Master's Thesis

Robust Aerial Image Matching in Temporal Variant Regions

Gernot Margreitner

Graz, May 2010

Graz University of Technology
Erzherzog-Johann-Universität

Institute for
Computer Graphics and Vision

Vexcel Imaging
Microsoft Photogrammetry

Thesis supervisor:

Univ.-Prof. Dipl.-Ing. Dr.techn. Horst Bischof

"Properly speaking, such work is never finished;
one must declare it so when, according to time
and circumstances, one has done one's best."

[Johann Wolfgang von Goethe, Italian Journey]

Abstract

This thesis deals with the problem of finding correspondences between images that capture the same image scene taken at different times. The time differences can vary from a few minutes up to several months which makes it even harder to reliably find correspondences. Basically, local features proved to be a powerful way to find such correspondences because they are robust to background clutter, occlusions, or changes of the viewpoint. Even though numerous comprehensive feature evaluations have been published, none of these works focused the performance evaluation in the presence of *temporal variations*. This is a major drawback because in many applications multi-temporal image matching is a crucial component in order to successfully solve the posed problem. Consequentially, this thesis presents a multi-temporal performance evaluation of selected local detectors and descriptors for non-planar *aerial* imagery.

The primary goal of this work is to develop a temporal insensitive image matching workflow that is robust to temporal changes in aerial imagery and achieves highly accurate correspondence alignments. Such a matching algorithm may serve as a fundamental component of a broad range of applications. For example, the demonstrated algorithm prototype can be used to enhance existing photogrammetric workflows, where manually intensive user intervention is usually required in order to correctly match images in the presence of temporal changes.

Kurzfassung

In dieser Diplomarbeit wird das Problem der Bildung von Bildkorrespondenzen zwischen Bildern, die zu unterschiedlichen Zeitpunkten aufgenommen wurden, studiert. Der Zeitraum, der zwischen den Aufnahmen vergangen ist, reicht von einigen wenigen Minuten bis hin zu mehreren Monaten, was eine zusätzliche Herausforderung für die zuverlässige Erkennung solcher Korrespondenzen darstellt. Methoden, die auf sogenannten "local features" basieren, stellen einen leistungsfähigen Ansatz zum Finden von Bildkorrespondenzen dar, weil sie insbesondere robust gegen Änderungen im Bildinhalt sind. Aus diesem Grund gibt es auch zahlreiche Publikationen, die sich mit der Evaluierung dieser Methoden auseinandersetzen. Das Problem ist dabei jedoch, dass sich keine dieser Evaluierungen dem Problem der *zeitlichen Änderung* des Bildinhaltes widmet. Genau mit solchen Änderungen ist man aber in vielen Anwendungsfällen zum Beispiel aus der Luftbildphotogrammetrie konfrontiert. Aus diesem Grund werden in dieser Arbeit die Auswirkungen von zeitlichen Veränderungen in *Luftbildern* auf die Performance von "local features" evaluiert.

Ziel dieser Arbeit ist die Entwicklung eines Workflows, der einerseits robust gegenüber zeitlichen Änderungen in Luftbildern ist und andererseits Bildkorrespondenzen mit hoher Präzision findet. Ein solches System könnte in vielen Anwendungen den Benutzerkomfort erheblich verbessern und gleichzeitig die Qualität steigern: So ist es zum Beispiel in vielen Anwendungen nach wie vor der Fall, dass der Benutzer manuell Korrespondenzen in Bildern suchen muss, wenn der automatische Ablauf durch die zeitlichen Veränderungen zwischen den Bildern zu keinen geeigneten Lösungen kommt.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

______________________________ ______________________________

Date Signature

Contents

List of Figures

List of Tables

Acknowledgements

It would have been impossible for me to do this work without the help of several people.

First and foremost I want to thank Prof. Horst Bischof for his immediate attention to my questions, his valuable comments and personal guidance in supervising my master's thesis. At this point, I would also like to thank Dr. Bernhard Reitinger, the initiator of this project, for his encouraging engagement. Special thanks go to Dr. Matthias Rüther for offering his help and passing his professional opinion.

I am deeply grateful for my parents for giving me the opportunity to do my studies and for their support in everything I did. Last but not least, I want to thank my beloved partner for her understanding and patience.

Gernot Margreitner
Graz, May 2010

Credits

In this thesis, I was gratefully able to use publicly available software which was a great help throughout my work. Hence, I would like to thank the following individuals and organisations for permission to use their material:

- Dr. Bernhard Reitinger and his team at Vexcel Imaging for providing parts of their data material.
- Dr. Thomas Pock for providing source code and binaries to solve TV-L^1 models [16].
- Dr. Andreas Ess for providing a 64-bit binary of the SURF library.
- Dr. Peter Kovesi for his collection of MATLAB functions for computer vision and image processing [21].
- Dr. Krystian Mikolajczyk for his collection of local feature detectors and descriptors [28].
- Dr. Andrea Vedaldi and Brian Fulkerson for the VLFeat [60] computer vision library.
- Intel Corporation and Willow Garage for the OpenCV [62] computer vision library.

Chapter 1

Introduction

Contents

During the last decade, the digital revolution brought a massive progress in terms of availability as well as capability of digital imaging devices. Driven by the technological advances and the broad applicability of digital imaging, an enormous amount of digital imagery is newly created every day. In order to efficiently manage this ever growing number of (partially redundant) data it is more and more important to develop automated procedures that automatically process the data without the need of manual user interaction.

One of these important steps toward automated image processing is called *image registration*. Image registration is the fundamental process of aligning overlapping images and stitching them seamlessly into high-resolution images. The list of applications based on image registration includes mosaic construction [6], change detection [44], or three-dimensional model extraction [54]. Of course, this list of possible application fields is by no means exhaustive. However, all application scenarios have a common technical background: An *unknown* spatial transformation maps locations in one image to corresponding locations in another image and the key to successfully align a set of images is to *determine* this spatial transformation. This task is also referred to as *image matching*. Even if the geometric fundamentals of this problem are well investigated, many challenges remain. For example, significant changes of the image characteristics pose a problem: If images are taken at different times, with different illumination conditions, or from widely separated viewpoints, it is, e.g., hard to find the same physical location in each image.

1.1 Problem Definition

Due to the numerous types of image degradation and the diversity of images to be matched, it is still impossible to develop a general-purpose algorithm that is applicable to all image matching tasks. For example, Yang et al. [65] studied the registration of challenging image pairs. Their goal was to develop a general-purpose registration algorithm that can cope with low image overlap, substantial scale changes, or physical changes in natural image scenes. The proposed algorithm is based on the extraction and matching of local features detected with both the Difference-of-Gaussian [24] and the Harris-Affine