The representation and manipulation of visual content in a computer vision system requires a suitable abstraction of raw visual content such as pixels in an image. In this thesis, we study region-based feature representations and in particular, hierarchical segmentations because they do make no assumptions about region granularity. Hierarchical segmentations create a large feature space that increases the cost of subsequent processing in computer vision systems.

We introduce a segment filter to reduce the feature space of hierarchical segmentations by identifying unique regions in the images. The filter uses appearance-based properties of the regions and the structure of the segmentation for the selection of a small set of descriptive regions. The filter works in two phases: selection with a criteria based on relative region size and a sorting based on a variational criteria. The filter is applicable to any hierarchical segmentation algorithm, in particular to bottom-up and region growing approaches. We evaluate the filter’s performance against an extensive set of ground-truth regions from a dataset containing image sequences with scenes of different complexity.

We demonstrate a novel region-to-region image matching approach as a possible application of our segment filter. A reduced segmentation tree is reconstructed based on the set of regions provided by the filtering. The reduction of the feature space by the segment filter simplifies our region-to-region matching approach. The correspondences between regions from two different images is established by a similarity measure. We use a modified mutual information measurement to compute the similarity of regions. The identified region correspondences are refined using the reduced segmentation tree. Our region-to-region matching approach is evaluated with an extensive set of ground-truth correspondences. This evaluation shows the large potential of both, our filtering and our matching approach.

Acknowledgements

I would like to thank my thesis supervisor Dr. Jochen Lang for his support throughout my research endeavours. His expertise, patience and resources made this thesis possible. I also like to thank my family and friends for their support.

Contents

1	Introduction	1
1.1	Region-to-Region Correspondence	4
1.2	Thesis Statement	5
1.2.1	Major Contributions	5
1.3	System Overview	5
2	Related Work	7
2.1	Building Image Signatures	8
2.1.1	Feature Detectors	8
2.1.2	Segmentation	11
2.2	Image Similarity	17
2.2.1	Hierarchical	20
2.3	Summary	21
3	Descriptive Segment Filter	22
3.1	Definition of Descriptive Segments	23
3.2	Segment Properties	23
3.2.1	Cost	25
3.3	Queue-based Approach	27
3.3.1	Cost Analysis	27
3.3.2	Outlier Detection	29
3.3.3	Choosing the Segment	33
3.4	Two-phase Approach	34
3.4.1	Selection	34
3.4.2	Sorting	37
3.5	Results	39
3.5.1	Experimental Dataset	40

3.5.2	Descriptive Segment Selection	41
3.5.3	Evaluation Syntax	43
3.5.4	Two-phase Approach Criteria	44
3.5.5	Performance Evaluation	47
3.5.6	Discussion	52
4	Matching Descriptive Segments	58
4.1	Reduced Hierarchy Construction	58
4.2	Establish Segment Correspondence	59
4.2.1	Refine Correspondences	60
4.3	Segment Correspondence Measures	60
4.3.1	Cost	61
4.3.2	Mutual Information	61
4.4	Score Matches	67
4.4.1	Homography Computation	69
4.5	Results	69
4.5.1	Planar Images	70
4.5.2	Images with a Dominant Plane	71
4.5.3	Images with a Small Shift	73
4.5.4	Evaluation	74
4.5.5	Analysis by Homography	77
4.5.6	Visual Conformance	78
4.5.7	Refinement Influence	79
4.5.8	Summary	79
4.6	Discussion	82
5	Conclusion	85
5.1	Future Work	86
A	Dataset	88

List of Tables

3.1	Evaluation of the threshold for outlier detection by deviation	31
3.2	Q Test thresholds for different confidence levels and sample sizes.	32
3.3	Evaluation of Q Test confidence levels	34
3.4	Overview of the experimental dataset	40
3.5	Results of the filtering approaches using the standard algorithm	49
3.6	Results of the filtering approaches using the alternative algorithm	51
4.1	Homography of planar images	71
4.2	Homography of images with a dominant plane	73
4.3	Translation of images with a small shift	74
4.4	Results of the matching approach	77

List of Figures

1.1	The high level architecture of our system	6
3.1	Reconstructed sequence of merge costs	28
3.2	Merge costs for a colour grid image	29
3.3	Merge costs of descriptive segments in a merge sequence	30
3.4	Example of deviation based outlier detection	32
3.5	Example of outlier detection based on Chauvenet’s criterion	33
3.6	Example of Q Test outlier detection	35
3.7	Example of selection by size	37
3.8	Example of sorting by a combined criteria	39
3.9	One image per sequence in the experimental dataset	41
3.10	Browsing through the merge history	43
3.11	Example of descriptive segment selection	44
3.12	Difference between desired and actual selection	45
3.13	Results of the selection phase of the two-phase approach	46
3.14	Results of the sort phase of the two-phase approach	48
3.15	Filter ratio for the different approaches	51
3.16	Descriptive segments identified for each approach	55
3.17	Visualized filtering results of an example image	56
3.18	Best two-phase filtering results of every sequence	57
4.1	Visualization of mutual information and its entropies	62
4.2	Mutual information for two rectangles	64
4.3	Example of matching by mutual information	66
4.4	Stripe image for mutual information evaluation	67
4.5	Mutual information as segment correspondence measure	68
4.6	Evaluation of the correct matches based on the computed homography	78

4.7	Visual evaluation of correct matches	79
4.8	Reduced number of matches by refinement	80
4.9	The best correspondence for two images per sequence	82
A.1	The backyard image sequence.	88
A.2	The bikes image sequence.	89
A.3	The boesch image sequence.	90
A.4	The graffiti image sequence.	91
A.5	The home image sequence.	92
A.6	The house image sequence.	93
A.7	The Leuven image sequence.	94
A.8	Segmentation algorithm comparison using backyard1 image.	95
A.9	Segmentation algorithm comparison using bikes1 image.	95
A.10	Segmentation algorithm comparison using boesch2 image.	96
A.11	Segmentation algorithm comparison using graffiti1 image.	96
A.12	Segmentation algorithm comparison using home3 image.	96
A.13	Segmentation algorithm comparison using house2 image.	97
A.14	Segmentation algorithm comparison using Leuven1 image.	97

Glossary of Terms

2D Two-Dimensional.

3D Three-Dimensional.

BSDS Berkley Segmentation Dataset.

CBIR Content Based Image Retrieval system.

CST Connected Segmentation Tree.

DAG Directed Acyclic Graph.

DP Dynamic Programming.

DOF Depth Of Field.

DoG Difference of Gaussian.

EM Expectation Maximization.

GLOH Gradient Location-Orientation Histogram.

IRM Integrated Region Matching.

MI Mutual Information.

MRI Magnetic Resonance Imaging.

MSER Maximally Stable Extremal Region.

MSHP Most Similar Highest Priority.

OCR Optical Character Recognition.

PCA Principal Components Analysis.

RANSAC RANdom SAmple Consensus.

RIFT Rotational Invariant Feature Transform.

SIFT Scale Invariant Feature Transform.

SLIC Simple Linear Iterative Clustering.

SRM Statistical Region Merging.

SURF Speeded Up Robust Features.

UCM Ultrametric Contour Map.

Chapter 1

Introduction

Visual sensing is one of the most fascinating abilities of the human organism. In a fraction of a second visual information is processed in our brain without further thinking. Even very complex situations are understood almost instantaneously which allows us to take appropriate actions if necessary. Depending on previous knowledge we are able to classify “objects” based on their appearance even if we have never seen this particular one before. Our understanding of the environment enables us to reach for objects regardless of obstacles in the way. Because of the automatism involved we do not realize the complexity of such a process.

While visual perception is an easily performed task for humans and animals it is extremely demanding for machines. In computer science the field of computer vision covers the efforts of creating visually aware systems. Over the past two decades an uncountable number of such systems evolved from research in this field. It received a large amount of attention due to its broad spectrum of applications. As a matter of its complexity, however, solutions are very specific to the problem they solve. According to Datta [23], “many years of research have made it clear that emulating human vision is very challenging, but that nonetheless, practical approaches can help to build useful systems”. The focus on a specific task lets us assign most of the available computer vision system into one of the following two categories.

Object recognition is the process of identifying and analyzing objects in visual content. A traditional discipline of object recognition is optical character recognition (OCR) which found its way into standard office software. More recently content based image retrieval systems (CBIR) evolved that allow the user to query a database of images. Such a system is queried either by a query image or by

keywords. A similarity measure between images is sufficient in case of a query by image. No understanding of the objects themselves is necessary. For query by keyword a classification of the objects in the images is required. This classification lets the system understand the content of the images and improves the results. Another application of object recognition is pose estimation. The pose, for instance, is significant when magnetic resonance images (MRI) are processed by a computer vision system. The processing of medical images in a computer vision system is of great interest and a lot of effort has been put on the development of such systems. Applications range from detection of anomalies due to diseases to augmented reality supported surgery. In surveillance, computer vision approaches are applied to count or even identify people.

Scene reconstruction leads to an understanding of the environment. Industrial robots were among the first environment aware machines. Such robots have their specific purpose and are very limited in their understanding of the environment. Typical computer vision applications of industrial robots include point tracking or camera motion estimation. More sophisticated robots usually use stereo vision to become aware of their environment. With stereo vision the relative position of objects in 3D space can be computed. Such technology is of particular interest to the military or in outer space exploration. When there is no stereo vision available a dense 3D model can be computed from images showing the scene from different viewpoints. In photo tourism [77] this approach is used to bring the images of popular places into context and embed them at the position they were taken from. Ultimately, a reconstructed scene can be populated with further information about the content of it. The applications of such augmented reality systems are almost unlimited. Consumer available systems, for instance, were lately introduced to enhance gaming experience by including the environment of the player into a game [78].

In the early days of computer vision visual sensors, such as cameras, were bulky and expensive. The availability of inexpensive handheld consumer cameras simplified the data acquisition for system that process natural images. The currently available cameras are able to capture any scene with enough detail. Furthermore, images are largely available on the Internet, e.g., on photo sharing platforms like Flickr. Such websites contain many thousands of images from an uncountable number of places in the world. Special sensors are only required when non-optical information is to be captured, for instance, magnetic resonance image in medical imaging.

Images and videos are the common machine understandable representation of visual content. Their low level unit is the pixel which is not a natural entity. Pixels are a consequence of the discrete sampling of visual content [70]. They are not a suitable representation of visual content because they poorly correspond to high level objects and their number is very high even for low resolution of the content. Therefore, a transformation of the pixels to a higher level representation is required in computer vision system. This reduces the unmanageable large feature space and puts some semantics on the content. It does, however, introduce the so called “sensory gap” because it summarizes the content and makes some assumptions about its importance. This is described by Datta [23] as “the gap between the object in the world and the information in a (computational) description derived from a recording of that scene”.

The high level representation most suitable for a computer vision system depends on its task as it must contain the information required to solve the problem it is built for. A large number of approaches are available in order to extract raw image data. Among them are feature detection methods like the scale invariant feature transform (SIFT) descriptor [54, 55]. It analyzes local invariants to build a set of features that can, for instance, be used for object recognition or image matching. It is one of the most popular feature detectors as seen by the modifications presented by various authors [43, 64, 51]. Other popular feature detectors include the speeded up robust features (SURF) [10] or the maximally stable extremal region (MSER) [62].

Beside feature detectors segmentations are a wide spread high level representations of images. A segmentation is a clustering of the visual content based on the raw image data. A perfect clustering will group all pixels that belong to the same object together. Due to the sensory gap such a clustering is not feasible. Already small changes in the appearance of an object may change the association of pixels to the segments. For this reason a segmentation is never perfect, but as Arbelaez et al. [6] outlined “many visual tasks can benefit from the reduction in complexity achieved by transforming an image with millions of pixels into a few hundred or thousand superpixels”. There are at least two types of segmentations: flat and hierarchical. While in flat segmentations there exists only one layer of segments in hierarchical segmentations the segments are arranged in a tree. In such a tree a segment is composed of its children, which are either composed of their children or are pixels. Hierarchical segmentations have the advantage over flat segmentations to provide segments for different levels of details. Their major downside is the much larger feature space of hierarchical segmentations. However, they do not assume any granularity for the segmentation like flat segmentation algorithms do. Our

particular interest in this thesis are hierarchical segmentations that create a binary tree by splitting their segments into exactly two children.

1.1 Region-to-Region Correspondence

The identification of corresponding points between images is an essential task many computer vision systems perform. In photo tourism, for instance, a 3D model of a scene is computed based on the photographs of popular objects [77]. The scene reconstruction uses the corresponding points in the images to estimate their point and field of view. The more images brought into correspondence the more accurate the reconstructed model becomes. This is in particular useful for popular places, where there are thousands of images available. In such systems corresponding points are often established using feature points identified by one of the various feature detectors. These detect local invariants, like corner points for example, which are unique to the images and are relatively stable when transformed. With an iterative approach, such as RANSAC by Fischler and Bolles [30], an initial correspondence of feature points is optimized to find the transformation backed up by most feature points.

The correspondence of pixels can also be established from the correspondence of regions. Regions contain many points and encode more information than a feature point does. These information include colour, texture and shape. Additionally, a hierarchical segmentation provides information about the structure of regions. This and the other properties of regions were successfully used in various systems as a similarity measure between images [21, 34, 35, 44, 52, 67, 81, 82, 83]. Among them are content-based image retrieval systems, applications for image matching and approaches to image classification.

The information about the structure of regions is, besides the independence on the level-of-detail, the major advantage of a hierarchical segmentation. In image matching, for instance, this information can be used in addition to the other properties of a region to find the best correspondence. The price of this hierarchical information is its large feature space caused by the encapsulation of the regions. This feature space is often reduced by building the hierarchical segmentation from super-pixels instead of raw pixels. Such super-pixels are constructed by an over-segmentation using a flat segmentation algorithm. Even if this lowers the number of segments available, the feature space stays relatively large. Our goal is to reduce this complexity by choosing a small set of segments that represent the image. This simplifies further tasks as we will show with our region-to-region matching algorithm.

1.2 Thesis Statement

Most of the regions contained in a hierarchical segmentation are not of interest to computer vision algorithms, e.g., image matching. With careful evaluation, the most important regions can be identified and extracted in order to reduce the complexity of these algorithms. This has the potential to greatly simplify processing without losing accuracy.

1.2.1 Major Contributions

We have developed a system that finds corresponding regions in two images. The system does not require deeper knowledge of the image domain and is parameter free except for the number of correspondences desired. It is generally independent from a specific segmentation algorithm as long as the segmentation is hierarchical. A small set of regions is extracted from the segmentations of the images. The visual and structural properties of the regions are used to determine which of them are essential to the images. We call these regions “descriptive segments” because they are the regions in the hierarchical segmentation that describe the images. Our system computes the best correspondence for each of these descriptive segments. Due to the reduced feature space this matching is simple yet powerful. In summary, the key contributions of our work are:

- The definition of descriptive segments,
- a filter that extracts these segments from a hierarchical segmentation, and
- a region-to-region matching algorithm based on a correspondence measure.

1.3 System Overview

Figure 1.1 contains the high level architecture of our system. The input to our system are two images of the same scene from different viewpoints or with other varying conditions, such as brightness or sharpness. For both images our system builds a hierarchical segmentation. This is done using a flat segmentation algorithm to over-segment the images and then a hierarchical algorithm to construct the encapsulation of the regions. There are two different combinations of flat and hierarchical segmentation algorithms implemented in our system, both based on publicly available algorithms. The segmentation tree is the input to the two-phases filtering approach. In a selection phase potentially descriptive segments are identified. In the later sorting phase these segments are sorted

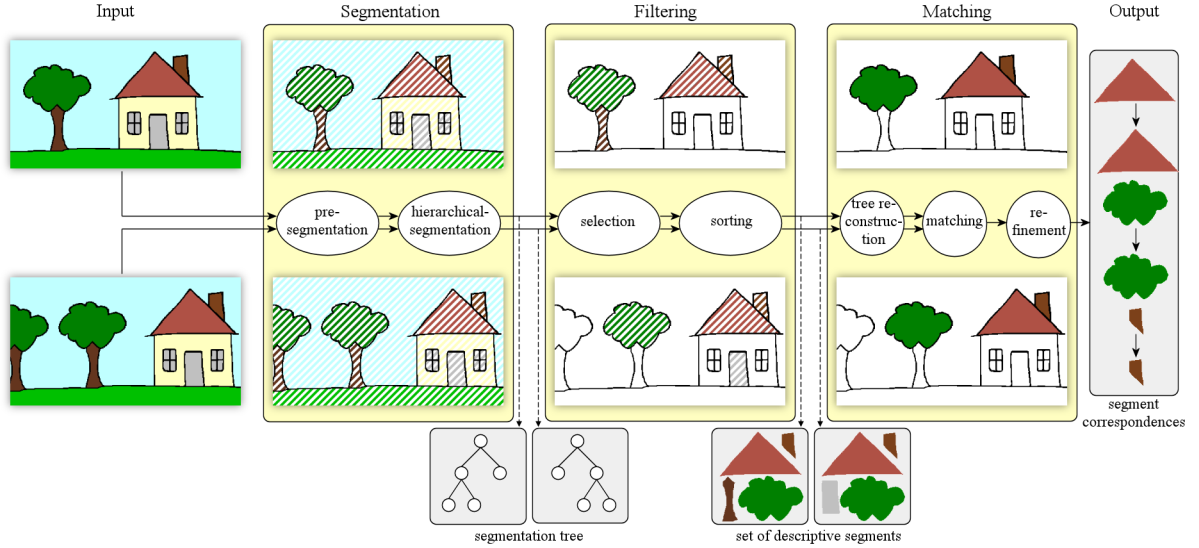


Figure 1.1: The high level architecture of our system.

in order of their relevance to the image. At the end of the filtering stage a set of descriptive segments is available. These segments contain the unique features of the images and were found to be the best representation of them. In the matching stage a reduced tree is assembled with the descriptive segments. A correspondence measure is used to find the best matches between the sets of descriptive segments. These matches are validated against their structure and obviously wrong matches are removed. After this refinement a list of matches agreeing with the semantics of both images is available. This is the result of our system.

Related work in the field of feature detection and similarity measures is given in Chapter 2. In Chapter 3 our approach to descriptive segment filtering is presented. Different approaches are compared to each other and the most successful one is tested against an extensive dataset. Chapter 4 describes our region-to-region matching approach that uses the descriptive segments provided by our filter. We evaluate our approach with different segment correspondence measures in order to find the one most suitable for our purpose. We conclude our work with Chapter 5 where we will summarize the results and provide future directions of this research.

Chapter 2

Related Work

The following sections will introduce the key contributions related to our work. The related fields include object recognition, image matching, content based image retrieval (CBIR), stereo vision and scene reconstruction.

Extracting information from images remains the major problem in any task of computer vision [23]. The fact that machines are not even close to being able to recognize a scene as well as humans is due to the sensory and the semantic gap. The sensory gap describes the difference between the scene and the information available in its computational description. Constructing an automatic comprehensible description of scenes is a very challenging problem because the underlying representation of a scene poorly corresponds to its content. We discuss different approaches to image description in Section 2.1. The difference between human and automatic perception is called the semantic gap. Even with an automatic recognizable description of a scene, the meaning of the visual data is hard to classify. Additionally the meaning of a scene can change due to prior knowledge that may not be available. For this matter even humans classify the same scene differently. Another example can be found in text analysis, where the same words can have completely different meanings. A particular application where the semantic gap matters is in image comparison. In Section 2.2 we will discuss approaches to image matching.

Over the last years, content-based image retrieval with its many related fields such as computer vision, database systems, statistics, information retrieval and machine learning, has received a lot of attention in research ranging from specific topics to whole systems [91, 80, 52, 46, 42, 19, 16, 90]. The purpose of a content-based image retrieval system is the organization of images based on their visual content. Such a system can be queried

- by association, where the result is refined iteratively by browsing through the images,
- by aim, where a particular image is queried, or
- by category, where images of a specific semantic class are requested.

These search patterns do not require a content-based image retrieval system to have an accurate description of the images [76]. It is sufficient for the system to present similar images according to the user's query. This is often accomplished by comparing the query image to the ones available in a database. The success of these systems heavily depends on a good similarity measure between images.

2.1 Building Image Signatures

The visual data of real world digital images is represented by pixels. Analyzing images only based on their pixel values does not suffice for accurate processing. An automatic-understandable description of the presented scene is required. Feature detectors and segmentation algorithms extract such a signature from the image. They construct a high level output of objects or salient points. This allows the automatic processing of the presented scene and is a necessary pre-processing step in applications, such as content-based image retrieval.

2.1.1 Feature Detectors

Feature detectors have widely been used in various applications of computer vision. They consist of a feature detection and a feature description step. Depending upon the algorithm the steps can either be combined or separated. The purpose of the feature detection step is to capture the visual properties of an image such that they can be summarized by a vector of features in the feature description step. There are various approaches and the one used depends on the particular requirements of the application. Popular techniques for feature detection use colour, texture, shape or salient points for detection. Extraction of features is done locally or globally. In global feature extraction the image is analyzed as a whole, for instance with a histogram. This analysis shares a speed advantage over local feature extraction. However these algorithms are often too rigid to reveal important image properties and the extracted features are oversensitive to their location. For this reason global extraction fails to identify visual characteristics of an

image and is avoided in modern algorithms [23]. In local feature extraction information is extracted by comparing single or blocks of pixels with their neighbours. This relationship between pixels and their neighbourhood identifies a feature more reliable than global extraction. In most applications of feature detection this is a far more important property than the greater speed achieved by global extraction [76].

Stereo matching and object recognition has been the traditional domain for feature detectors based on salient points (e.g. corners). These points of interest are detected by analyzing local invariants of the image. The gathered set of points is invariant to affine transformations and illumination changes [23]. In object recognition the Scale Invariant Feature Transform (SIFT) has outperformed others due to its stable results even on partially occluded images [54, 55]. In an image, feature points are identified by the minimums and maximums of the Difference of Gaussian (DoG) function, defined as

$$DoG(x, \sigma_1, \sigma_2) = g(x, \sigma_1) - g(x, \sigma_2) \quad , \quad (2.1)$$

whereas

$$g(x, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}} \quad (2.2)$$

is the Gaussian smoothing function computed in the vertical and horizontal directions of the image. The input image I is smoothed twice using $g(x, \sigma)$ to compute the DoG function. Calling the result of the first smoothing step with $\sigma = \sqrt{2}$ image A and the second smoothing step with $\sigma = 2$ image B , then

$$D = A - B \quad (2.3)$$

is the difference of Gaussian. For efficient detection of the SIFT keys, image D is arranged in a pyramid at different scales. The pyramid is constructed bottom up by taking image B and re-sampling it using bilinear interpolation. A pixel spacing of 1.5 pixels is chosen, meaning four adjacent pixels are linearly combined into one pixel for the next higher pyramid level. The difference of Gaussian is then computed for the re-sampled image. For a pixel to become a SIFT key it has to be a stable minimum or maximum across its eight neighbours and the pixels in the 3×3 neighbourhood at the next higher resolution pyramid level. The test is repeated for the coarser pyramid level. Most potential keys can be eliminated by just a few comparison keeping the cost of the algorithm low. The presence of an object in an image can then be detected by a comparison of SIFT keys with keys from a previously set up database using a best-bin-first search. SIFT has been extended by methods like phased-based local features [15], principal components analysis (PCA) SIFT [43], gradient location-orientation histogram (GLOH) [64]

and rotation-invariant feature transform (RIFT) [51]. While RIFT extends SIFT such that is fully rotational invariant the other methods aim to improve SIFT's robustness against illumination change, rotation and translation. PCA-SIFT and GLOH furthermore compact the features and make them more distinctive for comparison. Recently, Koniusz et al. [48] proposed a method in which image segments are used as bins for SIFT features. This method has shown to produce comparable results with approximately eight times less data than traditional methods. Reducing the number of salient points by a segmentation has also successfully been used by Zhang et al. [95]. Speeded Up Robust Features (SURF) introduce another feature detection and comparison algorithm [10]. It uses an integral image [87] in the detection and the description step of the algorithm, which reduces computation time. The computation of the matching step in SURF benefits from less feature vectors with lower dimension than SIFT. This makes SURF much more efficient than SIFT, at the cost of small losses in the result quality [9].

In stereo matching the correspondence of points in two images from different viewpoints is to be established. A maximally stable extremal region (MSER) is a region in an image that exceeds a certain intensity threshold [62]. These regions are computed by sorting the pixels of an image by their intensity. Putting the pixels back on the images in increasing or decreasing order creates growing regions. By maintaining a list of connected segments, an efficient union find algorithm constructs the final maximally stable extremal regions. By definition a region Q is continuous subset of D where $I : D \subset \mathbb{Z}^2 \rightarrow S$ and

- S is totally ordered.
- A neighbourhood relation is defined for $A \subset D \times D$.

An extremal region $Q \subset D$ is a region such that for all $p \in Q$, $q \in \delta Q$: $I(p) > I(q)$ or $I(p) < I(q)$, where $\delta Q = \{q \in D \setminus Q : \exists p \in Q : qAp\}$ is the region boundary. An extremal region Q_i is maximally stable if and only if $q(i) = \frac{|Q_{i+\Delta} \setminus Q_{i-\Delta}|}{|Q_i|}$, with $\Delta \in S$ being a parameter of the method, has a local minimum at i . These maximally stable extremal regions are invariant to affine image intensity transformation and therefore can be detected in stereo images. The correspondence between the images can then be computed by matching the set of extremal regions to find the affine transformation that describes the viewpoint change between the images. This approach has proven to be stable for stereo matching purposes. An improvement to MSER was proposed by Forssen [31]. His use of a colour based descriptor has shown to outperform the original grayscale descriptor while still being faster than other comparable algorithms. An interesting combination of MSER as a blob based detector with SIFT as a salient point detector

has been proposed by Forssen and Lowe [32]. While using the advantage of MSER the SIFT based descriptor encodes the shape of the regions. This reduces the sensitivity to small shape deformations making the approach more robust to illumination changes and partial occlusions.

With the increasing diversity of feature detectors one may not easily be able to chose the most suitable solution for a particular application. An excellent comparison of the previously discussed feature detectors is provided in Mikolajczyk et al. [65]. A performance evaluation of local descriptors like SIFT is done in Mikolajczyk and Schmid [64]. A detailed comparison of SURF and SIFT implementations can be found in Bauer et al. [9].

Shape is not only a key in segmentation based image signatures, it can also be used as an independent feature. The Chamfer distance [71] has traditionally been used for shape description and matching, for instance by Barrow et al. [8] and Borgefors [13]. More recently Berretti et al. [12] used local shape descriptors to represent tokens in a metric tree for efficient matching. Petrakis et al. [68] approximated shapes as sequences of concave and convex segments to match them in a dynamic programming (DP) approach.

Less relevant to our domain are feature detectors based on texture. These are very closely linked to the underlying problem they aim to solve. Texture detectors have successfully been applied to match aerial images or detect abnormalities in medical images. Markovian analysis [38] and wavelet transforms [85, 26, 52] have been used for feature extraction. The latter became popular due to their locality and compression efficiency [76]. An approach mixing salient points and wavelet based texture detection was introduced by Tian et al. [80].

2.1.2 Segmentation

A segmentation is a partitioning of the image into multiple regions. This transform of millions of pixels into a few super-pixels or regions reduces the complexity in further image analysis. It is crucial for an algorithm to encode the important properties of the image within the segmentation. A loss of information can lead to misinterpretation of the image in a later analysis. Since a segmentation of an image encodes the shapes of the regions, it is especially useful for object detection or shape analysis. It is desirable for a segment to represent objects as a whole in the image. Because objects may have poor boundaries and pixels are not related at a semantic level, this remains a challenging problem.

Even if the goal of image segmentation seems to be obvious it is hard to introduce a segmentation quality measure. This is due to the fact that for an image there is often more than one acceptable solution. An algorithm that produces good results for one image may not necessarily be accurate enough for another. The reliability of segmentation algorithms is, besides their computational complexity, among the major issues in segmentation. However, good segmentation, as a region based image signature, is considered a major step towards image understanding. Therefore, segmentation is an important field in computer vision [6, 23]. Segmentation algorithms can be characterized by the datastructure that they produce. We will first introduce planar algorithms in the following section and then talk about region merging approaches that construct hierarchical segmentation trees.

Planar

One of the possible approaches to image segmentation is to form segments by clustering the pixels. The k-means algorithm aims to partition a set of n datapoints into k clusters such that the datapoints belong to the closest cluster centre [53]. The algorithm starts by randomly choosing k initial cluster centres and assigning the datapoints to their closest cluster centre. Close is in terms of a distance that can include colour or luminance. The cluster centres are reassigned with the datapoints belonging to them and then, that algorithm re-iterates. This process is completed as soon as no data point changes its assigned cluster. Even though k-means clustering is NP-hard [58], the algorithm is popular due to its simplicity and observed speed [7].

Another clustering algorithm is mean shift which, in its generalization, has k-means-like algorithms as special cases [33, 20]. Given a planar Kernel function with a threshold λ

$$K(x) = \begin{cases} 1, & \text{if } \|x\| \leq \lambda \\ 0, & \text{if } \|x\| > \lambda \end{cases}, \quad (2.4)$$

or a Gaussian Kernel function with a constant c

$$K(x) = e^{c\|x\|^2}, \quad (2.5)$$

mean shift locates the maximum of the density function

$$m(x) = \frac{\sum_{s \in S} K(s - x)s}{\sum_{s \in S} K(s - x)} \quad (2.6)$$

by iteratively adjusting the neighbourhood $s \in S$ of x such that $s \leftarrow m(s)$. The algorithm completes when $m(s)$ converges to s . Mean shift and k-means like algorithms potentially over-segment the image. While this is not desirable in most cases and can be avoided by careful choice of the number of segments, it is desirable to construct super-pixels. Super-pixels can be the finest structure for another segmentation algorithm. They group several low level pixels together in order to reduce the computational complexity of a successive segmentation algorithm. Lately, planar segmentation algorithms that construct reliable super-pixels have evolved, e.g., SLIC Superpixels by Achanta et al. [1].

Assuming that pixels are vertices connected by weighted edges, building a segmentation becomes a weighted graph partitioning problem. This approach to image segmentation is taken in normalized cuts [74]. Edge weights are based on the difference in brightness and represent the likelihood that the pixels belong to the same segment. Malik et al. extended this property to include cues of contours and texture differences in edge weights. Yu and Shi [94] introduced a variation of the normalized cuts algorithm where a partial grouping is known. This leads to a constrained optimization problem which has applications in pathological images where the location and size of organs is known. Another graph-based segmentation algorithm was introduced by Felzenszwalb and Huttenlocher [29]. Besides the intensity difference along an edge of the graph this algorithm introduces a quantity that measures the intensity differences of the pixels within a segment. This greatly improves the algorithm's awareness of fine structures while not over-segmenting coarse regions.

The classification of foreground and background objects in images is a related problem which can be solved using segmentation. In Wang et al. [89] a multi-resolution segmentation algorithm based on wavelet coefficients is used for this classification in low depth of field (DOF) images. In such images the foreground object is particularly focused while the background remains blurry. Among other methods, Deng and Manjunath [25] use an unsupervised approach that segments an image into homogeneous colour and texture regions. Texture and colour has also been used by Chen et al. [18] in a sequential approach where colour is used to refine a segmentation by texture.

Hierarchical

Without prior knowledge it is hard for a segmentation algorithm to determine the correct level-of-detail. While Felzenszwalb and Huttenlocher [29] address this issue in their segmentation algorithm, there have been many approaches to hierarchical segmentation. These algorithms organize the segments in a hierarchy to reflect the containment of a

segment in another. This structure forms an irregular pyramid of segments which is equal to an object tree in the best case.

Algorithms based on region merging, region growing and split-and-merge are among the popular approaches to hierarchical segmentation [56]. While region merging and region growing are among the bottom-up approaches split-and-merge refines a single segment in a top down method. In this work we focus on region merging, in particular on the algorithm introduced by Beaulieu and Goldberg [11].

Given a set of initial segments $P_0 = \{S_1, S_2, \dots, S_n\}$, let B_i be the neighbourhood of S_i and $C_{i,j} = C(S_i, S_j)$ be the cost of merging S_i with S_j . The algorithm is initialized with

$$k=0 \tag{2.7}$$

$$CS = \{C_{i,j} \mid S_j \in B_i \text{ and } i > j\} \quad . \tag{2.8}$$

Iteratively merge the two most similar segments

$$k=k+1 \tag{2.9}$$

$$C_{u,v} = \min_{C_{i,j} \in CS} (C_{i,j}) \tag{2.10}$$

$$P^k = (P^{k-1} \cup \{S_k\}) \cap \overline{\{S_u, S_v\}} \tag{2.11}$$

$$B_m = (B_u \cup B_v) \cap \overline{\{S_u, S_v\}} \tag{2.12}$$

$$\forall S_j \in B_m, B_j = (B_j \cup \{S_k\}) \cap \overline{\{S_u, S_v\}} \tag{2.13}$$

$$CS = (CS \cup \{C_{k,j} \mid S_j \in B_k\}) \cap \overline{\{C_{i,j} \mid i, j = u \text{ or } v\}} \tag{2.14}$$

and stop when there are no more merges possible. An over-segmented input is required for the initial set of segments P_0 . While taking the image pixels as input is possible it was shown that a pre-segmentation of the image, for instance with the watershed segmentation [39], can yield better results. The segmentation outcome mainly depends upon the cost function (merge predicate) $C(S_i, S_j)$. This function is a dissimilarity measure that reflects the probability that two adjacent regions do not belong to the same object. Searching for the minimum dissimilarity choses the most similar segments as merge candidates. This algorithm has been criticized for its local construction by only comparing two segments at once [56]. However due to its simplicity, the interchangeability of the merge predicate and its history, it is widely used. Different merge predicates that provide a good image partition have been studied. Among them is Ward's criterion [92] that defines the cost function as the amount of error that is introduced by merging the two

segments. In Brox et al. [14] the intensity difference along the border is used as similarity measure. A predicate based on the Minimum Description Length principle has been presented by Haris et al. [39]. A probabilistic predicate based on the size, luminance, chrominance and shape of the segments was introduced by MacDonald [57]. Let q be the distribution of all splits of S into L and R . This distribution is defined as a combination of the four individual distributions

$$\begin{aligned} q(S, L, R) := & q_{\text{size}}(S, L, R, \omega_{\text{size}}) \\ & \times q_{\text{luminance}}(S, L, R, \omega_{\text{luminance}}) \\ & \times q_{\text{colour}}(S, L, R, \omega_{\text{colour}}) \\ & \times q_{\text{shape}}(S, L, R, \omega_{\text{shape}}) \end{aligned} \quad (2.15)$$

where the parameters $\omega_{\text{size}}, \omega_{\text{luminance}}, \omega_{\text{colour}}, \omega_{\text{shape}}$ are pre-determined constants. Each of the distributions model a particular property of the segments. The visual characteristics of segments are its size in pixels, its mean luminance and chrominance (in LUV colour space) across its pixels and its boundary length to its neighbour. A beta distribution

$$B(x, \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1} \quad (2.16)$$

with $B(\alpha, \beta)$ being the normalizing beta function is used to model the size distribution

$$q_{\text{size}}(S, L, R, \omega_{\text{size}}) = c_{\text{size}} \left(\frac{L_{\text{size}}}{S_{\text{size}}} \right)^{\omega_{\text{size}}-1} \left(\frac{R_{\text{size}}}{S_{\text{size}}} \right)^{\omega_{\text{size}}-1}. \quad (2.17)$$

where c_{size} is the normalizing constant and $S_{\text{size}} = L_{\text{size}} + R_{\text{size}}$. For both, luminance and chrominance, a Gaussian distribution

$$G(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\mu-x)^2}{2\sigma^2}} \quad (2.18)$$

is used to model the probability functions

$$q_{\text{luminance}}(S, L, R, \omega_{\text{luminance}}) = c_{\text{luminance}} e^{-\frac{(L_{\text{luminance}} - S_{\text{luminance}})^2 L_{\text{size}} + (R_{\text{luminance}} - S_{\text{luminance}})^2 R_{\text{size}}}{2 S_{\text{size}} \omega_{\text{luminance}}^2}} \quad (2.19)$$

$$q_{\text{colour}}(S, L, R, \omega_{\text{colour}}) = c_{\text{colour}} e^{-\frac{(L_{\text{colour}} - S_{\text{colour}})^2 L_{\text{size}} + (R_{\text{colour}} - S_{\text{colour}})^2 R_{\text{size}}}{2 S_{\text{size}} \omega_{\text{colour}}^2}}, \quad (2.20)$$

with $c_{\text{luminance}}$ and c_{colour} being the normalizing constants. A description language is taken to model the shape distribution. Let C be the length of a message required to address the boundary between L and R , then

$$-\log(q_{\text{shape}}(S, L, R, \omega_{\text{shape}})) = C + \omega_{\text{shape}}(R_{\text{shape}} + L_{\text{shape}} + S_{\text{shape}}) \quad (2.21)$$

is the shape probability function. The probability of S being split into L and R is computed along all possible merges of L and R . Then the most likely split

$$\max q(S, L, R) \quad (2.22)$$

is evaluated and the corresponding merge is performed. This merge combines the two segments L and R that have the highest probability of belonging to the same object. We may also invert this likelihood to form a dissimilarity predicate that can be used with the previously presented algorithm. The likelihood depends on the pre-determined constants ω which have been evaluated by MacDonald [57]. This optimization uses the ground truth of human segmented images to find optimal values for the constants.

Other approaches based on region merging have been introduced by various authors. Even if their algorithms slightly differ from the previously presented one their main contribution is the dissimilarity measure. In Ahuja [2] a force field is defined for each image region. Regions are attracted to each other by analyzing their respective common border and edges are detected between them. Arbelaez [4] introduced an ultrametric distance measure to model region boundary dissimilarities. This dissimilarity measure is modelled using a hierarchical segment operator to construct an ultrametric contour map (UCM). Experimental results have shown that the algorithm, initialized with an oriented watershed segmentation, outperforms other recent segmentation algorithms [6]. Due to its construction from contours it is also very effective for object boundary queries [5].

Evaluation

The results of segmentation algorithms are not easily evaluated. The judgement of a segmentation by humans is very subjective because most of the time there is not only one correct solution. Clearly, major objects should belong to their own segment, but we cannot easily classify the importance of them. For this reason, there have been methods proposed to evaluate segmentations. The evaluation is based on comparison between a segmentation and some ground truth. The ground truth can either be images segmented by different human subjects or the segmentation from a previously accepted algorithm. There are different databases of images available for comparison. Among the popular ones are the Berkley Segmentation Dataset (BSDS) [60], the PASCAL Visual Object Classes [27] and the Caltec-101 [28] dataset. Beside a broad selection of natural images, they provide human segmented ground truths and classifications of the images.

To evaluate the quality of a segmentation S , it can be compared to a ground truth G , or a set of ground truths G_k . The Rand Index, originally introduced for evaluation

of clustering, is one approach to segmentation evaluation [69]. The method sums up the pairs of pixels that belong to the same segment in S and G with the ones that do not belong to the same segment. The sum is then divided by the pairs of pixels in the image. The Probabilistic Rand Index is an extension to the Rand Index capable of measuring the quality with a set of ground truths. We define c_{ij} as being the event that pixels i and j have the same label and p_{ij} the probability of this event, then the Probabilistic Rand Index is

$$\text{PRAND}(S, G_k) = \frac{1}{T} \sum_{i < j} [c_{ij}p_{ij} + (1 - c_{ij})(1 - p_{ij})] \quad . \quad (2.23)$$

The covering of two segmentations S, S' is defined as

$$C(S' \rightarrow S) = \frac{1}{N} \sum_{R \in S} |R| \cdot \max_{R' \in S'} O(R, R') \quad , \quad (2.24)$$

where

$$O(R, R') = \frac{|R \cap R'|}{|R \cup R'|} \quad (2.25)$$

is the overlap of two regions R, R' and N is the total numbers of pixels in the image [6, 59]. The quality of a segmentation can either be evaluated by covering S with a ground truth G or the other way around. Given a set of ground truths the average value among all coverings can be taken. This way a good segmentation explains all the ground truths.

The precision-recall boundary evaluation method by Martin et al. [61] is among other popular evaluation methods . Their approach measures the fraction of boundaries detected by a segmentation algorithm compared to the ground truths. Another method introduced for clustering comparison is the Variation of Information approach [63]. It is a measure for mutual information between two clusters. In theory the approach shares some interesting properties, but as Arbelaez et al. [6] noticed it is less suitable for segmentation evaluation. This is mainly due to its unclear perceptual meaning and issue with multiple ground truths.

2.2 Image Similarity

In content-based image retrieval, stereo vision and other fields of computer vision a similarity measure between images is required. Since the raw data of the images cannot be compared directly, a signature, in the form of a feature vector or a segmentation, is

necessary for a comparison. A wide range of measures have been proposed to evaluate image similarity. Among them are vector distances, measures in a nonlinear manifold, stochastic or fuzzy measures and even approaches based on supervised or unsupervised learning. While the approaches aim to agree with the semantics they should be robust to noise, computationally efficient and invariant to the background. They are dependent on the chosen feature extraction algorithm and sometimes can not even be cleanly separated from it.

The most obvious similarity measure can be computed with the Euclidean distance of two single feature vectors in linear space. A more accurate distance measure of single vectors can be achieved by searching for a nonlinear manifold in which the feature vectors lie. The Euclidean distance can then be replaced by a geodesic distance measure improving the comparison results. The fact that single feature vectors are quite limited in their representation of a whole image makes this distance measure not very accurate. This limits the field of single vector measures to applications where small computational complexity is required. More advanced similarity measures use region based signatures or a summary of local feature vectors [23]. Both are discussed in the following paragraph.

An image signature presented in the form of a set of feature vectors is extracted from the image either by a segmentation or a feature detection algorithm. Let us weight each feature in the set z_i with a weight p_i , then the representation of an image is

$$I_m = \{(z_1^{(m)}, p_1^{(m)}), (z_2^{(m)}, p_2^{(m)}), \dots, (z_n^{(m)}, p_n^{(m)})\} \quad . \quad (2.26)$$

Given two images I_1 and I_2 the similarity measure $D(I_1, I_2)$ has to be computed by the similarity algorithm [23]. The integrated region matching (IRM) approach introduces weights $s_{i,j}$, with $1 \leq i \leq n_1$ and $1 \leq j \leq n_2$, in a significance matrix S [91]. These weights indicate the significance associating $z_i^{(1)}$ with $z_j^{(2)}$. The weights are subject to the following constraints

$$\sum_{j=1}^{n_2} s_{i,j} = p_i, i = 1, \dots, n_1 \quad (2.27)$$

$$\sum_{i=1}^{n_1} s_{i,j} = p_j, j = 1, \dots, n_2 \quad . \quad (2.28)$$

The weights are normalized with $\sum_i p_i^{(1)} = \sum_j p_j^{(2)} = 1$. A measure for the distance d between two feature vectors $z_i^{(1)}$ and $z_j^{(2)}$ is required. While this distance measure might be different depending on the application, Wang et al. [91] proposed a measure based on

colour, texture and shape of the segments. The distance of the two images I_1 and I_2 in their SIMPLIcity system is computed by

$$D(I_1, I_2) = \sum_i \sum_j s_{i,j} d(z_i^{(1)}, z_j^{(2)}) \quad . \quad (2.29)$$

Their recursive most similar highest priority (MSHP) algorithm maximizes the weights $s_{i,j}$ such that the lowest distance $D(I_1, I_2)$ is assigned the highest weight. Alternatively, the earth mover's distance by Rubner et al. [72] proposed an optimization approach to determine the weights $s_{i,j}$ under constraints 2.27 and 2.28. Given that p_i, p_j are probabilities the earth mover's distance is equivalent to the Mallows distance which is defined as

$$D(I_1, I_2) = \min_{s_{i,j}} \sum_i \sum_j s_{i,j} d(z_i^{(1)}, z_j^{(2)}) \quad . \quad (2.30)$$

A clustering of summarized local features is required for a comparison of images using the earth mover's distance. Such a clustering is presented by Grauman and Darrell [36] and has shown to be an effective approach to image matching.

In Ko and Byun [47] the Hausdorff distance was used as a measure of similarity. This distance matches every vector $z_i^{(1)}$ from image I_1 to the closest vector $z_j^{(2)}$. Among the matches, the one with the closest distance is taken as the similarity measure. For the distance to be symmetrical it is computed in both directions, from image I_1 to image I_2 and from image I_2 to image I_1 ,

$$D(I_1, I_2) = \max(\max_i \min_j d(z_i^{(1)}, z_j^{(2)}), \max_j \min_i d(z_j^{(2)}, z_i^{(1)})) \quad . \quad (2.31)$$

Instead of an image to image distance, it may also be desirable to have a region to image measure. In content-based image retrieval this allows the user to specify a region of interest to find similar images in a previously set up database. The Blobworld system uses an expectation maximization (EM) algorithm to segment an image into blobs [16]. Each blob b is represented by a colour-texture feature vector v . Given a query blob b_i the most similar blob, among every blob b_j in the database, has a score

$$\mu_i = \max_j e^{\frac{(v_i - v_j)^T \Sigma (v_i - v_j)}{2}} \quad . \quad (2.32)$$

The matrix Σ contains user adjustable weights for colour and texture features. Using fuzzy logic, the Blobworld system is able to handle compound queries.

2.2.1 Hierarchical

Image signatures in the form of a hierarchical segmentation are particularly useful for image similarity tasks. The previous approaches are only able to use geometric and photometric properties of the regions. This is due to the fact that objects may not appear at the same position in the different images. Therefore, only little structural information, such as the neighbourhood, can be used in comparisons. Hierarchical segmentations, in particular the ones based on region merging, contain more structural information in their region tree. The tree represents a segmentation at different scales and contains the merge history of the segmentation. Image similarity computation changes from a computer vision problem to a tree matching problem when region trees are solely used for image similarity computation.

Early work on the advantages of tree matching was performed by Cohen et al. [21] in stereo image matching. In more recent work segmentation trees are often enhanced with additional nodes and edges to form a graph. One possibility to interconnect the neighbour nodes is to construct a connected segmentation tree (CST) [3]. Such a tree becomes a hierarchy of region-adjacency graphs which preserves the parent-children relationship. It has shown that this data-structure is efficient for image matching. Another possibility is to construct a directed acyclic graph (DAG) by augmenting the segmentation tree [83]. New nodes are inserted wherever adjacent sibling regions are embedded in the same parent. Additionally, a connection between each region and all its children is added to the graph. The graph matching is then computed by the maximum subtree isomorphism [84]. That is the number of operations necessary to transform one subtree to match another. The possible operations include remove, insert, merge, split and match. Since insert and split are duals of remove and merge, only the later ones are applied. Matching cost between nodes is computed using the region's visual properties, such as brightness and shape. The image matching allows one-to-one, one-to-many and many-to-many node correspondence. In the follow up work of Todorovic and Ahuja [82], scale invariance was added to this approach. This scale invariance loses the dependency on the graph structure. It accounts for regions that are not present in either one of the graphs due to scale variation.

In object detection and classification Todorovic and Ahuja [81] used a modified version of the matching algorithm from Wilson and Hancock [93] for unsupervised categorization. Their system searches for the maximally matching subtree when a query image is compared to a labelled image from the database. The subtrees are maximally matching

when their structure is most similar and the cost between regions is minimal. The cost between regions is measured by their photometric and geometric similarity. Another approach to object detection is presented by Gu et al. [37] in their work based on the segmentation algorithm of Arbelaez et al. [6]. For an object a “bag of regions” is constructed by collecting the regions at every node of the tree. Each of the elements in the bag is then assigned a weight, describing their importance to the object. The weights are stored for a collection of training images. When detecting objects in a query image the presence of regions is verified according to their assigned weight. This way an object may still be classified correctly even if minor parts are missing.

2.3 Summary

In this chapter, we discussed some of the available image signatures and similarity measures. It depends upon the usage which of the image signatures and similarity measures is the most suitable for a computer vision system.

Two types of image signatures are available as presented in Section 2.1. While feature detectors focus on salient points in the image, segmentations partition the image in order to provide a set of regions. When this partitioning is done recursively a hierarchical segmentation is constructed in which regions are contained in each other. Both, feature detection and segmentation, are suitable image signatures because they provide a high level representation of the image.

In Section 2.2 different image similarity measures are shown. Most of these measures depend on a particular image signature whose quality influences the accuracy of the similarity measure. The distance between feature vectors is a popular measure of similarity that is used by many approaches. These approaches often work in conjunction with feature based image signatures. Region based image signatures are explored with hierarchical similarity measures that compute the similarity between images based on the similarity of the structure of their regions.

In this thesis we will use hierarchical segmentations as our image signature and a similarity measure based on the region properties. With this similarity measure the correspondence between regions is established. This correspondence is refined using the hierarchical structure of the segmentation.

Chapter 3

Descriptive Segment Filter

In earlier chapters we have discussed the extraction of features in natural images and have seen that it is a very challenging problem. The feature space is often very large and makes further processing an expensive task. Our approach reduces this feature space such that algorithms can work more efficiently. This is a challenging problem because important features need to be preserved while others can be discarded.

We have chosen to use a segmentation as our image signature. Segmentations have a broad field of application and, in particular, were used for image matching before [21, 34, 35, 67, 81, 82, 83]. As discussed in Section 2.1.2 segmentation algorithms may produce either, flat and hierarchical groupings. While a flat segmentation algorithm that partitions an image into a given number of segments can be used to calculate an image signature, we argue in this thesis that, a hierarchical algorithm is more suitable for image matching. Only with hierarchical segmentations small, but representative, portions of the image are preserved. The disadvantage of hierarchical segmentations is the larger number of segments that need to be processed, e.g., by an image matching algorithm. The goal of our filtering is the reduction of the available segments to a much smaller set of descriptive segments. Since these descriptive segments become the image signature they need to contain the unique properties of the image. The challenge is the reduction of the available segments to the ones that describe the image optimally. Such an optimal description contains segments that can also be found in optimal descriptions of images of the same scene. By having matching segments in both descriptions, we will show in Chapter 4 how to match these images.

3.1 Definition of Descriptive Segments

A hierarchical segmentation is an unsupervised partitioning of an image into a large number of segments. The granularity of these segments range from fine to coarse, from (super-)pixels to areas respectively. In this structure a segment is composed of its children or is a (super-)pixel. Many of these segments are intermediate segments, meaning they only exist as part of a larger object or area. These segments are not of interest to further tasks, like image matching, because their information is contained in their parents. Among the parents of an intermediate segment there might be a segment that represents an object or another distinct part of the image. Let us define the properties of these segments as follows,

1. Interior similarity: The children within the segment are very similar to each other and to the segment itself. This includes similar brightness, colour and size.
2. Exterior dissimilarity: The segment is very different to its surrounding. This includes dissimilar brightness, colour and size.

We argue that segments fitting this definition should be part of the image signature. Let us call them descriptive segments. This definition generalizes maximally stable extremal regions by including colour and size of the segments. The interesting property of descriptive segments is their stability. Their appearance may change when the viewpoint or illumination differs between images of the same scene, but this will not affect their interior and exterior properties. Segments that possess these independences are particularly useful for image representation. They describe the unique properties of the image which can also be found in other images of the same scene.

3.2 Segment Properties

Let us have a closer look at segment S in segmentation P of the image I . With whatever algorithm I is segmented, all segments in P share a common set of attributes. Since a segment is always composed of its underlying pixels we let $|S|$ be the number of pixels in S . Each pixel p in I is defined by its position (x, y) and its colour p_{rgb} in RGB colour space. The mean colour of S is

$$S_{rgb} = \frac{\sum_{x,y \in S} p_{rgb}(x, y)}{|S|} . \quad (3.1)$$

In comparison to the RGB colour space the $L^*a^*b^*$ colour space is a three dimensional perceptually uniform colour space with L^* being the lightness dimension and a^*b^* being the colour-opponent dimensions. In this colour space luminance L^* is separated from chrominance a^*b^* making individual analysis of segment luminance $S_{\text{luminance}}$ and segment chrominance $S_{\text{chrominance}}$ possible. The conversion from RGB to $L^*a^*b^*$ colour space requires several steps due to the non-linearity of the colour spaces [75]. RGB values are gamma corrected by default and have to be linearized before being transformed. The following inverse gamma function is used for this correction, so

$$G_T(C) = \begin{cases} (1+f)C^g - f, & \text{for } t < C < 1 \\ s \cdot C, & \text{for } 0 \leq C \leq t \end{cases} \quad (3.2)$$

Using the HDTV-RGB standard (ITU-R BT.709) the constants are $g = 0.45$, $f = 0.099$, $s = 4.5$ and $t = 0.018$. After the RGB values have been linearized they are transformed to the XYZ colour space by the following matrix multiplication

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} G_T(R) \\ G_T(G) \\ G_T(B) \end{bmatrix} \quad (3.3)$$

The white point of the conversion shall be the CIE Standard Illuminant D65 defined as

$$\begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} = \begin{bmatrix} 0.9504 \\ 1.0000 \\ 1.0889 \end{bmatrix} \quad (3.4)$$

With the help of the non-linear function

$$f(r) = \begin{cases} r^{\frac{1}{3}}, & \text{for } r > 0.008856 \\ \frac{7.787 \cdot r + 16}{116}, & \text{for } r \leq 0.008856 \end{cases} \quad (3.5)$$

the coordinates in XYZ colour space are transformed to the desired $L^*a^*b^*$ colour space by

$$\begin{bmatrix} L^* \\ a^* \\ b^* \end{bmatrix} = \begin{bmatrix} 0 & 116 & 0 & -16 \\ 500 & -500 & 0 & 0 \\ 0 & 200 & -200 & 0 \end{bmatrix} \begin{bmatrix} f\left(\frac{X}{D_x}\right) \\ f\left(\frac{Y}{D_y}\right) \\ f\left(\frac{Z}{D_z}\right) \\ 1 \end{bmatrix} \quad (3.6)$$

Due to the non-linearity of the colour space the Euclidean distance between two colours accounts for their perceptual change [75]. Given the average colour, the variance of the pixels in S is

$$S_v = \frac{\sum_{x,y \in S} p_{L*a*b*}(x,y) - S_{L*a*b*}}{|S| - 1} \quad . \quad (3.7)$$

The variance is a measure for the similarity of the pixels within S . The more pixels differ from their mean colour the more perceptually variant the segment is.

3.2.1 Cost

In our model of a hierarchical segmentation each segment S is composed of two children S_l and S_r or it is a leaf. The cost S_c is related to the probability that the two children S_l and S_r are merged to S at time t . Region merging or growing algorithms make use of such merge costs in their computation. The cost has interesting properties for the purpose of filtering. It reflects how likely it was to combine the two children to form S . Searching for unlikely merges may help us find the descriptive segments because the combination of two distinct segments is rare and by definition descriptive segments are distinct from their surrounding.

We use a combination of individual costs from the hierarchical segmentation criterion by MacDonald [57] introduced in Section 2.1.2. The cost is composed of $q_{\text{luminance}}$, $q_{\text{chrominance}}$ and depending upon the usage of q_{size} . The two costs used in this work are

$$\begin{aligned} S_{lc} = & c_{\text{luminance}} e^{-\frac{(S_{l\text{luminance}} - S_{\text{luminance}})^2 |S_l| + (S_{r\text{luminance}} - S_{\text{luminance}})^2 |S_r|}{2|S| \omega_{\text{luminance}}^2}} \\ & + c_{\text{chrominance}} e^{-\frac{(S_{l\text{chrominance}} - S_{\text{chrominance}})^2 |S_l| + (S_{r\text{chrominance}} - S_{\text{chrominance}})^2 |S_r|}{2|S| \omega_{\text{chrominance}}^2}} \end{aligned} \quad (3.8)$$

and

$$S_{lcs} = S_{lc} + c_{\text{size}} \left(\frac{|S_l|}{|S|} \right)^{\omega_{\text{size}} - 1} \left(\frac{|S_r|}{|S|} \right)^{\omega_{\text{size}} - 1} \quad . \quad (3.9)$$

The parameters of the cost function were empirically determined to be

$$\{\omega_{\text{luminance}}, \omega_{\text{chrominance}}, \omega_{\text{size}}\} = \{1, 1, 0.7\} \quad (3.10)$$

with

$$c_{\text{luminance}} = c_{\text{chrominance}} = c_{\text{size}} = 1 \quad . \quad (3.11)$$

These parameters have shown to map the difference between the segments realistically.

Independent of the segmentation criterion the cost is a measure for the similarity between the two children of S . We will later show that analyzing the cost can help us find descriptive segments.

Normalization

When we compare the cost of merging small segments with the one of merging large segments, we notice an increase in the order of magnitudes. With increasing segment size each cost component becomes larger. Because the size component in Equation (3.9) is directly dependent on the segment size, this increase is no surprise. With larger segments, the number of pairs of segments considered for merging diminishes in region merging, and increasingly expensive merges need to be selected. This implies that segments with a large luminance and chrominance cost are merged, which explains the increase of these two components.

A normalization is required for the cost of small segments being comparable with the one of large segments. Nock and Nielsen [66] in their work on statistical region merging (SRM) perform such a normalization to adjust the threshold of the merge predicate. Given two adjacent regions S and T , with their average colours $\bar{S}$ and $\bar{T}$, they can be merged if the threshold $b(S, T) \geq |\bar{S} - \bar{T}|$ is not exceeded. Having an image in RGB colour space the threshold is computed for each of the colour channels individually. The threshold

$$b(S, T) = g \sqrt{\frac{1}{2Q} \left(\frac{1}{|S|} + \frac{1}{|T|} \right) \ln \frac{2}{\delta}} \quad (3.12)$$

is a function of the region size and the constants g , Q and δ . Using an RGB image I with $|I|$ pixels the constants are $\delta = \frac{1}{6|I|^2}$ for the image size normalization, $Q = 32$ for the colour space quantization and $g = 256$ is the colour space depth for 8 bit images.

Let us assume a comparison of the cost of two regions S and T is required. Without loss of generality we compute the boolean comparison of $S_c < T_c$ by normalizing both costs

$$S_c \cdot b(S_l, S_r) < T_c \cdot b(T_l, T_r) \quad (3.13)$$

$$S_c \cdot g \sqrt{\frac{1}{2Q} \left(\frac{1}{|S_l|} + \frac{1}{|S_r|} \right) \ln \frac{2}{\delta}} < T_c \cdot g \sqrt{\frac{1}{2Q} \left(\frac{1}{|T_l|} + \frac{1}{|T_r|} \right) \ln \frac{2}{\delta}} \quad (3.14)$$

$$S_c \cdot \sqrt{\frac{1}{|S_l|} + \frac{1}{|S_r|}} < T_c \cdot \sqrt{\frac{1}{|T_l|} + \frac{1}{|T_r|}} \quad (3.15)$$

Since the variables δ, Q, g are positive constants across the image they get cancelled in the inequation above. All that is left is the normalization by the size of segments S and T , respectively their children S_l, S_r and T_l, T_r .

3.3 Queue-based Approach

The core of region merging algorithms, such as the one by Beaulieu and Goldberg [11] presented in Section 2.1.2, is a merge queue. This queue contains every possible merge of two adjacent segments with an associated cost given by the properties of the two segments. Most cost functions, like the one used by MacDonald [57], define the cost as a dissimilarity measure where the appearance of the segments play an important role. Merging the two segments with the lowest dissimilarity in the queue combines the two most similar segments in terms of the defined cost function. The measure of cost has, consequently, great influence on the resulting segmentation. We claim that the analysis of the merge order and the merge cost for the two segments at merge time t will let us find the descriptive segments.

3.3.1 Cost Analysis

A hierarchical segmentation often comes with information about the time of merge t for each segment. Given this information it is possible to reconstruct the history of merges. For each merge in this history we reconstruct its theoretical merge cost S_{lcs} . The necessary luminance, chrominance and size parameters can be computed directly from the image without further data from the segmentation. We use this cost to measure the visual implications of a merge. Even if the exact cost function used by the segmentation algorithm is not known this gives us a measure of the probability for the combination of two segments. Since the history does not contain information about the unperformed merges we are unable to reconstruct further information, such as the mean of the merge costs at any point in time. Without such information we can only rely on the cost history itself for our analysis.

Figure 3.1 shows an example of such a sequence of merge costs from an example image. The upper bound of this history seems to be limited by an exponential curve. We can explain this limit by the fact that with increasing segment size less merges are available for consideration, meaning a merge of less similar segments is required. The cost is additionally dependent upon the segment size which increases as well. While the upper bound of our cost is limited the lower bound is not. This implies that a merge of two quite similar segments is possible at any time during the history.

Our descriptive segments as defined in Section 3.1 are uniform in terms of their interior. Let us assume D is such a descriptive segment. The pixels within D will be similar making it very probable that D is assembled by the segmentation algorithm.

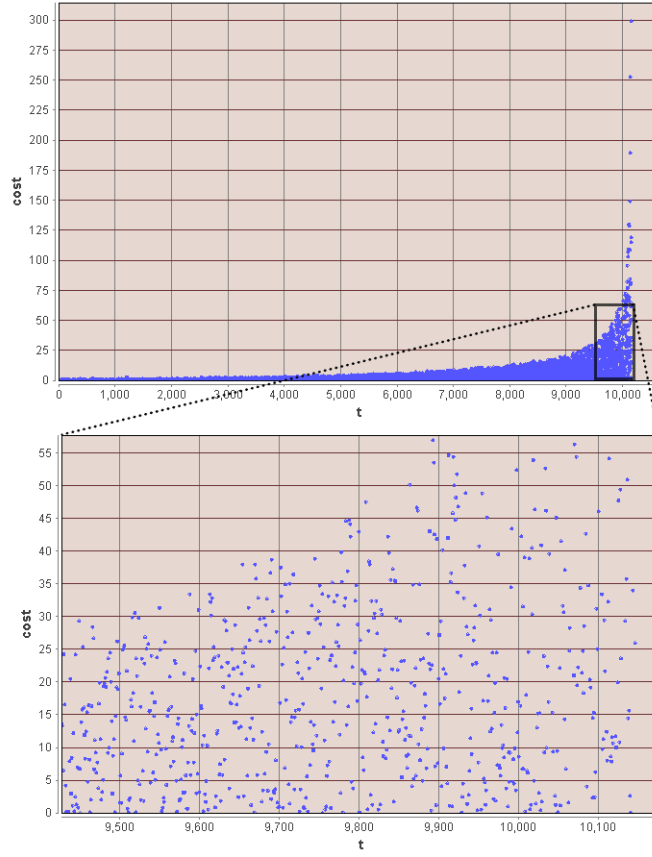


Figure 3.1: The S_{lcs} cost at the time of merge t for each segment S . The data is reconstructed from the bikes1 example image introduced in Appendix A.

Once D is put together the merge queue will contain all possible combinations of D with its neighbours. Assuming the cost function used by the segmentation algorithm includes appearance-based properties, like brightness and colour of the segments, then it will be expensive to merge D with one of its neighbours. This lets the segmentation algorithm perform cheaper merges before combining D with a neighbour. At the time when there is no cheaper merge available, D has to be combined with a neighbour to form D_p . This brings together two dissimilar segments and we will notice a high cost in our merge sequence. Since there is only a limited number of such merges within the history we should be able to detect them by a successive analysis of the reconstructed cost.

3.3.2 Outlier Detection

The history of every segmentation is different and with it is its cost. The absolute value of the cost is dependent on the similarity and the size of the segments. In Figure 3.2 an extreme example of cost values is shown. They remain near zero before increasing rapidly for the last few merges where the squares are combined. In comparison to the example of Figure 3.1 maximum cost values are about fifteen times higher for an image that has approximately ten times less segments due to its size. This significant difference of absolute cost values require a local examination of the cost.

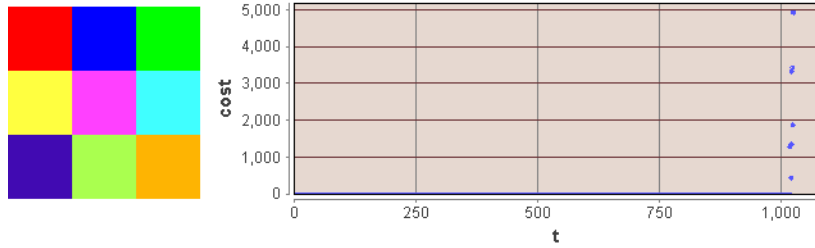


Figure 3.2: The cost of the merges stays relatively low while the squares are assembled. For the last merges the cost increases rapidly because the squares get combined.

As discussed earlier we notice a potentially exponential upper bound of our cost. We could fit an exponential curve in the form of $a \cdot e^{b \cdot x}$ to this boundary, e.g., using non-linear least squares. However, analyzing the descriptive segments in their merge history shows that a global reconstruction of this curve is not required. In Figure 3.3 we see two descriptive segments at the time of their merge with a neighbour. In both cases the cost is a local maxima across its preceding merge costs. There is no need to construct an approximated upper bound of the cost to identify these merges. A statistical analysis will most likely be more accurate than using a distance measure from an approximated upper bound.

Given the discrete cost samples in a merge history we use outlier detection to find the local maxima. A merge cost is an outlier if it is distant from other costs in a local set Q . Let us construct the local set

$$Q = S_{lcs_j} \forall j \in i - q \dots i \quad (3.16)$$

with the costs of a potential outlier at position i in the merge history. The size of this set $|Q|$ is controlled by the constant q which was chosen to be ten in our evaluation. The

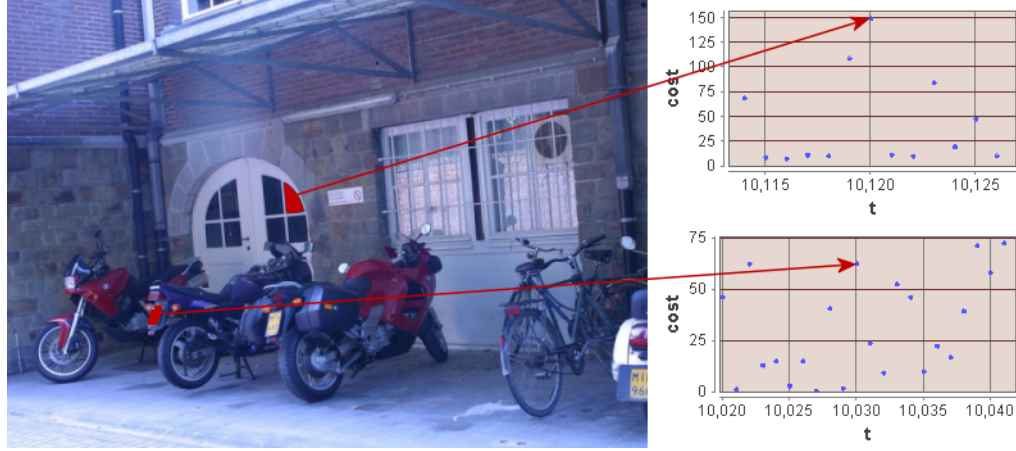


Figure 3.3: The perspective cost of two descriptive segments at the time of their merge with a neighbour. The merge cost is a local maxima across its preceding and successive merges. The example shows the analysis of the bikes1 image from the experimental dataset.

cost at position i is an outlier if it is very distant from the other elements in Q

$$S_{lcs_i} \gg S_{lcs_j} \forall j \in Q, j \neq i \quad . \quad (3.17)$$

Methods of outlier detection make only small assumptions about the distribution of the samples since an outlier can occur by chance in any dataset. This is beneficial to our approach since we are not able to determine the underlying distribution of the cost values. It is important to notice that an outlier is not removed from the original dataset and is available in subsequent sets of Q . Next we will present three different approaches towards outlier detection.

Deviation from Window Mean

A simple form of outlier detection is by using the deviation from the mean. Under the assumption that the samples are independent and Gaussian distributed let us compute the mean Q_μ and standard deviation Q_σ for the costs $S_{lcs} \in Q$. The cost S_{lcs_i} is an outlier if

$$\frac{S_{lcs_i} - Q_\mu}{Q_\sigma} > O_d \quad . \quad (3.18)$$

The constant O_d is the minimum deviation threshold. The larger this threshold the further away from the mean an outlier has to be to be. Thresholds lower than one are not recommended since outliers are then less than the standard deviation away from

the mean. In Table 3.1 the influence of the threshold is evaluated using a subset of our dataset introduced in Section 3.5.1. On the one hand we see that for these images thresholds $O_d \geq 2$ are too limiting, meaning not many segments pass the filter. On the other hand we argue that a threshold of $O_d = 1$ is too liberal considering the huge number of segments that are classified as outlier. The threshold $O_d = 1.5$ is a reasonable alternative and will be our choice for the evaluation in Section 3.5.5. An example where this outlier detector has been applied is shown in Figure 3.4.

			$O_d = 1$		$O_d = 1.5$		$O_d = 2$		$O_d = 3$		$O_d = 4$	
I	$ P $	$ D $	$ F $	$ Z_0 $	$ F $	$ Z_0 $	$ F $	$ Z_0 $	$ F $	$ Z_0 $	$ F $	$ Z_0 $
backyard1	10221	9	2342	2	1002	1	389	1	63	1	14	1
bikes1	10147	22	2282	10	1169	7	521	4	100	3	32	0
boesch2	10387	8	2379	4	1049	2	410	2	64	0	13	0
graffiti1	9804	10	2183	1	891	0	321	0	50	0	15	0
home3	10095	16	2297	5	1036	4	439	2	79	0	20	0
house2	10246	10	2367	7	1090	3	468	1	102	0	36	0
Leuven1	10250	10	2287	2	1043	1	417	0	68	0	17	0

Table 3.1: Evaluation of the threshold O_d using images from the experimental dataset introduced in Section 3.5.1. Please refer to Section 3.5.3 for the definition of the symbols.

Chauvenet

Another method based on the assumption of independent Gaussian distributed samples is Chauvenet’s criterion [17]. By using the sample size, mean, standard deviation and Gaussian error function $erf(x)$ the probability p of the sample having the given value is computed as follows

$$p = erf\left(\frac{S_{lcs_i} - Q_\mu}{Q_\sigma \sqrt{2}}\right) \quad . \quad (3.19)$$

Let O_c be the threshold, then S_{lcs_i} is an outlier if

$$p \cdot |Q| < O_c \quad . \quad (3.20)$$

The typical value of the threshold O_c is 0.5. According to Chauvenet, samples with lower probability are very unlikely to occur and are outliers. Figure 3.5 provides an example filtering result with the Chauvenet criterion. The performance over a comprehensive dataset is evaluated in Section 3.5.5.



Figure 3.4: Visual result of the deviation based outlier detector for the bikes1 image from the experimental dataset. The standard segmentation algorithm is used to partitioned the image into 10'147 segments. These segments, shown in the left image, were filtered by the deviation based outlier detector. The result in the right image contains 1'169 potentially descriptive segments.

Q Test

For the Q Test by Dean and Dixon [24] the samples in the set Q are ordered by their cost value. The relation between the closest value in Q to S_{lcs_i} to the range of all values in Q is compared to a predefined table of confidence values. There are three different confidence levels provided by Dean and Dixon as shown in Table 3.2.

	Number of samples							
Confidence level	3	4	5	6	7	8	9	10
90%	0.941	0.765	0.642	0.560	0.507	0.468	0.437	0.412
95%	0.970	0.829	0.710	0.625	0.568	0.526	0.493	0.466
99%	0.994	0.926	0.821	0.740	0.680	0.634	0.598	0.568

Table 3.2: Q Test thresholds for different confidence levels and sample sizes.

Given the sample size $|Q|$ and the confidence level the threshold O_q can be evaluated. Let us assume S_{lcs_k} is the cost closest to S_{lcs_i} , then S_{lcs_i} is an outlier if

$$\frac{S_{lcs_i} - S_{lcs_k}}{\max(Q) - \min(Q)} > O_q \quad . \quad (3.21)$$

The approach is most suitable for small datasets because the threshold O_q changes with the sample size. The approach is based on the ordered data samples drawn from a



Figure 3.5: Chauvenet’s criterion was used in this example of outlier detection for the image bikes1 from the experimental dataset. Like in Figure 3.4 the original image was segmented by the standard segmentation algorithm. In comparison to the deviation based filtering Chauvenet’s criterion is more restrictive and finds only 879 potentially descriptive segments. These are shown in the right image.

Gaussian distribution. Like the previous methods of outlier detection such a distribution of Q is to be assumed. Table 3.3 shows experimental results using different confidence levels. We see that even for the lowest confidence level only a few segments are classified as outliers. For this reason we use the 90% confidence level in our evaluation. An example using this confidence level is presented in Figure 3.6.

3.3.3 Choosing the Segment

With the queue analysis and outlier detection we are able to identify particularly expensive merges. These merges combine two neighbours S_l and S_r to form a new segment S . In order to find the descriptive child one has several approaches. The analysis of the variance is one of those. According to our definition of descriptive segments in Section 3.1 their interior is uniform. The child whose variance is lower matches this definition better because pixels in this segment are more similar to each other than in the other segment. The selection of the descriptive segment by cost follows the same argumentation. The cost is a measure for dissimilarity between two segments which means that it is lower for the child of S whose children were more similar to each other. In our experiments, however, we found that choosing the smaller of the two children of S leads to the best results. In comparison to the variance and cost criteria the size criterion incorporates the exterior property of the descriptive segment definition as well. The merge of a segment

			$O_q = 0.412$		$O_q = 0.466$		$O_q = 0.568$	
I	$ P $	$ D $	$ F $	$ Z_0 $	$ F $	$ Z_0 $	$ F $	$ Z_0 $
backyard1	10221	9	137	1	106	1	61	0
bikes1	10147	22	147	1	108	0	71	0
boesch2	10387	8	146	0	129	0	93	0
graffiti1	9804	10	138	0	110	0	69	0
home3	10095	16	130	0	107	0	74	0
house2	10246	10	168	0	142	0	83	0
Leuven1	10250	10	158	0	123	0	73	0

Table 3.3: Evaluation of the threshold O_q with the sample size $|Q| = 10$. The thresholds represent confidence levels of 90%, 95% and 99%. The presented data is computed using images from the experimental dataset introduced in Section 3.5.1. For the definition of the symbols please refer to Section 3.5.3.

that is uniform in its interior and unique among its neighbours is avoided for as long as possible. This delay is seen by the difference in size between S_l and S_r .

3.4 Two-phase Approach

In this section we present a second approach to descriptive segment filtering. This approach is solely based on the segmentation hierarchy and does not require information about the merge history. This makes it independent of the used hierarchical segmentation algorithm. The filtering is split into two phases. In the first phase a selection of potentially descriptive segments are found. In the second phase these potentially descriptive segments are ordered by their relevance to the image. We will later show that with the right selection and sorting criteria descriptive segments will be among the most relevant segments.

3.4.1 Selection

The input to our first phase is a hierarchical segmentation. We assume this hierarchy is represented by a binary tree where each node has two children or is a leaf. Leaves should represent pixels or super-pixels and be the smallest unit in our segmentation. A bottom up segmentation algorithm, such as region merging or region growing, builds a hierarchy



Figure 3.6: In comparison to the deviation and Chauvenet based outlier detectors the Q Test criterion is very restrictive. In this example of the bikes1 image from the experimental dataset only 147 segments were classified as potentially descriptive. The other two outlier detectors find more descriptive segments on the same example as shown in Figure 3.4 and Figure 3.5. Visually this difference can be seen by comparing the rightmost image of the three figures.

that results in a single root segment. The selection algorithm will visit each segment in this tree by traversing it from its root. Without being too restrictive it will decide if a segment is descriptive. This decision is based on the information available about the segment and its parents. This includes structural relationship in our selection. Since selection is a filter to the following sorting phase it should not be too restrictive. It is more desirable for a non-descriptive segment to pass the selection than for a descriptive segment to get discarded. While in the first case the sorting might penalize the segment, in the later case it is not available anymore.

Different strategies are possible to evaluate if a segment S in the segmentation hierarchy might be a descriptive segment. We present three of those strategies that depend on the relationship between S , its parent S_p and possibly its grandparent S_{pp} . A combination of these strategies were used to evaluate the selection performance and will be presented later.

Variance The pixel variation of uniform segments are relatively small. When such a segment gets combined with a neighbour the variance jumps due to the broader spectrum of pixels in the segment. In this case the variance increases from S to its parent S_p . When selecting by variance, segments are chosen that show such an increase.

Size Many segmentation algorithm aim at an approximately balanced merge tree and in such a tree all segments at a given height in the tree have similar size. A descriptive segment that is distinct from its neighbouring segments, remains stable over many merges and therefore its size will fall behind its neighbours. until it will be combined with a neighbour, but then their neighbour will be a much larger segment. This property is exploited when selecting by size. The multiplier $m = \frac{|S_p|}{|S|}$ has to exceed a chosen threshold for a potentially descriptive segment to be selected. In Figure 3.7 such a selection is presented for an example image.

Cost The cost introduced in Section 3.3.1 measures the dissimilarity between two neighbour segments. In segmentation algorithms such dissimilarity measures are used to determine the merge order. The dissimilarity measure provides information about how likely it was that two segments were combined. For the following methods of selection we use both the S_{lc} and S_{lcs} cost functions.

Increase When using a dissimilarity measure for selection we notice two properties that a descriptive segments fulfils. First, the children of a descriptive segment are similar which means their dissimilarity measure, or cost, will be low. Second, a descriptive segment is very different to its neighbours implying a high parent cost. A segment is, therefore, potentially descriptive when its parent cost is higher than its own cost.

Decrease An interesting evaluation is by selecting the counter argument of the one above. Given the previous theory this should select the wrong segments. We will see if this holds in our evaluation.

Peak The peak selection is an extension of the selection by increasing cost. We assume S is a descriptive segment that is combined with a highly dissimilar neighbour to form S_p . For a segment to be descriptive it should be outstanding in its neighbourhood. It is uncommon for a parent of a descriptive segment to be descriptive as well. For this reason S_p should be more similar to its neighbour than S was, meaning the cost of S and S_{pp} are lower than the one of S_p . This makes S a descriptive segment when there is a peak of the cost at S_p .



Figure 3.7: The size selector with a threshold of $m \geq 3$ has chosen 1’762 segments for the bikes1 image from the experimental dataset. We see these potentially descriptive segments on the right and notice that it is covered completely. This is due to the hierarchical segmentation of the image. The filtered image is coarser in comparison to the segmentation shown in the left image. This segmentation is computed using the standard segmentation algorithm from the original image given in Appendix A. The visual difference between the segmentation and the selection is not large, but we can see the change for instance at the floor where the segments are coarser.

3.4.2 Sorting

The input to the sorting phase is an unordered set of possibly descriptive segments extracted by the selection phase. This unordered set is sorted by the descriptiveness of the segments such that the most descriptive segments appear at the top of the list. By traversing this list in order the most relevant segments can be extracted for further processing.

The order is evaluated by comparing the properties of two segments with each other. These properties include the visual and structural information, such as variance, size and cost, in the comparison. In contrast to the selection phase, where the comparison is done between a segment and its parent, in the sorting phase the properties are compared between two unrelated segments. This requires sorting criteria that do not favour small or large segments.

For sorting a total order of the segments is established by comparing segment properties. Given two segments S and T these properties can either be compared relatively or absolutely. While in absolute comparison the properties are directly compared between S and T , in relative comparison the relationship of S to S_p and T to T_p is determinant.

Without loss of generality let us assume S is more descriptive than T , meaning $S \geq T$. We propose the following comparators for sorting and explain why they are suitable for this purpose.

Variance (relative) The variance will increase when a descriptive segment, whose variance we assume is low, gets combined with one of its neighbours. Given two segments the one whose difference is larger is more descriptive than the other because it was more uniform before the merge. The variance of a segment is independent on its size and can be used as criterion by $S_{p_v} - S_v \geq T_{p_v} - T_v$.

Size (relative) Size is a good indicator of segment stability. The merge of a descriptive segment with a neighbour is avoided for as long as possible. As a result a descriptive segment gets combined with a much larger neighbour. The segment whose difference in size to its parent is larger will be considered the more descriptive. The criterion for this comparator is $|S_p| - |S| \geq |T_p| - |T|$.

Parent cost (absolute) As discussed earlier descriptive segments are very different to their neighbours. We can measure this dissimilarity by the cost introduced in Section 3.2.1. The cost required to merge a segment with its neighbour is available from the parent of the segment. Given two segments the one that was more expensive to merge with our measure of cost is the more descriptive one. Formally, this is defined by $S_{p_{lcs}} \geq T_{p_{lcs}}$.

Combinations The combination of criteria from the previous comparators introduce other interesting strategies. With such combinations our sorting gets more sensitive to descriptive segments, ranking them higher than with a single criterion. However, experiments have also shown that combining criteria is not trivial. Results may easily be worse than with a single criterion. Next we present two combinations which we found to be successful in practise.

Variance (relative) and parent cost (absolute), size (relative) We assume that both, the variance and the cost criteria must hold for a descriptive segment. In case the two do not clearly identify the more descriptive segment the size argument is used as a fallback. In our experimental evaluation in Section 3.5.5 this criteria was the best performing one. Figure 3.8 shows an example of sorting by this criteria.

Variance (relative) and normalized parent cost (relative), size (relative)

This is the same as the above, only we normalize the cost and compare it relatively between the two segments. Normalizing the cost has the advantage of being more fair when comparing small to large segments. However, we noticed that by normalizing the cost and compare it absolutely small segments are considered too descriptive. Here, relative comparison shows better behaviour and provides better results.



Figure 3.8: This Figure shows the 30 most descriptive segments using the combined “variance (relative) and parent cost (absolute), size (relative)” sorting criteria. Like in Figure 3.7 selection was done by size with a threshold of $m \geq 3$. We see that from the segmentation on the left the two-phase approach is capable of outlining important features of the image by identifying the descriptive segments. For more results please see Section 3.5.5 or refer to Appendix A.

3.5 Results

In this section the results of our descriptive segment filters are presented. We will first define our experimental dataset and a syntax for the evaluation. In the succeeding section we will experimentally determine the best performing selection and sorting criterion for the two-phase approach. Then we advance to the performance evaluation where we will compare the two-phase approach to the queue-based approaches and their outlier detectors. After the analysis of the results and comparison between different segmentation algorithms we conclude the section with a discussion.

3.5.1 Experimental Dataset

The experimental evaluation of our filters is performed on a set of 36 natural images. These images were taken from seven different objects of interest. In each of these seven image sequences the conditions vary. The used dataset covers variations in viewpoint, image blur and light adjustment. While three of the image sequences were presented by Mikolajczyk et al. [65] in their work on affine regions detectors the other four sequences were captured by ourselves with a typical consumer camera. The different sequences with their origin are summarized in Table 3.4. An example image of each sequence is presented in Figure 3.9. For a complete overview of the experimental dataset please refer to Appendix A.

Name	Number of images	Varying condition	Source
backyard	4	Viewpoint	Own work.
bikes	6	Blur	Mikolajczyk et al. [65]
boesch	4	Viewpoint	Own work.
graffiti	6	Viewpoint	Mikolajczyk et al. [65]
home	5	Viewpoint	Own work.
house	5	Viewpoint	Own work.
Leuven	6	Brighthness	Mikolajczyk et al. [65]

Table 3.4: Overview of the experimental dataset used to evaluate our filters. The 36 natural images are arranged in sequences that each show a scene under varying conditions. While three of the sequences were taken from the literature, the other four were captured by ourselves.

There are many well known datasets in the literature and the Berkley Segmentation Dataset [60] is among the most popular ones. Those datasets are not suitable for our evaluation because they were designed to evaluate segmentation algorithms and contain a complete partitioning of the images. With our filter, however, we aim to provide only a subset of a partition in form of the descriptive segments. There is no need for every pixel to be in a partition of the image after filtering. Moreover, the segments may even be embedded in each other, making a minimized hierarchical set of segments available. To our knowledge there is no dataset that provides such a set of descriptive segments. For this reason we have selected the descriptive segments manually by the process described in the next section.



Figure 3.9: One example image of each sequence in the experimental dataset. The images are from the sequences (f.l.t.r.) backyard, bikes, boesch, house, graffiti, home and Leuven.

Our filtering is performed with the input from two different algorithms. While with the first one all of the 36 natural images are tested, the second one is only used with one image of each sequence. It serves as verification of our filtering and shows that it is not dependent on a specific region merging or region growing algorithm. Our standard algorithm is the region merging algorithm presented by D. MacDonald [57]. Our images have a typical size of 640x480 pixels and are pre-segmented before being presented to the region merging algorithms. This pre-segmentation is done using SLIC superpixels by Achanta et al. [1]. The pre-segmented images contain approximately ten thousand superpixels and are then bottom up segmented by the region merging algorithm. The alternative algorithm uses a region growing approach by Lareau [50]. The criteria used in this region growing is based on the Euclidean distance of the segment colour in $L^*a^*b^*$ colour space. As input to the region growing we use a mean shift segmentation [33]. Depending on the image we get about ten to fifty thousand superpixels.

3.5.2 Descriptive Segment Selection

It is crucial for the evaluation of our filters that the manually selected descriptive segments comply with the definition of descriptive segments from Section 3.1. We are aware that visual selection is subjective. For this reason only segments that obviously satisfy the properties of descriptive segments are selected. We specifically pay attention to the

segment border which should clearly separate it from its neighbours. In particular this is the case for segments whose brightness and colour differ much from segments in their neighbourhood. Furthermore, brightness and colour is required to be stable within a descriptive segment. A large difference in size of a segment to the neighbour it is merged with, may also indicate a descriptive segment. When all of these properties apply to a segment, the segment is clearly descriptive according to our definition.

The large number of segments within a hierarchical segmentation makes the manual selection of descriptive segments a time consuming task. We found the most effective approach to choose descriptive segments within a hierarchical segmentation is by browsing backwards through its merge history. Because we start from the last merge in the history, every step towards earlier merges refines the image until the (super-)pixel level is reached. Due to the large number of merges in the merge history it is not feasible to evaluate every merge available. According to the definition of descriptive segments there is no need to consider intermediate segments that are only available as part of larger objects. This allows us to stop the search for descriptive segments at a level at which objects are found. This object level is dependent on the image and the used segmentation algorithm. In our case the two segmentation algorithms favour to merge intermediate segments before combining objects. This is due to their merge predicates which include luminance, chrominance, size and boundary length of the segments to consider. The advantage of this property is that we only need to consider the last few hundred merges in the history to find the set of descriptive segments. In Figure 3.10 this process is visualized.

Careful evaluation of every merge is required to identify the descriptive segments. We see the two segments that were merged with every step we take through the merge history. We analyze the two segments and compare them to the definition of descriptive segments. In case one of them complies with this definition we add it to the set of descriptive segments. An example of descriptive segment selection is given in Figure 3.11.

We can choose descriptive segments in unsegmented images by looking for regions in the image that comply with the definition of descriptive segments. When we compare such a selection with one using the segmentation hierarchy we notice that not all desirable segments are available. In Figure 3.12 the difference between the desired and the actual selection is shown for three example images. In case a desired segment is not available we cannot penalize the filtering for the bad segmentation and omit the segment. In other cases, a the filter finds a very similar segment than what has been marked as a



Figure 3.10: We browse through the merge history to find descriptive segments. Starting from the last merge (root level) we proceed until we reached the object level of the segmentation. We consider it unnecessary to go further than this object level because major segments were assembled at this point and mostly intermediate segments are merged before it. For every merge we compare the two segments involved to see if either of them complies with our definition of descriptive segments. In case one of them does, it is added to the set of descriptive segments. The example in this figure explains the process for the bikes1 image in our experimental dataset.

desired segment (a subsegment for instance). We do not count this partial overlap as a successful identification of a descriptive segment. In our evaluation we have to keep that in mind and do not expect an error free filtering. We use a large dataset to minimize the effect on the outcome. For each image our manual selection should contain as many descriptive segments as possible. Please refer to Appendix A for a complete overview of the manually selected segments.

3.5.3 Evaluation Syntax

Let us first establish a common syntax for our evaluation. Given any set of segments X , let $|X|$ be the size of the set and $\overline{|X|}$ be the average size of many sets. For the evaluation of the approaches every image I in the dataset is considered. Let P be the segmentation of I , built with one of our two segmentation algorithms. We have manually chosen a set of descriptive segments $D \in P$ as previously discussed. We apply one of



Figure 3.11: The three descriptive segments were selected from the bikes1 image in our experimental dataset. All of three segments are distinct from their surrounding, have a clear border to their neighbours and are internally stable. The window segment (no. 1) is enclosed by the door whose colour is very different compared to the window. The same applies to the sign segment (no. 2) that strikes out on the wall due to its different colour. Even more conspicuous is the number plate segment (no. 3) that stands out among its neighbours. The three segments satisfy not only the exterior, but also the interior property of descriptive segments. The pixels within them might have some variance but over all they are very similar. This qualifies them for being a descriptive segment.

the various filters discussed in Section 3.3.2 and 3.4.1 to P and we call the respective result F , i.e., the segments passing the filter. This set F contains the segments which were classified as potentially descriptive by the approaches. The intersection $Z_0 = F \cap D$ describes the overlap between the potentially descriptive segments and the manually selected descriptive segments. The approaches are capable of providing only a limited set of descriptive segments. For every F there also exists a subset $F_x \subset F$ containing only the x most descriptive segments. Expanding the previous definition of Z_0 we define the set of correctly classified descriptive segments as follows

$$Z_x = \begin{cases} F \cap D, & \text{for } x = 0 \\ F_{\{[x \cdot |D|]\}} \cap D, & \text{for } x > 0 \end{cases}. \quad (3.22)$$

3.5.4 Two-phase Approach Criteria

In Section 3.4.1 and 3.4.2 we have introduced various criteria for selection and sorting in our two-phase approach. In this section we evaluate these criteria to find the best



Figure 3.12: Dependent on the segmentation algorithm not every desired segment is available. The second row shows the desired segments for the first images of the bikes, graffiti and Leuven sequences (f.l.t.r.). The difference is visible in the third row that presents the manually chosen segments in segmentations using the standard algorithm.

performing one. This evaluation is done on the experimental dataset presented earlier. Images were segmented with the standard segmentation algorithm. Later we will show that the best performing criteria will also work well on images segmented using the alternative algorithm. We first establish the selection criterion before we find the sorting criterion.

Selection Criterion

The quality indicator for a selection criterion is its ability to choose as few as possible segments while preserving the descriptive segments. Formally, we require a selection criterion to have a high filter ratio $\frac{|P \setminus F|}{|P|}$ while letting pass the descriptive segments $\frac{|Z_0|}{|D|}$. For every selection criterion the average of these ratios are computed over the images in

the experimental dataset. Figure 3.13 summarizes the results of this computation.

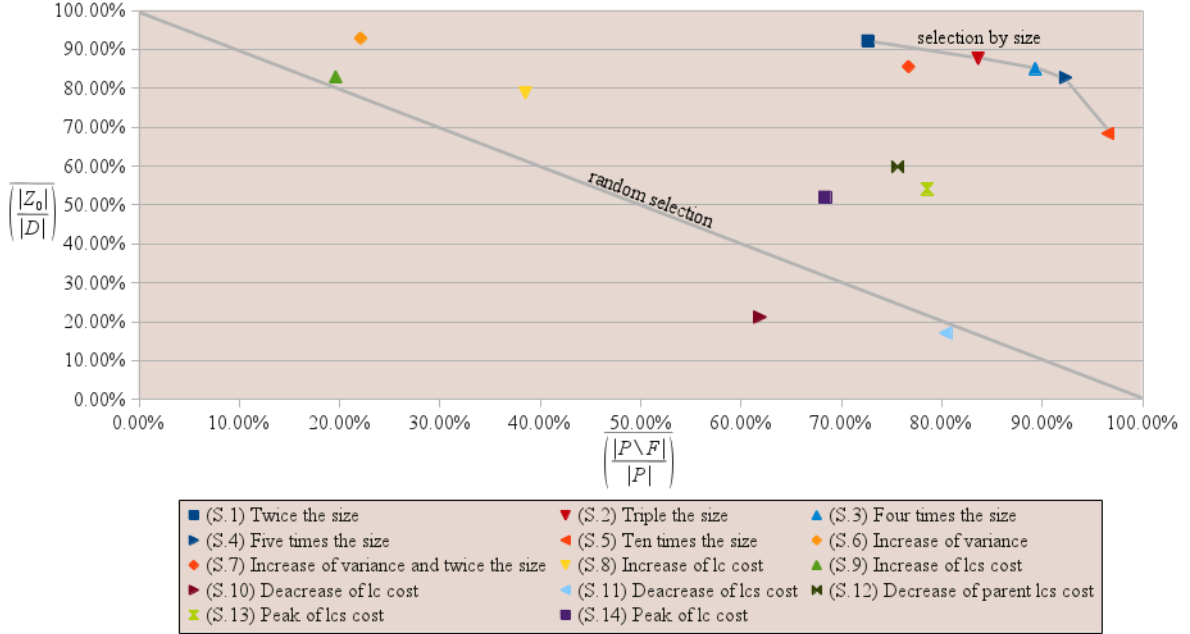


Figure 3.13: The selective performance for each criterion. While the range axis shows the averaged percentage of input segments discarded by the selector the domain axis indicates the average percentage of descriptive segments that passed the selector. The line from top right to bottom left shows the result of random selection.

We see most of the selectors perform better than random selection. Only two of the presented selectors are worse than random selection. These selectors clearly use the wrong criterion for selection and are not suitable for this purpose. We notice these two selectors were used as a counter argument to the increase of cost criterion. Even if increase of cost criterion is better than random selection we see that it is not able to provide good filtering properties. Not even half of the segments were filtered out by this criterion and, therefore, these selectors leave too many segments to our sorting. The best results of the cost argument is provided by the one using the peak theory. These filter out up to 80% of the segments, but do not preserve enough descriptive segments. We can summarize that selection by the cost criteria is not suitable. The result of the variance criterion is also not satisfactory and we have to discard this selector as well. The only selection criterion that performed well is the size criterion. The fact that descriptive segments are merged with much larger neighbours seems to be true. We can adjust this selector by changing the size multiplier. However, our results show that a combination of

a descriptive segment with a neighbour that is at least twice as big (triple the size) has the best performance. By increasing the multiplier the filter becomes more selective but at the expense of discarding descriptive segments. Additionally, we see that combining the variance and the size criterion does not improve our results. Therefore, we select the segments that match the condition $3 \cdot |S| < |S_p|$.

Sorting Criterion

With the established selection criterion we can continue with the evaluation of the sorting criteria. Sorting orders the segments that pass the selector by their descriptiveness. It is important that the manually selected descriptive segments appear at the top of the list. For every criteria $\overline{Z_x}$ is computed over all images in the dataset for $x = 0, 1, 2, 5, 10$. The ratio $\frac{\overline{Z_x}}{|D|}$ is the quality measure for the sorting criteria. The results for the different values of x are shown in Figure 3.14. We see all of the tested sort criteria perform quite well. None of the criteria orders the segments completely wrong. We notice that ordering by variance has the worst performance. Especially with increasing x it stays behind the other criteria. With small x sorting by cost outperforms sorting by size. With larger x sorting by size is more beneficial and becomes the best of the single sort criteria. Only the combination of variance, cost and size criteria is able to provide better results. It is also the better of the two combined criteria. These results clearly show that the used combination provide the best results. For this reason we use this criterion in our two-phase approach.

3.5.5 Performance Evaluation

With the selector and sorting criteria for the two-phase approach determined we can continue and compare it to the queue-based approaches. In the following evaluation the queue-based approaches are named by their method of outlier detection. The queue-based approaches start from the last merge and then follow the order of merges until they reach the first merge, or the required number of segments is found. The results in Table 3.5 are broken down by approach and the performance is shown for every image in the dataset. For the computation the standard segmentation algorithm was used.



Figure 3.14: The average percentage of the available segments in the intersection of the $x \cdot |D|$ most descriptive segments and the set of manually selected segments. The more segments are available the better the sort criteria is. Results are shown for each of the defined sort criteria.

[illegible]

Table 3.5 – continued from previous page

			Chauvenet			Deviation			Q Test			Two-phase		
I	$ P $	$ D $	$ F $	$ Z_2 $	$ Z_0 $	$ F $	$ Z_2 $	$ Z_0 $	$ F $	$ Z_2 $	$ Z_0 $	$ F $	$ Z_2 $	$ Z_0 $
boesch2	10387	8	799	2	2	1049	2	2	146	0	0	1661	3	7
boesch3	10564	9	786	0	0	1067	0	0	126	0	0	1679	3	8
boesch4	10260	7	707	2	2	1030	3	3	135	0	0	1615	3	5
graffiti1	9804	10	847	0	0	891	0	0	138	0	0	1624	4	8
graffiti2	9836	8	846	1	1	889	1	1	144	0	0	1655	2	6
graffiti3	9806	11	836	1	1	857	0	0	129	0	0	1623	2	7
graffiti4	9923	9	808	0	0	906	2	2	130	0	0	1623	4	7
graffiti5	10081	8	871	0	0	876	0	0	143	0	0	1768	1	7
graffiti6	10069	9	872	0	0	931	0	0	152	0	0	1689	0	7
home1	10089	13	731	1	1	1044	4	4	129	0	0	1626	7	12
home2	10104	16	697	3	3	1020	5	5	133	1	1	1601	13	16
home3	10095	16	727	2	2	1036	4	4	130	0	0	1614	10	15
home4	10072	15	736	5	5	1085	5	5	130	0	0	1600	7	13
home5	10101	18	724	3	3	1037	3	3	139	1	1	1563	14	17
house1	10213	9	727	1	1	1052	1	1	145	1	1	1654	2	8
house2	10246	10	782	1	1	1090	3	3	168	0	0	1609	1	9
house3	10248	18	748	2	2	1123	8	8	136	0	0	1633	15	18
house4	10178	13	745	2	2	1057	4	4	137	0	0	1635	5	13
house5	10183	19	761	2	2	1061	5	5	130	0	0	1662	15	19
Leuven1	10250	10	874	0	0	1043	1	1	158	0	0	1713	7	9
Leuven2	10244	10	870	1	1	1078	1	1	128	0	0	1561	5	8
Leuven3	10226	9	858	1	1	1072	1	1	136	0	0	1619	1	7
Leuven4	10223	10	820	0	0	1107	1	1	147	0	0	1666	5	9
Leuven5	10276	8	793	0	0	1145	0	0	129	0	0	1671	2	7
Leuven6	10300	6	870	0	0	1185	0	0	163	0	0	1650	3	5

Table 3.5: Detailed results of the filtering approaches using the standard algorithm. Given the segmentation P of image I and the manually chosen descriptive segments $D \in P$, let F be the set of filtered segments using one of the approaches. Then, the set of all matching segments is $Z_0 = F \cap D$, and the limited set is $Z_2 = F_{\{|2|D||\}} \cap D$.

Table 3.6 – continued from previous page

			Chauvenet			Deviation			Q Test			Two-phase		
I	$ P $	$ D $	$ F $	$ Z_2 $	$ Z_0 $	$ F $	$ Z_2 $	$ Z_0 $	$ F $	$ Z_2 $	$ Z_0 $	$ F $	$ Z_2 $	$ Z_0 $

Table 3.6: Results of the filtering approaches using the alternative segmentation algorithm. A subset of the experimental dataset, containing the most representative image of each sequence, was used. For comparison reasons the corresponding results using the standard segmentation algorithm are shown in the second row of every image. For an explanation of the symbols please refer to Table 3.5.

Result Analysis

The results in Table 3.5 and 3.6 are summarized in Figure 3.15 and 3.16. The first of the two Figures shows the average ratio $\frac{|F|}{|P|}$ for every approach using the two segmentation algorithms. This presents how many segments on average are considered descriptive by the approaches compared to the total number of segments in the partitions. There are differences between the two segmentation algorithms. While outlier detection in the standard segmentation algorithm is very restrictive compared to selection, the results are more similar to each other with the alternative segmentation algorithm. However, this is not indicative of the performance of an approach but outlines how important the quality of the segmentation is for the selection of descriptive segments.

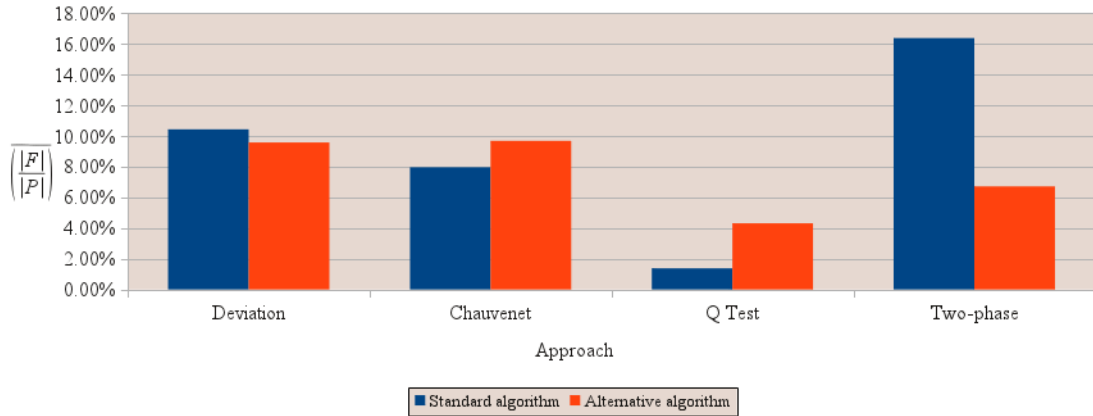


Figure 3.15: Ability of each approach to filter the segments. It does not indicate the quality of the filtered set.

For the visualization in Figure 3.16 we introduce “random” approach. This approach picks segments randomly and shows how unlikely it is to find the descriptive segments this way. We see that the two-phase approach outperforms the queue-based approaches by an order of magnitude. Between the queue-based approaches we do not notice huge differences. Each of the outlier detectors provide approximately the same result. It seems that the outlier detectors are able to identify descriptive segments when they appear at the end of the merge queue. It does not matter much if the intersection with the manually selected segments is taken on the first x or over all segments. This is quite surprising and implies that especially small descriptive segments cannot be identified by queue analysis. Positive is that queue-based analysis is better than random selection. A visualization for one result containing all approaches is presented in Figure 3.17. At all times the two-phase approach performs better than queue-based analysis. We see that with both segmentation algorithms the selection works out well. More than 80% of the manually selected descriptive segments are available in the sorting phase. The performance of the sorting is better with the standard algorithm. Here we find more than 40% of descriptive segments with $x = 2$. For the alternative algorithm we have to take at least $x = 5$ to get the same number of descriptive segments. Nevertheless, from thousands of segments the filter is able to choose a handful of descriptive segments. For the best performing image of each sequence, the chosen segments are presented in Figure 3.18. Please refer to Appendix A for the complete result.

3.5.6 Discussion

We have presented two approaches towards descriptive segment filtering. Both approaches are capable of identifying descriptive segments to some degree. While the queue-based approach did not meet our expectations the two-phase approach clearly performed much better. We have evaluated both approaches with two different segmentation algorithms. It is no surprise that results differ depending upon the quality of the segmentation algorithm. Apart from the “Q Test” approach all the other approaches perform not as well with the alternative algorithm as with the standard algorithm. Due to the region growing approach with the heavy dependency on appearance-based properties the alternative algorithm produces less balanced merge trees and, therefore, merges descriptive segments later than the standard algorithm. There are more large regions that grow incrementally which contribute to the increase in depth of the segmentation hierarchy. This is a disadvantage for the two-phase approach because selection will let

more non-descriptive segments pass. Since sorting is also dependent on the segment size, tiny segments are considered descriptive. The reason for worse performance of the queue-based approach with the alternative algorithm is similar. Due to large regions growing around the descriptive segments many of the descriptive segments get merged eventually in sequence. The queue-based approach is not able to detect outliers in these sequences and fails to find the descriptive segments. We can summarize that a segmentation with a balanced segmentation hierarchy using a reasonable size criteria is important for the quality of the result.

The queue-based approach did not perform very well. As can be seen from the results presented in Figure 3.17, the approach clearly misses the descriptive segments. Even different methods of outlier detection have not improved performance. When we study the reconstructed merge queue, humans are able to identify the outliers and can find the descriptive segments based on the merge queue. It seems, however, that these outliers can not be identified with the evaluated approaches. This may be due to the limited amount of samples and the unknown underlying distribution of the data. One difficulty is that we can not predict the behaviour of the cost. Even if its upper bound may be roughly exponential globally, locally it can behave completely different. This makes outlier detection extremely difficult. Additionally, the time of merge for each segment is required which not every algorithm supplies. Depending on the segmentation algorithm, the time of merge may even lead to wrong interpretation. An algorithm may successively merge several descriptive segments under high cost. In the merge sequence this will not be visible due to the higher cost of all merges. The analysis of the merge sequence is, therefore, highly dependent upon the segmentation algorithm and its merge decisions. Another issue is the identification of the descriptive segments. While our experiments have shown that from an identified outlier the smaller segment is more descriptive, we notice many tiny segments in our result. This does not only visually look worse but also makes further processing more difficult. All of these issues indicate that queue-based analysis as we used it in our work is not suitable for the purpose of descriptive segment filtering.

In comparison to the queue-based approach the two-phase approach performs better than expected. It seems to be highly sensitive to descriptive segments and is able to identify these unique features in an image. First reducing the set of segments and then ordering them seems to be an eligible approach to descriptive segment filtering. On average the first phase of the algorithm is able to discard more than 80% of the segments. We have shown that different selectors are possible, but some do not perform and we

have chosen the most successful one. The size difference of a segment to its parent seems to be the most compelling selection criteria. For sorting size seems to play a secondary role. The sorting criteria is based on variance and appearance-based similarity. Using this combination the manually chosen segments appear on the list of the most descriptive segments. With both, selection and sorting, the hierarchical structure of the segments play an important role. The judgement of the individual segments is based on the relationship of the segment and its parent. This information is available from all hierarchical segmentation algorithms and does not require knowledge about the implementation of the algorithm. We also see that results differ between the two algorithms. Especially, we notice some difficulties with the alternative segmentation algorithm. Tiny segments are clearly be considered too descriptive by our two-phase filtering. This is due to the hierarchical structure of the segmentation. In the standard algorithm size plays a more important role than in the alternative algorithm. And as a result, tiny segments - that are unique - survive for too long. When they get merged both selection and sorting consider these segments as too descriptive compared to our manually selected ground truth. This lowers the performance of the two-phase approach.

Analyzing the results of our two-phase approach we see some encouraging behaviour. Beside the segments that were manually selected, the algorithm presents segments that we also find descriptive in retrospect. Especially with the standard segmentation algorithm there are just a few wrong segments. Many of the other segments one can consider descriptive, but not all of them match with our manually selected segment. This influences the result because there might be a descriptive segment which causes the exclusion of a matching segment due to the limited size of the set. Furthermore, it can happen that a descriptive segment does not match a manually selected one entirely, and is therefore not counted as a match. Considering all these special cases the result of the two-phase approach is better than our experiments suggest. When we have a look at the results of the image sequences in Appendix A we see that there are descriptive segments that appear across several images in a sequence. We see our two-phase algorithm is capable of extracting the unique properties of images under varying conditions which satisfies our needs for descriptive segment filtering. In the succeeding chapter we use these descriptive segments in an approach to region matching.

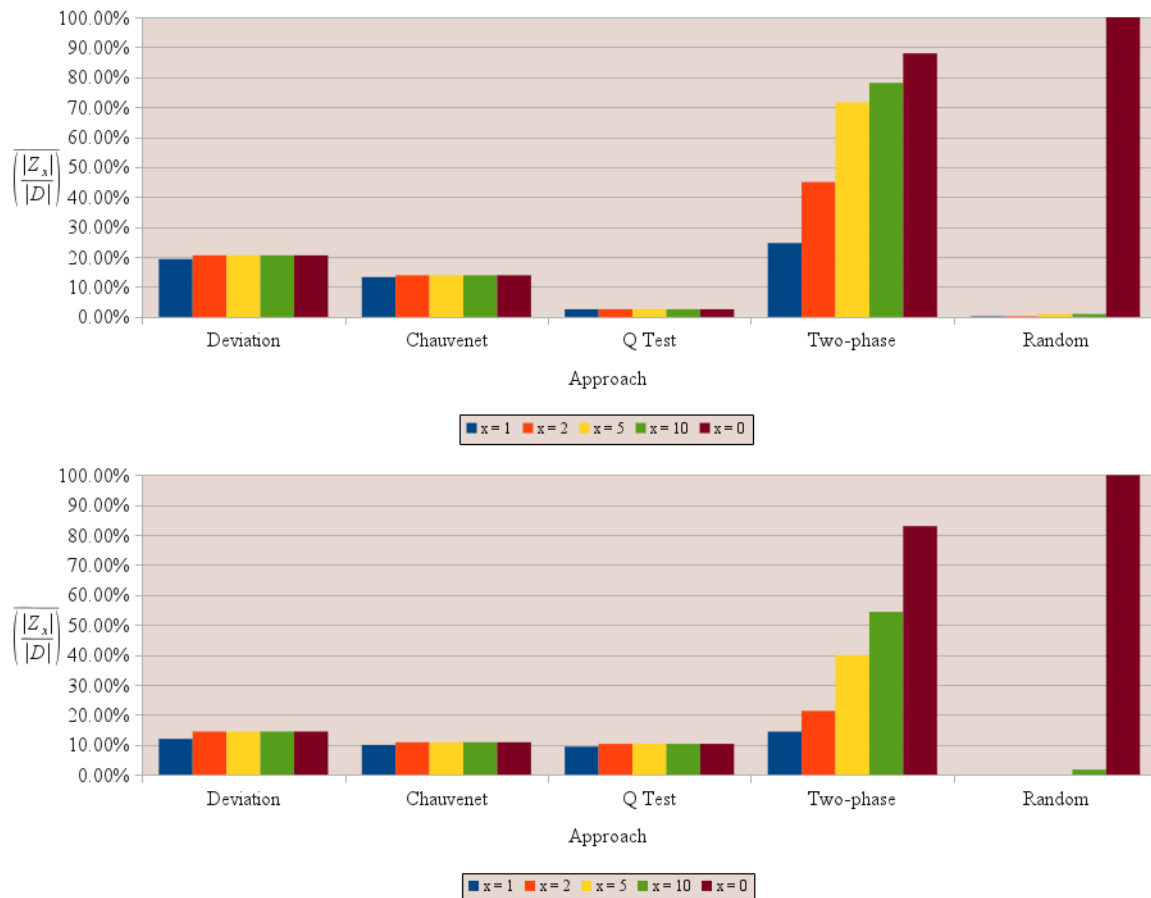


Figure 3.16: The upper of the two diagrams shows the average of the successfully identified descriptive segments using the standard segmentation algorithm. For each approach success rate is indicated with different set sizes. The lower diagram contains the same evaluation but uses the alternative segmentation algorithm.

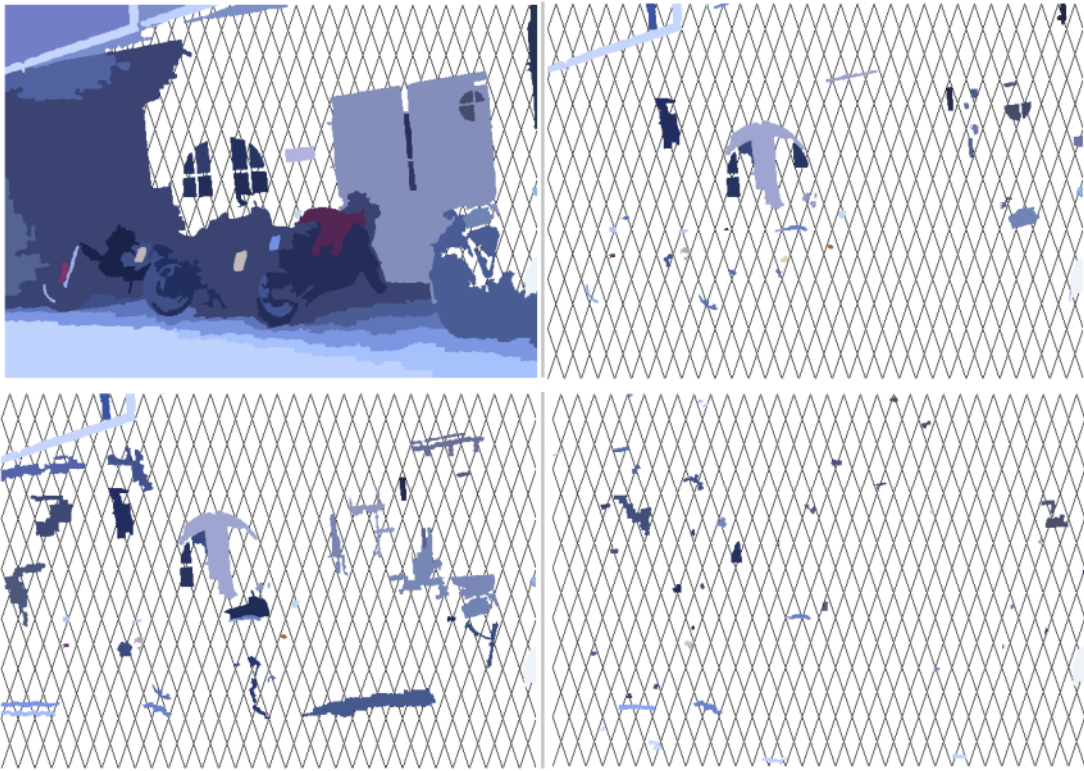


Figure 3.17: The visualized results of the bikes1 image using the standard segmentation algorithm. The 46 most descriptive segments are shown for the (f.l.t.r.) two-phase, deviation, Chauvenet and Q Test approach.



Figure 3.18: Best filtering result of each image sequence using the two-phase filter and the standard segmentation algorithm. Results are presented for (top-down) backyard4, bikes2, boesch1, graffiti4, home2, house3 and Leuven1. The columns contain (f.l.t.r.) the image, the manually selected segments, the descriptive segments and the intersection between the two. Please refer to Appendix A for all results.

Chapter 4

Matching Descriptive Segments

Image matching has been a popular topic in literature over the last decade [83, 93, 21, 82, 45, 79, 41]. A large number of approaches have evolved from the work on this field. The approaches differ on the type of images they aim to match. A common way to establish the image correspondence is by matching features. In our case features are the segments that passed the descriptive segment filter. The matching is simplified due to this relatively small set of features. We will present an approach to segment matching that is based on a correspondence measure between the segments of images. Additionally, we will use the hierarchical structure of the segments to refine the matches. We will evaluate this approach using images from our experimental dataset.

4.1 Reduced Hierarchy Construction

An unordered set of segments is provided by our descriptive segment filter. The algorithm presented in this section reconstructs the hierarchical structure of the segments independent of the order they appear in the input. This reduced hierarchy will provide information about the hierarchical dependency of segments to our descriptive segment matching approach. The reduced tree is composed of only the descriptive segments from the input and possible placeholders required to form a valid binary tree.

Let the tree of the segments be O and the reduced tree be R . Start the algorithm by marking the segments in O . Then, iterate over all descriptive segments and let the current segment of the iteration be S . Begin with a loop over the parents of S and call the current segment C . Stop the loop when C is a marked segment or when C is the root of O . In the later case remember C as root of R . In the loop search for all

marked children of C . If more than one child of C was found then C is promoted to be a placeholder. These nodes are required for R being a valid binary tree. They may be used later in addition to the descriptive segments. The process completes once every segment in O has been processed.

After the completion of the loop the root of R is known and all segments for R are marked in the tree O . To build the reduced tree R start with a recursive call using the root of R as S . Within the recursion search the marked children of S . If no child was found S is a leaf in the reduced tree T . At this point the recursion completes. In case S has one or two children a new segment S_r is created. Populate the left and right child of S_r by another recursive call using the children of S . There might be the situation that S has only one child and S is a placeholder in R , then S_r can be discarded and the child of S is to be returned. Otherwise S_r is returned by the recursion.

Once the reduced tree R is built it is traversed to fill in missing leaves. This completes the tree. It has the same structure as a segmentation tree from a hierarchical segmentation, but only contains the descriptive segments and the placeholders. These placeholders are ordinary segments from O that were not included in list of descriptive segments, but were necessary for the structure of R . Dependent on the usage these segments may provide further information about the descriptive segments.

4.2 Establish Segment Correspondence

Given two sets of descriptive segments F_p and F_q of the segmentations P and Q the task is to find the best corresponding segment $D_q \in F_q \forall D_p \in F_p$. Dependent on the size of the set F_q there might not be a correspondence for every D_p . Let us define the segment correspondence measure $M(S, T)$. This function is a measure for the similarity of two segments from different segmentations. With it the best matching segment in F_q for every $D_p \in F_p$ is identified. In Section 4.3 we will present two measures based on the visual properties of the segments. We assume there are two segments $S_p \in F_p$ and $S_q \in F_q$ that represent the same region in their segmentations P and Q and look for a correspondence measure that is maximal for this combination

$$M(S_p, S_q) = \max M(S_p, D_q), \forall D_q \in F_q \quad . \quad (4.1)$$

For every $D_p \in F_p$ the correspondence measure to all segments in F_q is computed. This list is sorted in descending order and the best correspondence for every D_p is taken into consideration for output. In case more than one segment in F_p links to the same

segment in F_q only the best correspondence is kept. The others are re-linked to their next correspondence in order. The whole process continues until the mapping is unique.

Let $n = \max(|F_p|, |F_q|)$ be the number of segments in the larger of the two sets F_p and F_q . The sort required to order the correspondences of a segment $D_p \in F_p$ to the segments in F_q takes $O(n \log(n))$ when using merge sort. This sort is done for every segment in F_p which adds another n to the worst case performance. Making the correspondences unique takes no more than $O(n^2)$. This leads to the conclusion that the worst case performance is $O(n^2 \log(n))$. In practise is not an issue due to the relatively small sizes of F_p and F_q .

4.2.1 Refine Correspondences

With the help of the hierarchical structure of the segments a refinement of the matching is possible. This refinement is performed on the reduced hierarchy constructed from the set of descriptive segments. Let us assume the previous matching found a correspondence of $D_p \in F_p$ to $D_q \in F_q$. In case a child of D_p is not matched with a child of D_q the correspondence is obviously not correct. In the image such a mismatch implies that a subpart of an object in one image is not within the object in the other. However, it may also be the case that the parent match was wrong. The correspondence measure is queried to decide whether the parent or the child match is to be removed. The match with the larger of the two correspondence measures is kept while the other one is discarded. The refinement completes when there are no more structural discrepancies between the images.

4.3 Segment Correspondence Measures

Given two segments S and T the correspondence measure $M(S, T)$ is a function of confidence that S and T is the same segment in two different segmentations. In case of image matching the two segmentations are computed from two different images. Viewpoint, illumination and camera settings, may have changed between the two images, leading to a distortion of the segments. We require the correspondence measure to deal with this situation and that the measure is comparable between different S and T . The value should be larger when two segments represent the same region than for two others. In combination with the previous algorithm the ultimate goal is to find the correct matches of segments between two segmentations. In this Section we will present two measures of segment correspondence. While the first one is based on the appearance of the segments

the second one uses a measure from probability theory.

4.3.1 Cost

In Section 3.2.1 we have presented a measure for the likelihood that two segments S_l and S_r are combined in a hierarchical segmentation at time t . This cost is composed of luminance, chrominance and size components that are not only of interest to the segmentation algorithm. They are a measure of similarity between the segments and are, therefore, suitable as correspondence measure. Let us compute the correspondence measure for the two segments S and T with their combined size $s = |S| + |T|$, their averaged luminance l and chrominance c in L*a*b* colour space and the constants ω

$$\begin{aligned} M(S, T) = & -\left(e^{-\frac{(S_{\text{luminance}} - l)^2 |S| + (T_{\text{luminance}} - l)^2 |T|}{2s\omega_{\text{luminance}}^2}} \right. \\ & + e^{-\frac{(S_{\text{chrominance}} - c)^2 |S| + (T_{\text{chrominance}} - c)^2 |T|}{2s\omega_{\text{chrominance}}^2}} \\ & \left. + \left(\frac{|S|}{s} \right)^{\omega_{\text{size}} - 1} \left(\frac{|T|}{s} \right)^{\omega_{\text{size}} - 1} \right) . \end{aligned} \quad (4.2)$$

The negation of the components ensures that the more similar two segments are the larger, meaning closer to zero, their cost is. This way the cost can be used as measure for segment correspondence in our algorithm.

4.3.2 Mutual Information

The traditional application of mutual information (MI) is in probability theory where it is a measure for the dependency of two random variables X and Y [73]. We like to adopt mutual information such that it becomes a correspondence measure between descriptive segments. Let us assume we know the individual probability distribution of these random variables P_X and P_Y and their joint distribution P_{XY} . Then the random variables drawn from their distribution define the marginal entropy

$$H(X) = - \sum_{x \in X} P_X(x) \log(P_X(x)) \quad (4.3)$$

$$H(Y) = - \sum_{y \in Y} P_Y(y) \log(P_Y(y)) \quad . \quad (4.4)$$

The joint distribution is the combined distribution of the two individual distributions P_X and P_Y . This defines the joint entropy as

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} P_{XY}(x, y) \log(P_{XY}(x, y)) \quad . \quad (4.5)$$

The mutual information of the two distributions is the one they share. By computing the difference between the two marginal entropies and the joint entropy this information is left. Formally, the mutual information is

$$MI(X, Y) = H(X) + H(Y) - H(X, Y) \quad . \quad (4.6)$$

Figure 4.1 shows a visual interpretation of the mutual information. Mutual information is a measure for the information X and Y share. The larger the mutual information, the more we know about Y by knowing X . The smaller the mutual information becomes the less we can know about the other random variable. With mutual information of zero the two random variables are independent from each other, meaning X does not share any information about Y and vice versa.

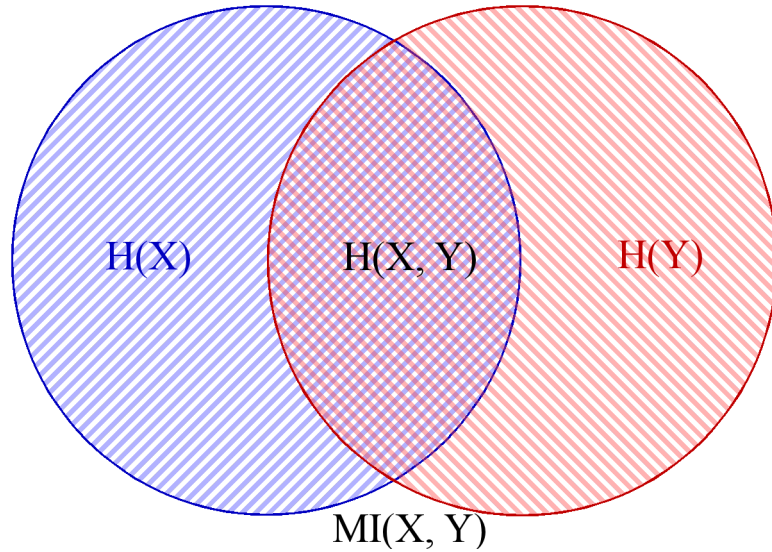


Figure 4.1: The joint entropy $H(X, Y)$ is the intersection of the marginal entropies $H(X)$ and $H(Y)$. The more the marginal entropies intersect the more we can say about one random variable knowing the other. Mutual information $MI(X, Y)$ measures the dependency between these two random variables.

Mutual Information in Alignment

Mutual information has successfully been applied to many topics, and in particular in communication theory where it measures the information shared between a sender and receiver of a message [88, 22, 79]. Of special interest to our problem of image matching is its application in image alignment. Let I and J be two grey scale images of similar

content, for instance an MRI image of a head like it occurs in medical imaging. Images in the context of probability are like distributions where random variables can be drawn from. Values appear depending on the content of the image. The probability of a value q to appear is commonly taken to be by the number of occurrences in the image, so the probability distribution is given by

$$P_I(q) = \sum_{x,y \in I} \begin{cases} 1, & \text{for } (x,y) = q \\ 0, & \text{otherwise} \end{cases} . \quad (4.7)$$

To construct the marginal entropy of an image the probability distribution has to be normalized by the number of samples in an image. Furthermore, let Q be the set of all possible values in an image, then the marginal entropies of the two images I and J are

$$H(I) = - \sum_{q \in Q} \left(\frac{P_I(q)}{|I|} \right) \log \left(\frac{P_I(q)}{|I|} \right) \quad (4.8)$$

$$H(J) = - \sum_{q \in Q} \left(\frac{P_J(q)}{|J|} \right) \log \left(\frac{P_J(q)}{|J|} \right) . \quad (4.9)$$

Beside the marginal entropies the joint entropy is required to compute the mutual information the two images I and J share. When we assume that the images completely overlap the joint entropy is the probability that given a value in I another value in J occurs

$$P_{IJ}(q,r) = \sum_{x,y \in I} \begin{cases} 1, & \text{for } I(x,y) = q \text{ AND } J(x,y) = r \\ 0, & \text{otherwise} \end{cases} . \quad (4.10)$$

Like the marginal entropies the joint entropy has to be normalized by the number of samples. For the two images I and J the joint entropy is

$$H(I, J) = - \sum_{q \in Q} \sum_{r \in Q} \left(\frac{P_{IJ}(q,r)}{|I|} \right) \log \left(\frac{P_{IJ}(q,r)}{|I|} \right) . \quad (4.11)$$

In case the two images do not completely cover each other only the overlapping portions of the images are considered for both the marginal and joint entropy. According to Viola and Wells [88] and Collignon et al. [22] maximizing the mutual information finds the optimal alignment of two images.

Normalized Mutual Information

A disadvantage of image alignment by mutual information is its dependency on the overlap of the images [79]. This can be seen when the alignment of two identical images

only differs in their overlap of the background region. Because the more of the background region overlaps the more probable its value is, meaning the mutual information increases. Figure 4.2 visualizes this influence of the background overlap for an example image. We see that by maximizing the mutual information we will favour the alignment that has the smallest overlap. This is not desired and a measure of mutual information that is independent of the overlap is required. In his approach to medical image registration Studholme [79] proposed the normalized mutual information, which is

$$Y(I, J) = \frac{H(I) + H(J)}{H(I, J)} \quad . \quad (4.12)$$

The ratio of the marginal entropies to the joint entropy compensates for the change in uncertainty when the overlap changes. This makes $Y(I, J)$ independent of the overlap as seen in Figure 4.2. The optimal alignment is found at the maximum of $Y(I, J)$. At this point the joint entropy is minimized with respect to the marginal entropies.

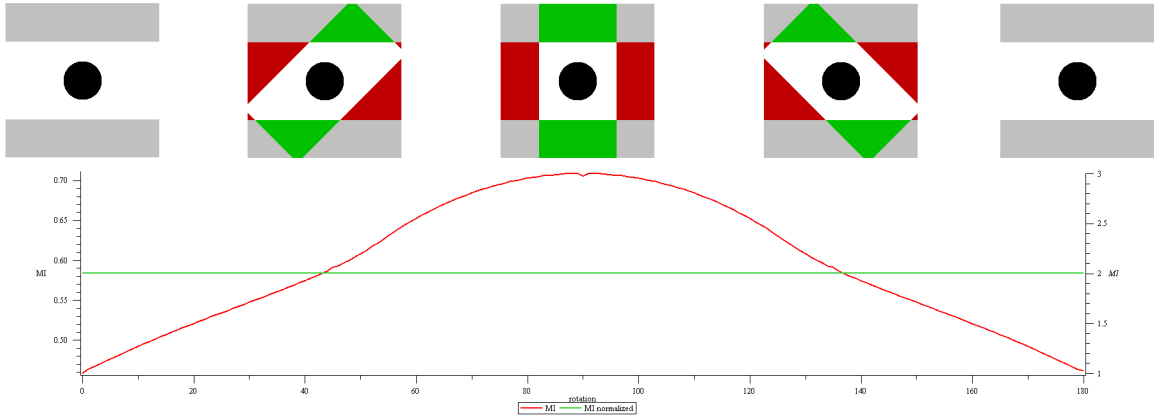


Figure 4.2: Mutual information of two rectangles for changing rotation. Both rectangles are of the same size and contain a circle in their centre. Rotation of one rectangle changes the overlap of their background (white). While the traditional mutual information is dependent on the background overlap the normalized mutual information is not.

Prepare Segments for Matching

In our approach to image matching descriptive segments from two images are compared. Mutual information can only be applied to compute the entropies of two segments when they overlap. In our segmentation the segments may appear at different positions which makes a transformation to a coordinate system where their centre of gravity is at the

origin required. This way same segments have a large overlap when their distortion, that originates from viewpoint changes between the two images, is small. A change in viewpoint affects all other segments as well and we expect mutual information to compensate for the mismatch between the same segments. Figure 4.3 contains examples of matches between the segments of two images. Even if the corresponding segment is distorted we see it still overlaps to a great extend.

The segments and their underlying pixel values are available in RGB colour space. For mutual information this colour space is too large, which makes a transformation of colour values to grey scale required. We use the simplified transformation of the HDTV-RGB standard (ITU-R BT.709) to convert each colour channel to a unified luminance value [75]

$$Y(R, G, B) = 0.2126 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B \quad . \quad (4.13)$$

We are aware of the error introduced by this conversion, but found it accurate enough for the purpose of segment matching. The correct luminance value L^* can be computed by the conversion from the RGB to the $L^*a^*b^*$ colour space as described in Section 3.2.

Mutual Information for Segment Correspondence

When mutual information is used in image alignment the two images overlap to a great extend. This is not necessarily the case with two segments. In Figure 4.3 we see that for many segments the overlap is quite small. This will lead to wrong interpretation as we can see in Figure 4.5a. Even if the two images perfectly match at $tx = 160$, mutual information also suggest perfect matches at $tx = \{80, 120, 200, 240\}$. This is not surprising since mutual information is a measure for the information shared only between the overlapping regions of the images. Furthermore, it makes no difference when the white pixels correspond to the grey ones, it is only important that the correspondence is uniform. This is, however, a disadvantage when we compare segments that overlap each other only partially. The shared information between them might be higher in comparison to two segments with more overlapping pixels. We overcome this issue by two extensions to mutual information. First, we introduce a correspondence value for the non-overlapping pixels. At the positions where pixels from either segment do not overlap pixels from the other segment a background value is inserted. This background value is distinct from all possible segment values and extends the value ranges of the marginal and joint entropies. The effect of this background value is seen in Figure 4.5b. In this example the only perfect match is at $tx = 160$. The other peaks became smaller due to

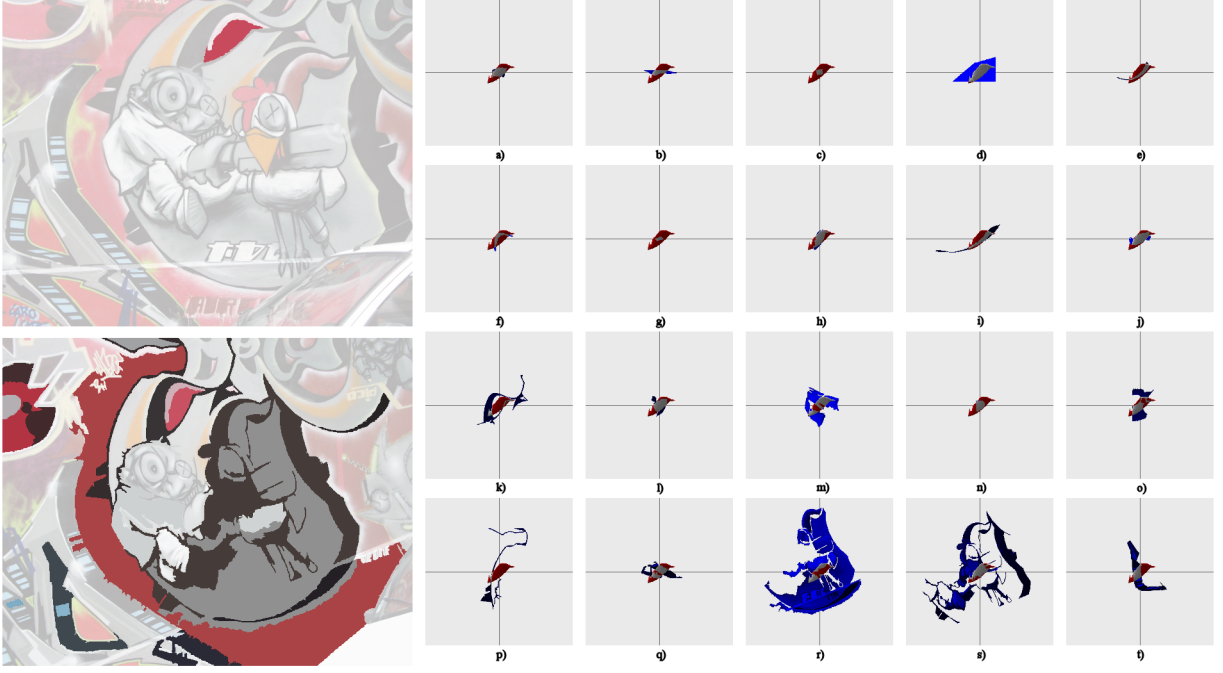


Figure 4.3: Match by mutual information from one segment of the graffiti1 image (top) to 20 segments of the graffiti2 image (bottom). The segments are translated to a new coordinate system where their centre of gravity is at the origin. Then, they are converted to grey scale. For visualization purposes the segment of the graffiti1 image utilizes the red colour channel while the segments of the graffiti2 image use the blue colour channel. At the pixels where they overlap the grey scale value of the graffiti1 segment is shown. Mutual information is used to rate each correspondence of the graffiti1 segment to the graffiti2 segments. There is great difference in the overlap of the segments which should influence mutual information. The correct correspondance is shown in h).

the mismatch of the background regions. We see, however, that with only a few pixels overlap the background regions contribute too much to the mutual information. For this reason we propose a direct dependency of the mutual information to the overlap. This second extension normalizes the mutual information by the overlap of the two segments. The ratio for the normalization is given by the division of the size of the overlap by the area the two segments cover. While this penalizes little overlap it favours matches where the segments completely cover each other. In Figure 4.5c we see how this changes the mutual information for our example. The emphasis now is clearly on the perfect match in expense to smaller overlaps.

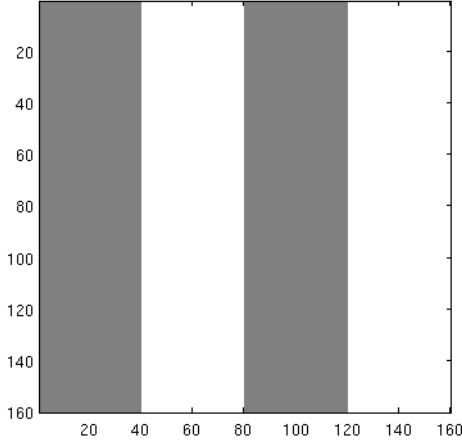


Figure 4.4: Stripe image used to evaluate mutual information as correspondence measure.

4.4 Score Matches

Our approach to image matching finds the correspondences between the segments of two images showing the same scene under varying conditions. It is uncertain whether or not a match is correct, meaning the two segments correspond to the same region in both images. For our experimental evaluation, however, we need to know the number of correct matches. Only this way we can evaluate the approach and choose the best correspondence measure.

The evaluation can be done manually when there are only a few matches to consider. For each correspondence the segments in both images are compared. In case they both represent the same region in the images the match is correct. Due to viewpoint changes it may not always be obvious that the two segments represent the same region. Furthermore, there are situations where one segment only partially maps to the segment in the other image. In this case the match is considered correct when the smaller of the two regions cover the other one to at least $\frac{3}{4}$.

Manual evaluation is not suitable for larger sets of matches because it becomes prone to errors. For a machine based categorization of the matches the transformation between the images is required. Under the assumption of a pinhole camera model with constant focal length the relation between two planar surfaces in the images is given by a homography matrix [40]. A homography matrix is a 3x3 matrix that provides a linear transformation of the points in the plane. This linear transformations includes rotation,

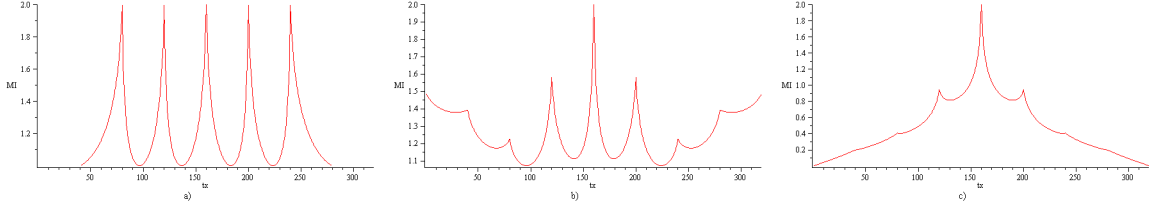


Figure 4.5: Mutual information is not directly suitable as segment correspondence measure because it depends on the overlap of the regions. To visualize the behaviour of our improved mutual information the stripes image in Figure 4.4 is shifted over itself. At a translation of $tx = 0$ the images are next to each other having no overlap. Then, the left image is shifted along the x axes until it is on the right after $tx = 360$. Curve a) contains the original mutual information, b) the modified entropies including the non-overlapping pixels and c) the overlap normalization used as segment correspondence measure.

scale and skew. The translation component can be incorporated in the homography matrix because of the homogeneous 3×3 matrix for 2D points in a plane and no separate addition of a translation vector is needed. Let us assume we are given the homography matrix H . A point

$$P_i = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (4.14)$$

in the planar surface of I is mapped to its counterpart in image J by the matrix multiplication

$$P'_j = \begin{bmatrix} \omega' x_j \\ \omega' y_j \\ \omega' \end{bmatrix} = H \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (4.15)$$

To get the point P_j in the plane of J the point P'_j is normalized by ω . With the help of the homography all pixels from a segment in I can be mapped to image J . When the transformed pixels correspond with a descriptive segment from image J the correct match was found. Like with visual correspondence we do not require the segment to completely cover each other. A coverage of $\frac{3}{4}$ of the larger segment is sufficient for a match to be considered correct. The correct matches will be our ground-truth for the performance evaluation of our approach.

4.4.1 Homography Computation

The homography matrix is computed using the combined SIFT and RANSAC approach [54, 40]. In both images SIFT descriptors are constructed. From these descriptors an initial match is computed by the algorithm presented by Lowe [54] which rejects matches that are too ambiguous. In an iterative refinement RANSAC then optimizes the correspondences between the descriptors by ignoring descriptors that do not match the model being considered. The homography matrix is constructed at the start of each step of the iteration. The best homography is the one that explains the most correspondences. For the computation of the homography we use the VLFeat package from Vedaldi and Fulkerson [86].

4.5 Results

The evaluation of our region matching approach is based on the same experimental dataset as used in the descriptive segment filter analysis. Details about the dataset are given in Section 3.5.1. The dataset is arranged in seven sequences of four to six images. The images are ordered by similarity, meaning the visual difference between successive images is the smallest in the set. We use only the successive images for matching. For a sequence of n images there will be $n - 1$ matches to compute. Let I be the preceding image of J in the sequence. The descriptive segments are computed for both images using the two-phase filtering approach. The size of the set of descriptive segments is chosen to be 30 for I and 50 for J . Selecting more descriptive segments in J ensures that for each descriptive segment in I a (not necessarily correct) correspondence can be found. It also reduces the risk of the correct corresponding descriptive segments not being present in J . The results of matching are very dependent on the filtering. Matching can only be successful when the descriptive segments are present. Wrong matches are unavoidable because there will always be descriptive segments in I that have no corresponding segment in J .

For automatic evaluation the approach presented earlier is used to construct the homography matrix H between two planar or mostly planar images I and J . Typically, a large number of SIFT features is found in our experimental images. Not all of them correspond well between images as seen by the number of in- and outliers when constructing the homography matrix. A robust homography matrix, that explains the transformation of most SIFT features, has many inliers and just a few outliers. Beside the number of

in- and outliers the determinant of the homography matrix provides information about the computed transformation [49]. A determinant of 1, as given by the identity matrix for instance, implies that there is no transformation between the images. The larger the difference of a determinant to one is, the larger the transformation becomes. Because the transformation between our images is small, an especially large transformation implies that the estimated homography is incorrect. The number of ground-truth segments found for the two images confirm this as well. By going back to the results of descriptive segment filtering we see that there are matching descriptive segments in most sequences. This leads us to the assumption that there is only no ground-truth correspondence between the images when the homography matrix is incorrect.

When it comes to planar surfaces the sequences in our dataset are very different from each other. There are sequences that contain planar images, images with a dominant plane and images with a small shift of camera position. All of these sequences can be evaluated automatically. The sequences of planar images and of images with a dominant plane can be evaluated by computing the homography between two images I and J of the sequence. The images with a small shift by estimating and applying the shift without any parallax. We can not evaluate sequences that do not fit into these categories, like the backyard and the boesch sequence. The planar surfaces contained in images of these sequences are not large enough or are cluttered with many fore- / background objects. We will, however, evaluate every sequence visually regardless of image planarity. We will summarize the results of this evaluation at the end of this section. Before we do so we will also discuss the influence of refinement on our results.

4.5.1 Planar Images

The approach presented earlier is used to compute the homography matrix H between planar images I and J . In our dataset only the graffiti scene is planar. Images from this sequence contain many matching feature points which results in a stable homography matrix. This homography matrix is used to compute the ground-truth correspondences between the two sets of descriptive segments. The following table contains the results of this computation.

I	J	Features I/J	In-/ Outliers	H	$\det(H)$	GT
graffiti1	graffiti2	1182/1432	579/52	8.812e-01 3.134e-01 -3.19e+01 -1.83e-01 9.365e-01 1.231e+02 2.495e-04 -2.29e-05 1.000e+00	0.902	6
graffiti2	graffiti3	1432/1362	547/35	6.723e-01 -5.31e-01 2.646e+02 5.978e-01 8.650e-01 -1.49e+02 1.967e-04 -6.37e-05 1.000e+00	0.853	4
graffiti3	graffiti4	1362/1389	636/31	5.633e-01 8.524e-01 -7.58e+01 -6.99e-01 7.754e-01 2.966e+02 1.377e-04 3.546e-05 1.000e+00	1.072	1
graffiti4	graffiti5	1389/1339	506/37	6.461e-01 -3.91e-01 2.392e+02 5.729e-01 7.725e-01 -9.68e+01 8.281e-05 -9.83e-05 1.000e+00	0.692	3
graffiti5	graffiti6	1339/1317	593/26	7.086e-01 -6.19e-01 2.368e+02 5.544e-01 8.791e-01 -1.23e+02 1.514e-04 -3.31e-05 1.000e+00	0.939	2

Table 4.1: Based on the SIFT features of two planar images I and J the homography matrix H is computed using RANSAC. A large number of inliers with just a few outliers is desired for a robust homography matrix. The determinant of the homography matrix is a measure for the transformation between the images. The ground-truth correspondences (GT) is established between the sets of descriptive segments using the homography matrix.

We see by the large number of inliers that the homography matrices were successfully constructed. This is not surprising because the wall, where the graffiti is drawn at, is a perfect planar surface. Due to bad filtering there are only a few ground-truth correspondences found. This is why we cannot expect many matches.

4.5.2 Images with a Dominant Plane

Images having a dominant plane may also be suitable for computing a homography matrix H . Such a dominant plane should cover most areas in the images because the

Table 4.2 – continued from previous page

I	J	Features I/J	In-/ Outliers	H	$\det(H)$	GT
-----	-----	-------------------	------------------	-----	-----------	----

Table 4.2: Using the same method as in Table 4.1 the homography matrix is computed between images having a dominant plane.

Both, the home and house sequence have a dominant plane for which the homography can be computed. The first correspondence of the home sequence could not be established correctly. Because there was no ground-truth match found we exclude it from evaluation. The other three relations of this sequence are reliable enough and are included in the evaluation. The results of the house sequence have to be excluded completely from evaluation because the homography matrices are not stable enough. There are issues with the first and third correspondence as seen by the small number of inliers. The other two homography matrices are more reliable, but there was almost no ground-truth match established. We will judge these correspondences only visually.

4.5.3 Images with a Small Shift

A small change of camera position between two images can be estimated by analyzing corresponding feature points. An average over the horizontal and vertical shift of feature points provides a sufficient approximation of the movement. In practise, however, we apply the same approach like before and construct the homography matrix H between images. This is possible because images taken from almost the same viewpoint do not need to be planar for computing the homography matrix H . It appears to the algorithm as they have no depth at all, meaning everything lays on a single plane. The last column of the homography matrix contains the translation vector V that describes the change of camera position. The following table contains the translation vector for the applicable sequences.

I	J	Features I/J	In-/ Outliers	V	GT
bikes1	bikes2	1119/1351	757/25	$[1.346\text{e}+01, -1.78\text{e}+01, 1.000\text{e}+00]^T$	7
continued on next page					

Table 4.3 – continued from previous page

I	J	Features I/J	In-/ Outliers	V	GT
bikes2	bikes3	1351/1482	978/28	$[-1.47\text{e}+01, -2.25\text{e}+00, 1.000\text{e}+00]^T$	7
bikes3	bikes4	1482/1675	1051/21	$[-3.91\text{e}+00, -6.78\text{e}+00, 1.000\text{e}+00]^T$	10
bikes4	bikes5	1675/1865	1207/21	$[1.102\text{e}+00, 2.002\text{e}+00, 1.000\text{e}+00]^T$	9
bikes5	bikes6	1865/1744	1227/30	$[-3.67\text{e}+00, -2.67\text{e}+00, 1.000\text{e}+00]^T$	8
Leuven1	Leuven2	869/879	616/22	$[3.367\text{e}+00, -2.35\text{e}+00, 1.000\text{e}+00]^T$	11
Leuven2	Leuven3	879/842	608/25	$[-6.17\text{e}-02, -9.80\text{e}-01, 1.000\text{e}+00]^T$	13
Leuven3	Leuven4	842/828	580/29	$[2.584\text{e}+00, -3.57\text{e}+00, 1.000\text{e}+00]^T$	18
Leuven4	Leuven5	828/838	502/29	$[-5.96\text{e}+00, 1.191\text{e}+00, 1.000\text{e}+00]^T$	11
Leuven5	Leuven6	838/871	486/26	$[1.291\text{e}+00, -6.18\text{e}+00, 1.000\text{e}+00]^T$	10

Table 4.3: Estimating the change of camera position for images with a small shift does not require the construction of a homography. For practical reasons, however, we use the same approach like for planar images. This is possible because images with a small shift look like planar images to the algorithm that have no depth at all. From the established homography matrix H we can extract the translation vector V describing the change of camera position. Based on this vector we compute the ground-truth matches (GT) between the sets of descriptive segments.

The images of the bike and Leuven sequences in our experimental dataset were taken from almost the same viewpoint. Beside this minor change of camera position there is no other transformation between these images. The large number of SIFT features brought in correspondence by RANSAC suggest that the homography matrices are stable, and so are the translation vectors V . The many ground-truth correspondences established between descriptive segments of images in these sequences confirm its stability.

4.5.4 Evaluation

The construction of the homography matrix is not possible for some sequences in our experimental dataset due to the absence of a planar surface in the scene. This does not allow us to construct the set of ground-truth matches and, therefore, we will be unable to evaluate these sequences automatically. The correct matches can, however, be manually

counted by visual comparison of the segments. We use such a manual approach for all correspondences in addition to the homography based evaluation. Manual counting of correct matches can be difficult, especially when the viewpoint changed between images. To provide accurate results the set of matches is reduced to the best 12. The matches are selected according to the correspondence measure $M(S, T)$. If refinement was used the matches were chosen after it. Both correspondence measures are used in the evaluation. The results are contained in the following table.

I	J	Refined	Cost			MI		
			Matches	Correct	Visual	Matches	Correct	Visual
backyard1	backyard2	no	30	-	0	30	-	1
		yes	30	-	0	28	-	1
backyard2	backyard3	no	30	-	3	30	-	3
		yes	29	-	3	29	-	3
backyard3	backyard4	no	30	-	0	30	-	0
		yes	30	-	0	28	-	0
bikes1	bikes2	no	30	2	3	30	6	8
		yes	29	2	3	26	5	7
bikes2	bikes3	no	30	6	7	30	7	9
		yes	29	6	7	26	7	9
bikes3	bikes4	no	30	9	8	30	8	8
		yes	26	7	7	20	7	8
bikes4	bikes5	no	30	9	9	30	9	10
		yes	29	8	9	26	9	10
bikes5	bikes6	no	30	7	9	30	8	11
		yes	24	6	9	25	7	10
boesch1	boesch2	no	30	-	0	30	-	0
		yes	30	-	0	25	-	0
boesch2	boesch3	no	30	-	3	30	-	5
		yes	26	-	3	27	-	5
boesch3	boesch4	no	30	-	0	30	-	1
		yes	30	-	0	29	-	1
graffiti1	graffiti2	no	30	5	4	30	6	4

continued on next page

Table 4.4 – continued from previous page

I	J	Refined	Cost			MI		
			Matches	Correct	Visual	Matches	Correct	Visual
		yes	29	2	2	28	18	10
Leuven4	Leuven5	no	30	1	2	30	11	11
		yes	29	1	2	29	11	11
Leuven5	Leuven6	no	30	0	2	30	10	9
		yes	30	0	2	30	10	9

Table 4.4: The results of the segment matching approach for two correspondence measures. For each combination of subsequent images in the sequences the number of correct matches and the visual evaluation is shown. Using the refinement introduced in Section 4.2.1 the number of matches is reduced based on the hierarchical structure.

4.5.5 Analysis by Homography

The homography matrix, or its translation vector respectively, can only reliably be constructed for the bikes, graffiti, home and Leuven sequences. Figure 4.6 summarizes the results of Table 4.4 by showing the average number of correct matches per sequence. For comparison the number of ground-truth matches is included as well. Because evaluation is done against this ground-truth our approach cannot perform better than it. Assuming the homography matrix is accurate the number ground-truth matches equals the number of corresponding segments in the set of descriptive segments of I and J . For example, among the bikes sequence there are on average around 8 descriptive segments that correspond between two images of the sequence.

The influence of the correspondence measure is large for some sequences. It is no surprise that the cost measure is less suitable as correspondence measure than mutual information. The cost measure is most suitable as a dissimilarity measure for segmentation algorithms, but it is too unstable when it comes to rate the correspondence. The result of the Leuven sequence in comparison to the bikes sequence is surprising. There is no change in viewpoint of both sequences, in one brightness and in the other blur is the varying condition. Not only does brightness have a direct influence on the cost, it also changes the appearance of the segments because borders in the image vanish. This has a double effect on the size of the segments which do no longer match very well.

The preferred correspondence measure is clearly mutual information. For all the sequences it nearly reaches ground-truth level, meaning most of the available correct matches were established. It seems not only able to cope with changes in brightness and blur and it also handles viewpoint changes quite well. The graffiti sequence does, unfortunately, not provide many ground-truth segments to be meaningful. Certainly, the distortion of the segments is less in the home sequence compared to the graffiti sequence, but we can see that mutual information is able to handle both well.

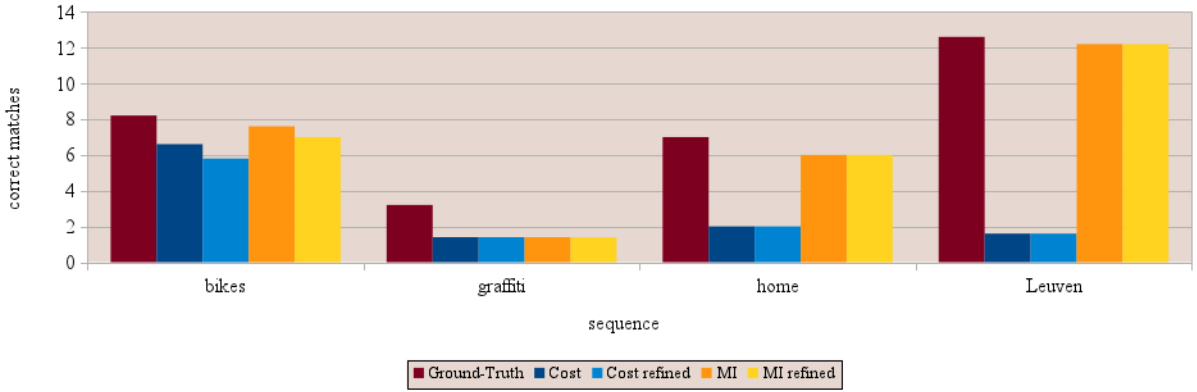


Figure 4.6: Evaluation of the correct matches based on the computed homography. The correspondence measures have a great influence on the matching approach. Mutual information is able to find most of the ground-truth matches and is the preferred correspondence measure.

4.5.6 Visual Conformance

Even when no homography is available we can always manually count the number of correct matches. This visual evaluation is done on the best 12 matches and the result is summarized in Figure 4.7. These results confirm that mutual information is the better of the two correspondence measures. Except for the graffiti sequence it clearly outperforms the cost measurement. As we see our approach performs much better on the bikes, home and Leuven sequences than on the backyard, boesch, graffiti and house ones. For almost every correspondence, however, it is able to establish at least one correct match. Only between 6 of the 29 computed correspondences no correct match is identified. In the bikes and Leuven sequence almost every match is correct.

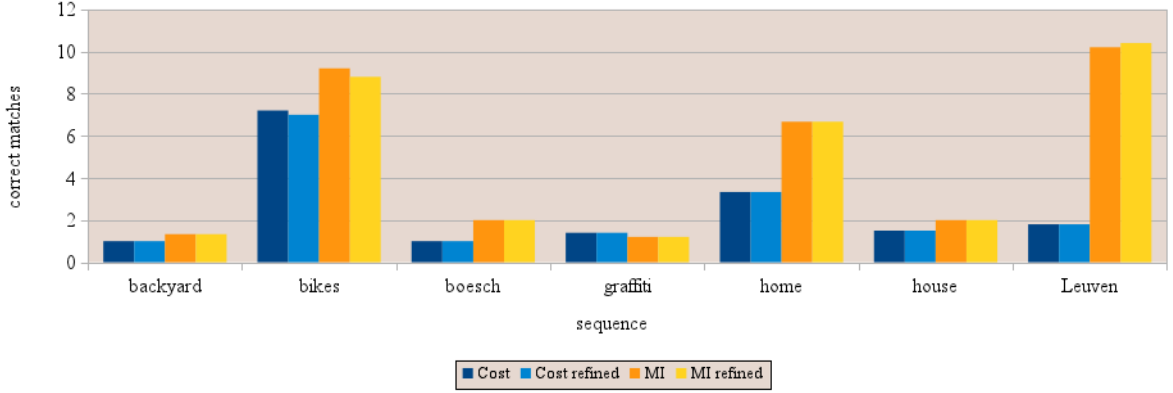


Figure 4.7: Average number of correct matches per sequence by visual evaluation. The mutual information correspondence measure is preferred because it outperforms the cost measure. Our approach performs well on the bikes, home and Leuven sequence but fails on the backyard, boesch, graffiti and house sequences.

4.5.7 Refinement Influence

After our approach established the correspondences between the descriptive segments of both images, refinement is used to validate these matches against the hierarchical structure of the segmentation. When a match conflicts with the structure of the segmentation it is removed. This reduces the set of available matches. From the previous results presented in Figure 4.6 and Figure 4.7 we see that refinement has only minor influence on the number correct matches. This is desirable since the goal of refinement is to remove obviously wrong matches. In Figure 4.8 the average number of available matches after refinement is shown. We see that refinement approximately removes between 10% and 20% of the matches for the mutual information correspondence measure. Therefore, it improves the ratio of the correct to the incorrect matches.

4.5.8 Summary

Our approach to descriptive segment matching has shown moderate results. Performance is very dependent on the input as we have seen by the differences between the sequences. While for some sequences the results were satisfactory for others they were not. At least there were only a few image pairs where no correct match was identified. This is encouraging because our approach was able to find matches even if some images were difficult to match. This difficulty was seen at the construction of the homography where

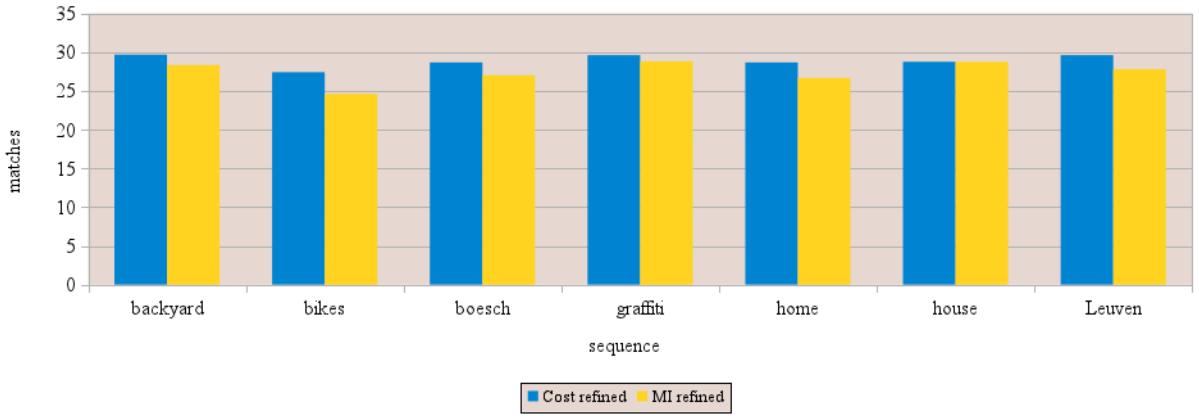


Figure 4.8: Refinement is able to remove wrong matches by validation of the hierarchical segmentation structure. From the 30 matches provided by our segment matching approach about 10% to 20% are removed. As seen by the results in Figure 4.6 and Figure 4.7 this has only minor influence on the number of correct matches, meaning refinement has successfully removed incorrect matches.

the combined SIFT and RANSAC approach was not able to find a homography for images even if they contain a planar surface.

The mutual information correspondence measure has clearly outperformed the cost-based measure. Among all sequences it has shown equal or better performance and the difference between the measures is large for some sequences. We conclude that cost, in contrast to mutual information, is not a suitable correspondence measure for segments from different images. Using hierarchical information to remove wrong matches reduces the set of matches without major influence on the number of correct matches. This refinement is independent on the content of the images and can be used for every sequence. Because results are very different let us summarize them per sequence.

- backyard: The backyard sequence is probably the most difficult sequence in our dataset. Beside the change of viewpoint the windows cause reflections. In the foreground are plants which lead to tiny segments. This makes it difficult for filtering to find the descriptive segments and matching has to deal with similar segments that may not belong to the same region in the two images.
- bikes: Both filtering and matching are able to cope well with the added blur between the images. This sequence has shown decent results with our approach to image matching.

- boesch: The building in this sequence does not provide an enough planar surface for the homography to be constructed. The difficulty of this sequence is especially due to unclear segments borders caused by the windows and their shutters. In the front of the building are cars which complicate the scene additionally. We notice, however, that correct matches of descriptive segments are found on the cars. The building itself does not seem to be very suitable for matching.
- graffiti: The graffiti sequence was previously used in literature as an example of feature matching. The combined SIFT and RANSAC approach to compute the homography works very well for this sequence. Nevertheless, our approach fails to find correct matches. There is only a small ground-truth caused by a distinct set of descriptive segments. Due to the similarity of some of these descriptive segments our approach to segment matching is not able to establish correct correspondences. The huge viewpoint change complicates matching even more. This leads to only a few correct matches among the images of this sequence.
- home: In comparison to the graffiti sequence the viewpoint change is smaller in this sequence. Furthermore, it is not subject to rotation. The larger set of ground-truth matches indicate that descriptive segment filtering is better with this sequence. Most of these correct matches were established by our matching approach which shows its ability to handle changes in viewpoint.
- house: Even if this sequence is similar to the home sequence its performance is worse. The reason for this is not caused by the different image sizes, because positions are not evaluated in our approach. It is more a result of the similarity of the segments. The house of this sequence has many, almost equal sized, windows and window shutters. This lets filtering choose different windows or shutters and matching establish correspondences between those. This may also be the reason for the homography construction to fail.
- Leuven: Like the bikes sequence the viewpoint in this sequence is constant. This simplifies matching as we can see by the results. A change in brightness between the images is handled well by our approach.

The success of our approach to segment matching depends on good filtering and distinct descriptive segments. It works the best for sequences with only little viewpoint change and handles other varying conditions, such as blur and brightness change, well.

For sequences where construction of the homography is possible correct matches almost reach ground-truth level, meaning when the segments are available they are matched correctly. To conclude this summary we show the best correspondences between two images of each sequence in Figure 4.9.

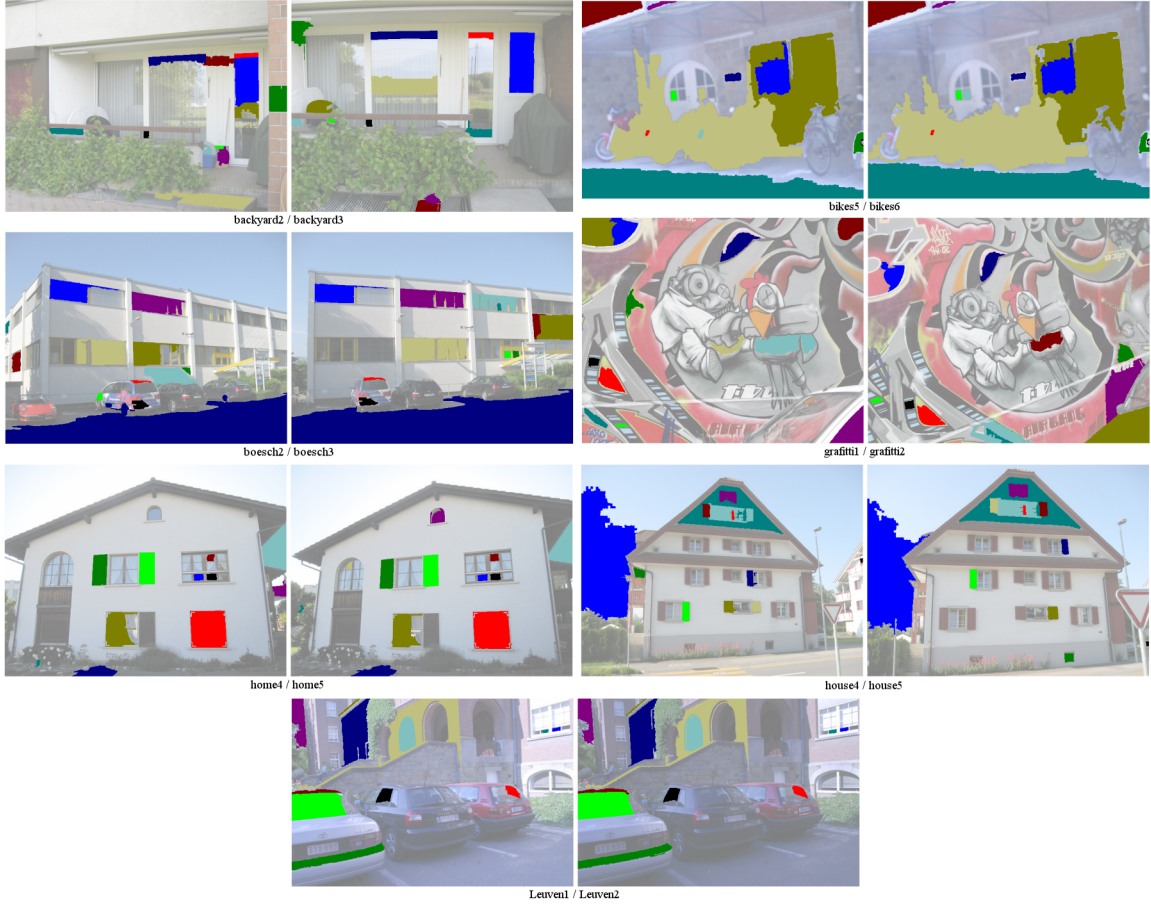


Figure 4.9: The best correspondence for two images of each sequence in the experimental dataset. For each correspondence the best 12 matches are shown. For visualization purpose the matches in the two images have the same colour. This way correct matches are distinguishable from wrong ones.

4.6 Discussion

We have presented an approach to descriptive segment matching using mutual information as a correspondence measure. With the reduced set of descriptive segments matching

is a much simpler task than with a full segmentation hierarchy. Segment matching is very dependent upon the segment filter which has to provide corresponding descriptive segments. Without them matching is impossible due to the incompatible image signatures. For some of the evaluated image sequences the provided signatures were not suitable for matching. These sequences were also difficult for other feature matching approaches like the combined SIFT and RANSAC approach used to construct the homography. Even though our approach is based on local decisions, it still finds most of the available ground-truth matches. A global view on matching could, however, improve our approach. We have seen difficulties for instance with the house sequence where segments are very similar. By considering the position of these similar segments we might be able to distinguish them and find the correct match. This requires some assumptions about the transformation between the images. When there is only a horizontal translation of the viewpoint, for instance, it is desirable for a segment at the bottom not to correspond with one at the top. Due to this restriction a similar segment at the bottom is preferred. Including more of the hierarchical information in the matching is another considerable improvement. In our approach the hierarchical information is only used for a refinement that removes obviously wrong matches. We believe, however, that the information in the reduced hierarchy can be used to support the matching itself. Additionally, it may also be possible to use a tree-based matching approach to find the correspondences of the segments. Such matching approaches have been used before, for instance by Glantz et al. [35], for matching whole segmentation trees. Approaches like this may work even better for a reduced tree that only contains descriptive segments.

The results are promising, but further work is required to improve our region matching approach to become an image matching approach. For image matching the construction of the transformation between the images is required. Such a construction needs relations between pixels rather than regions and requires some assumptions about the images. Regions usually contain enough pixels for which a correspondence can be established. Of particular interest are feature points, like local invariants and corners, of the regions. Additionally, the centre of gravity or other special points of the regions may provide suitable pixel correspondences as well. Under the assumption of the regions being on a planar surface the pixel correspondence can be used to compute a homography matrix. With the homography established the correspondence of other regions on the same planar surface can be verified. This makes the identification and removal of mismatches possible. Ultimately, with an approach like RANSAC the homography backed up by most region correspondences can be found. For the more general case where regions are not on a

planar surface a fundamental matrix can be computed from the correspondence of pixels. With the fundamental matrix any point in one image is mapped to an epipolar line in the other. This epipolar geometry makes it more difficult to verify region correspondences, but it is still possible by the analysis of the mapping of feature points in both regions.

With a relatively simple algorithm matching regions in complex images was possible. This has shown that the concept of reducing the segments in an image to a small number of descriptive segments simplifies successive tasks. We believe that, with the proposed improvements, this approach has the potential to become competitive to the current state-of-the-art image matching approaches.

Chapter 5

Conclusion

In this thesis, we proposed a region-to-region matching system based on a filtered hierarchical segmentation. The system is composed of segment filtering and a region-to-region matching approach. In the filtering phase of our system the large number of segments of a hierarchical segmentation is reduced in order to lower the complexity of successive algorithms. Careful selection of the segments included in this reduced set is required to retain an accurate image signature. We have seen that filtering based on the criteria introduced in Section 3.1 leads to suitable image signatures. Successive processing, like our region-to-region matching, benefit from these simplified signatures due to the limited availability of segments.

Hierarchical segmentations are very interesting image signatures. They do not assume any granularity of the regions and contain the fine and the coarse structures of an image. The availability of regions of any size comes at the cost of a large feature space. Approaches that use hierarchical segmentations have to deal with the large number of segments available. Our descriptive segment filter, being the key contribution of our work, chooses a set of segments important to the image. The importance of these descriptive segments to the image is computed based on their appearance and their position in the hierarchical segmentation. Two different approaches were evaluated for such a filtering. The queue-based one, which uses the merge history of a bottom-up hierarchical segmentation algorithm, has shown to be unsuitable for the selection of descriptive segments. Instead, we propose a two-phase filtering approach that is composed of a selection and a sorting phase. In the selection phase potentially descriptive segments are chosen from the hierarchical segmentation by the difference in size compared to their parent. The sorting phase takes these potentially descriptive segments and orders them by importance to

the image. This sorting uses a combination of variance, appearance-based cost and size of the segments to determine the ordering. For the evaluation of our approach we use a comprehensive dataset composed of image sequences from the literature and our own work. Based on this dataset we have shown that our filtering identifies the important regions independent of the used segmentation algorithm. This reduced image signature lowers the complexity of many algorithms that use hierarchical segmentations.

An approach that benefits from the simplified image signature is our region-to-region matching. Given two images of the same scene we use a modified mutual information measurement to estimate the similarity of regions. Based on this similarity a set of region correspondences between the images is established. The approach is generic and does not make any assumptions about the images. This is especially useful for situations where no projected geometry exists, for instance when matching an image to a sketch or a 3D model. For the evaluation of our approach we established a set of ground-truth matches for the dataset used in filtering. We have seen that correct region correspondences are identified for most of the images in the dataset. The magnitude of the number of correct matches however varies depending upon the image content. Even though the results are encouraging further work is required to refine the matching.

Our approach successfully reduces a hierarchical segmentation to a set of descriptive segments. The chosen segments are a suitable signature that contains unique features of an image. Based on the simplified image signature region matching with a correspondence measure is possible. This is only one of the many applications our descriptive segment filter is suitable for. Any approach that uses hierarchical segmentations can benefit from this filtering.

5.1 Future Work

We would like to verify the effectiveness of our approach to hierarchical segment filtering with other applications and additional segmentation algorithms. The segmentation algorithms that we used had minor influence on the filtering but there are some assumptions about the structure of the hierarchical segmentation, i.e., the significance of segment size, that might not necessarily be true for other hierarchical segmentation algorithms. In this case an adjustment of the selection and sorting criteria is needed. Regardless, the refinement of the criteria, especially for sorting, is an avenue of research for further improving the filter. Though the results were satisfactory a sort criteria with a larger penalty for non-descriptive segments may even enhance them.

The presented region-to-region matching approach is one application of our filtering. The construction of a reduced hierarchy based on the descriptive segments makes the filtering suitable for any algorithm that uses a hierarchical segmentation. This includes algorithms in the field of content based image retrieval, stereo vision, object recognition, scene reconstruction and many more. In content based image retrieval, for instance, a similarity measure based on descriptive segments could be computed. This task is similar to our region matching and the correspondence measure can possibly be modified for it. It is definitely desirable to use the filtering for applications different than matching.

Our matching approach features the computation of a region-to-region correspondence based on a similarity measure. This similarity measure uses a modified mutual information measurement which requires the transformation of the regions to a coordinate system where their centre of gravity is at the origin. This causes same regions in two images not to overlap each other when the images are rotated. A proposed improvement to the similarity measure is to maximize the overlap of regions by translation and rotation. This will enhance the similarity measure because the entropy shared between the regions is dependent on their overlap.

There are various ways to establish a transformation between images. Most of them require a set of corresponding pixels between the images. It is feasible to find feature point correspondences based on the region correspondences. Corresponding feature points may also be found in an asymmetric approach where one image is left untouched. The transformation between the images can not be computed without assumptions about them. When the features are on a planar surface a homography is to be computed. A fundamental matrix describes the transformation in case they are not. Generally, this is only applicable when the imaging process can be described with projective geometry but if matching computer-generated images that may not hold.

Refinement validates the established region correspondences against the structure of the segmentations. Matches that do not agree with the structure are removed. The hierarchical structure, especially its reduced form composed of descriptive segments only, certainly contains more information than our matching approach currently uses. Hence, refinement can be improved to add or reallocate matches based on the hierarchical structure. Ultimately, the hierarchy may be the only datastructure necessary for matching. The number of operations necessary to transform a tree of descriptive segments into another one can be a measure for similarity. Among a set of images, the image with the reduced hierarchy that requires the least number of operations to match the hierarchy of the query image, could be chosen as the match.

Appendix A

Dataset

This appendix contains the images in the experimental dataset. Each figure shows one of the seven sequences of the dataset. Within the figures the rows contain (f.l.t.r.) the original image, the manually selected descriptive segments, the two-phase filtering result and the union between the two. For more information about the two-phase filtering refer to Section 3.4. All images were segmented using the standard segmentation algorithm.



Figure A.1: The backyard image sequence.



Figure A.2: The bikes image sequence.



Figure A.3: The boesch image sequence.

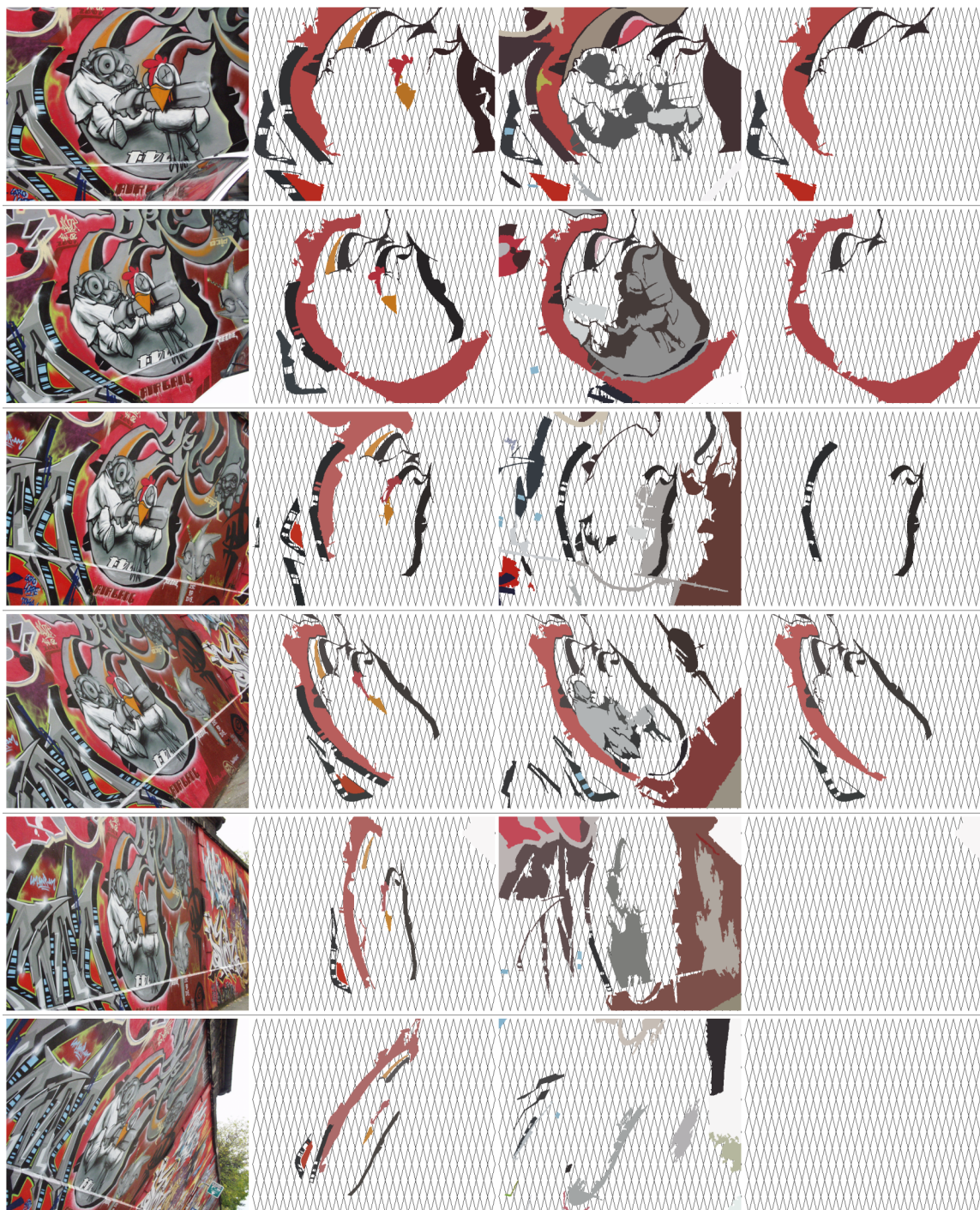


Figure A.4: The graffiti image sequence.



Figure A.5: The home image sequence.



Figure A.6: The house image sequence.



Figure A.7: The Leuven image sequence.

The following figures present the difference between the standard (top) and the alternative (bottom) segmentation algorithm. Each figure contains the same columns as the previous ones.

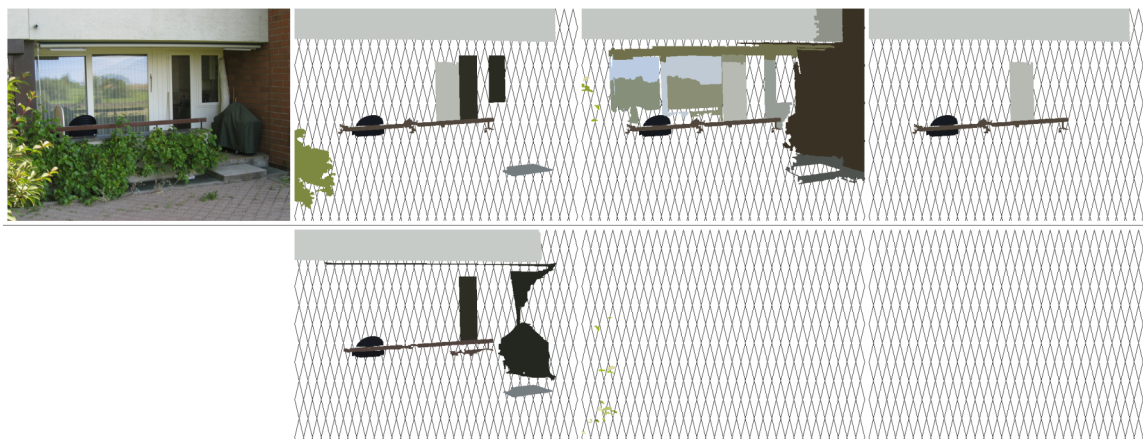


Figure A.8: Segmentation algorithm comparison using backyard1 image.

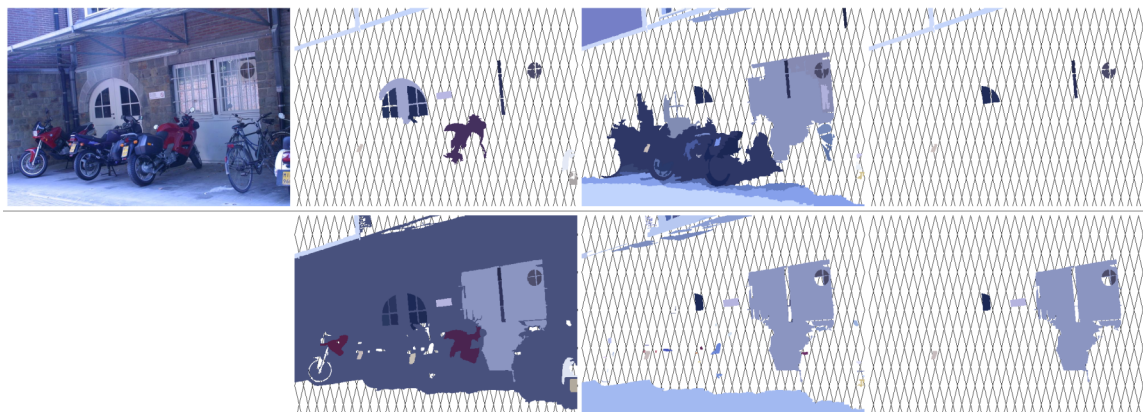


Figure A.9: Segmentation algorithm comparison using bikes1 image.

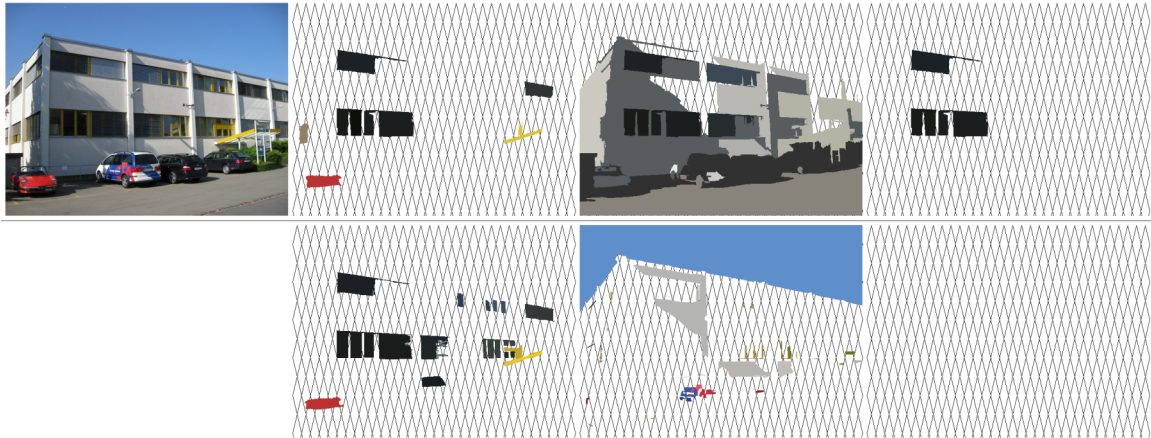


Figure A.10: Segmentation algorithm comparison using boesch2 image.

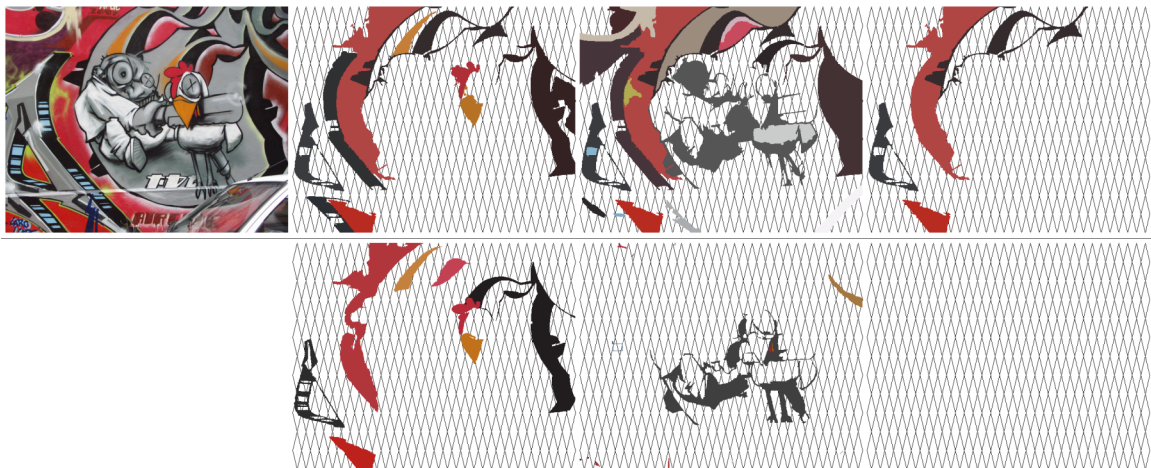


Figure A.11: Segmentation algorithm comparison using graffiti1 image.

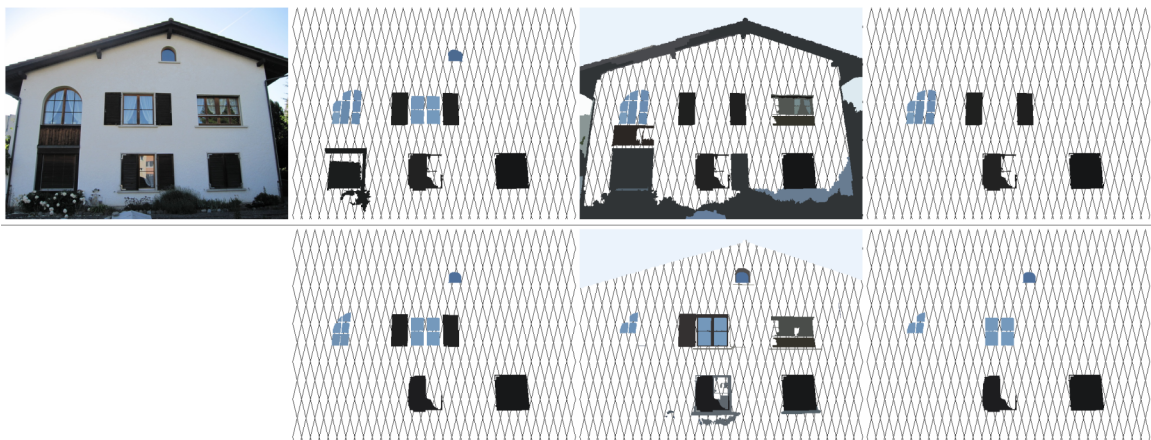


Figure A.12: Segmentation algorithm comparison using home3 image.

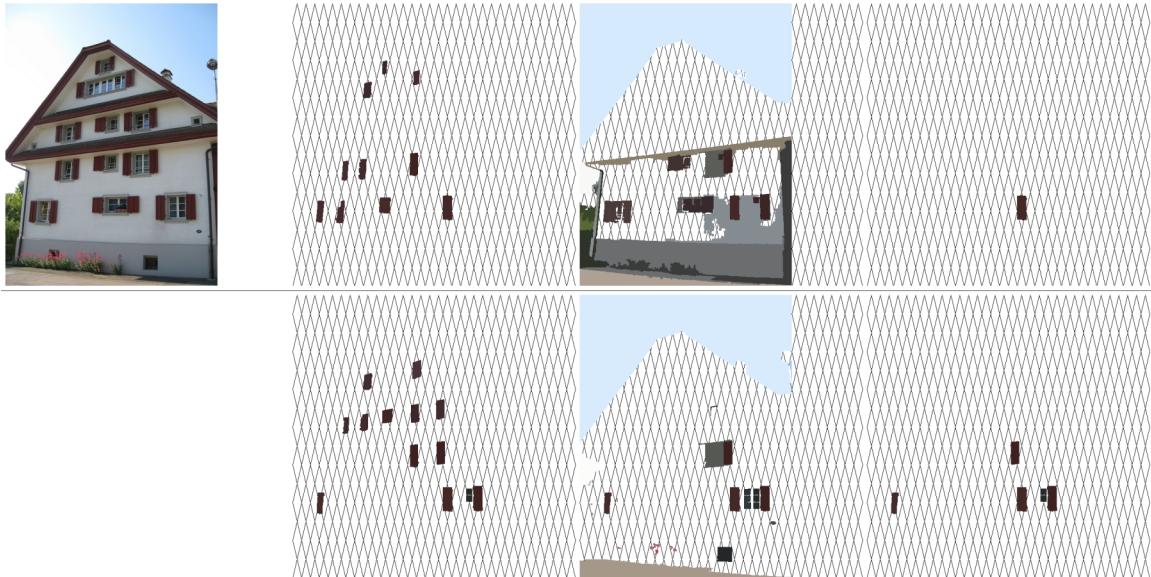


Figure A.13: Segmentation algorithm comparison using house2 image.

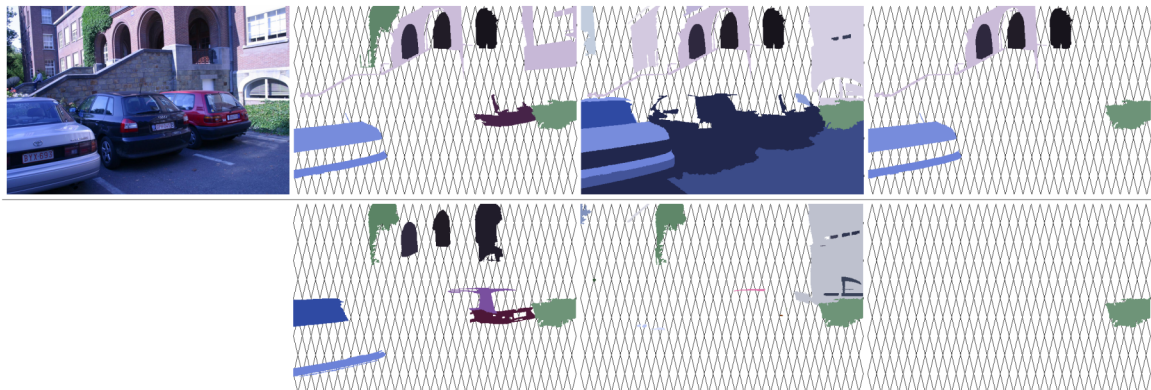


Figure A.14: Segmentation algorithm comparison using Leuven1 image.

Bibliography

- [1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC Superpixels. Technical report, École Polytechnique Fédérale de Lausanne, EPFL, 2010.
- [2] N. Ahuja. A transform for multiscale image segmentation by integrated edge and region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1211–1235, 1996.
- [3] N. Ahuja and S. Todorovic. Connected segmentation tree — a joint representation of region layout and hierarchy. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2008*, pages 1–8, 2008.
- [4] P. Arbelaez. Boundary extraction in natural images using ultrametric contour maps. In *Proc. Conf. Computer Vision and Pattern Recognition Workshop CVPRW '06*, 2006.
- [5] P. Arbelaez and L. Cohen. Constrained image segmentation from hierarchical boundaries. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2008*, pages 1–8, 2008.
- [6] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2009*, pages 2294–2301, 2009.
- [7] David Arthur and Sergei Vassilvitskii. How slow is the k-means method? In *Proceedings of the twenty-second annual symposium on Computational geometry*, SCG '06, pages 144–153, New York, NY, USA, 2006. ACM.
- [8] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. In

- Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*, pages 659–663, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc.
- [9] Johannes Bauer, Niko Sünderhauf, and Peterbarr Protzel. Comparing several implementations of two recently published feature detectors. In *Prezentováno pri International Conference on Intelligent and Autonomous Systems IAV*, 2007.
 - [10] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Proceedings of European Conference on Computer Vision*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417, Berlin, Heidelberg, 2006. Springer Berlin / Heidelberg.
 - [11] J.-M. Beaulieu and M. Goldberg. Hierarchy in picture segmentation: a stepwise optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):150–163, 1989.
 - [12] S. Berretti, A. Del Bimbo, and P. Pala. Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Transactions on Multimedia*, 2(4):225–239, 2000.
 - [13] G. Borgefors. Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, 1988.
 - [14] Thomas Brox, Dirk Farin, and Peter H.N. de With. Multi-stage region merging for image segmentation. In *In: Proceedings of the 22 nd Symposium on Information Theory*, page 189, 2001.
 - [15] G. Carneiro and A. D. Jepson. Multi-scale phase-based local features. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, volume 1, 2003.
 - [16] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: image segmentation using expectation-maximization and its application to image querying. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(8):1026 – 1038, August 2002.
 - [17] W. Chauvenet. *A Manual of Spherical and Practical Astronomy V2: Theory and Use of Astronomical Instruments*. Kessinger Publishing, 1891.

- [18] Junqing Chen, T. N. Pappas, A. Mojsilovic, and B. Rogowitz. Adaptive image segmentation based on color and texture. In *Proc. Int. Conf. Image Processing 2002*, volume 3, pages 777–780, 2002.
- [19] Yixin Chen and J.Z. Wang. A region-based fuzzy feature matching approach to content-based image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(9):1252 – 1267, September 2002.
- [20] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.
- [21] L. Cohen, L. Vinet, P. T. Sander, and A. Gagalowicz. Hierarchical region based stereo matching. In *Proc. CVPR '89. IEEE Computer Society Conf Computer Vision and Pattern Recognition*, pages 416–421, 1989.
- [22] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Seutens, and G. Marchal. Automated multimodality image registration using information theory. In *Information Processing in Medical Imaging*, 1995.
- [23] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40:5:1–5:60, May 2008.
- [24] R. B. Dean and W. J. Dixon. Simplified statistics for small numbers of observations. *Analytical Chemistry*, 23:636–638, 1951.
- [25] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001.
- [26] M. N. Do and M. Vetterli. Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance. *IEEE Transactions on Image Processing*, 11(2):146–158, 2002.
- [27] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. <http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html>.
- [28] Li Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

- [29] Pedro Felzenszwalb and Daniel Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004.
- [30] Martin A. Fischler and Oscar Firschein. *Readings in computer vision: issues, problems, principles, and paradigms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.
- [31] P.-E. Forssen. Maximally stable colour regions for recognition and matching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR '07*, pages 1–8, 2007.
- [32] P.-E. Forssen and D. G. Lowe. Shape descriptors for maximally stable extremal regions. In *Proc. IEEE 11th Int. Conf. Computer Vision ICCV 2007*, pages 1–8, 2007.
- [33] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.
- [34] R. Glantz, M. Pelillo, and W. G. Kropatsch. Hierarchical matching of panoramic images. In *Proc. 12th Int Image Analysis and Processing Conf*, pages 328–333, 2003.
- [35] Roland Glantz, Marcello Pelillo, and Walter G. Kropatsch. Matching segmentation hierarchies. *Int. J. of Art. Intell. and Pattern Recognition*, 18(3):397–424, 2004.
- [36] K. Grauman and T. Darrell. Efficient image matching with distributions of local invariant features. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2005*, volume 2, pages 627–634, 2005.
- [37] Chunhui Gu, J. J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2009*, pages 1030–1037, 2009.
- [38] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.
- [39] K. Haris, S. N. Efstratiadis, and N. Maglaveras. Watershed-based image segmentation with fast region merging. In *Proc. Int. Conf. Image Processing ICIP 98*, pages 338–342, 1998.

- [40] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [41] Yong Seok Heo, Kyoung Mu Lee, and Sang Uk Lee. Mutual information-based stereo matching combined with sift descriptor in log-chromaticity color space. In *Computer Vision and Pattern Recognition*, pages 445–452. IEEE, 2009.
- [42] Feng Jing, Mingjing Li, Hong-Jiang Zhang, and Bo Zhang. An efficient and effective region-based image retrieval framework. *Image Processing, IEEE Transactions on*, 13(5):699–709, May 2004.
- [43] Yan Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2004*, volume 2, 2004.
- [44] Jaechul Kim and K. Grauman. Asymmetric region-to-image matching for comparing images with generic object categories. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2344–2351, 2010.
- [45] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 15–18, 0-0 2006.
- [46] B. Ko and H. Byun. Frip: a region-based image retrieval tool using automatic image segmentation and stepwise boolean and matching. *Multimedia, IEEE Transactions on*, 7(1):105–113, 2005.
- [47] ByoungChul Ko and Hyeran Byun. Integrated region-based image retrieval using region’s spatial relationships. In *Proc. 16th Int Pattern Recognition Conf*, volume 1, pages 196–199, 2002.
- [48] P. Koniusz and K. Mikolajczyk. On a quest for image descriptors based on unsupervised segmentation maps. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 762–765, 2010.
- [49] D. Santosh Kumar and C. V. Jawahar. Robust homography-based control for camera positioning in piecewise planar environments. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, 906–918, 2006.
- [50] D. Lareau. Thesis in preparation. Master’s thesis, University of Ottawa, 2011.

- [51] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Semi-local affine parts for object recognition. In *In BMVC*, pages 959–968, 2004.
- [52] Jia Li, James Z. Wang, and Gio Wiederhold. Irm: integrated region matching for image retrieval. In *Proceedings of the eighth ACM international conference on Multimedia*, MULTIMEDIA '00, pages 147–156, New York, NY, USA, 2000. ACM.
- [53] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [54] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. Seventh IEEE Int Computer Vision Conf. The*, volume 2, pages 1150–1157, 1999.
- [55] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, November 2004.
- [56] D. MacDonald, J. Lang, and M. McAllister. Evaluation of colour image segmentation hierarchies. In *Proc. 3rd Canadian Conf. Computer and Robot Vision*, 2006.
- [57] Darren T. MacDonald. Image segment processing for analysis and visualization. Master’s thesis, University of Ottawa, 2008.
- [58] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar k-means problem is np-hard. In Sandip Das and Ryuhei Uehara, editors, *WALCOM: Algorithms and Computation*, volume 5431 of *Lecture Notes in Computer Science*, pages 274–285. Springer Berlin / Heidelberg, 2009.
- [59] Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *British Machine Vision Conference (BMVC)*, September 2007.
- [60] 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 *Proc. 8th Int’l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.
- [61] 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*, 26(5):530–549, 2004.

- [62] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004. British Machine Vision Computing 2002.
- [63] Marina Meilã. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 577–584, New York, NY, USA, 2005. ACM.
- [64] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.
- [65] 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*, 65:2005, 2005.
- [66] R. Nock and F. Nielsen. Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1452–1458, 2004.
- [67] M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1105–1120, 1999.
- [68] E. G. M. Petrakis, A. Diplaros, and E. Milios. Matching and retrieval of distorted and occluded shapes using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1501–1516, 2002.
- [69] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846 – 850, 1971.
- [70] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. Ninth IEEE Int Computer Vision Conf*, pages 10–17, 2003.
- [71] A. Rosenfeld and J. L. Pfaltz. Distance functions on digital pictures. *Pattern Recognition*, 1(1):33–61, July 1968.
- [72] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:2000, 2000.

- [73] Claude E. Shannon, Warren Weaver, and Shannon. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.
- [74] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [75] Peter Shirley and Steve Marschner. *Fundamentals of Computer Graphics*. A K Peters; 3rd Revised edition, 2009.
- [76] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.
- [77] Noah Snavely, Rahul Garg, Steven M. Seitz, and Richard Szeliski. Finding paths through the world’s photos. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, 27(3):11–21, 2008.
- [78] Sony. Playstation move motion controller. <http://us.playstation.com/ps3/playstation-move/>, 2011.
- [79] C. Studholme. An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recognition*, 32(1):71–86, January 1999.
- [80] Q. Tian, N. Sebe, M. S. Lew, E. Loupilas, and T. S. Huang. Content-based image retrieval using wavelet-based salient points, 2001.
- [81] S. Todorovic and N. Ahuja. Extracting subimages of an unknown category from a set of images. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, volume 1, pages 927–934, 2006.
- [82] S. Todorovic and N. Ahuja. Scale-invariant region-based hierarchical imagematching. In *Proc. 19th Int. Conf. Pattern Recognition ICPR 2008*, pages 1–5, 2008.
- [83] Sinisa Todorovic and Narendra Ahuja. Region-based hierarchical image matching. *Int. J. Comput. Vision*, 78:47–66, June 2008.
- [84] Andrea Torsello and Edwin R. Hancock. Computing approximate tree edit distance using relaxation labeling. *Pattern Recognition Letters*, 24:1089–1097, May 2003.
- [85] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560, 1995.

- [86] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008.
- [87] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2001*, volume 1, 2001.
- [88] P. Viola and III Wells, W. M. Alignment by maximization of mutual information. In *Proc. Conf. Fifth Int Computer Vision*, pages 16–23, 1995.
- [89] J. Z. Wang, Jia Li, R. M. Gray, and G. Wiederhold. Unsupervised multiresolution segmentation for images with low depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):85–90, 2001.
- [90] James Z. Wang, Nozha Boujemaa, Alberto Del Bimbo, Donald Geman, Alexander G. Hauptmann, and Jelena Tesić. Diversity in multimedia information retrieval research. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, MIR '06, pages 5–12, New York, NY, USA, 2006. ACM.
- [91] James Ze Wang, Jia Li, and Gio Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. In *Proceedings of the 4th International Conference on Advances in Visual Information Systems*, VISUAL '00, pages 360–371, London, UK, UK, 2001. Springer-Verlag.
- [92] J. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.
- [93] R. C. Wilson and E. R. Hancock. Structural matching by discrete relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):634–648, 1997.
- [94] S. X. Yu and Jianbo Shi. Segmentation given partial grouping constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):173–183, 2004.
- [95] Hui Zhang, Rouhollah Rahmani, Sharath R. Chollei, and Sally A. Goldman. Local image representations using pruned salient points with applications to cbir. In *Proc. 14th Annual ACM Int. Conf. on Multimedia*, 2006.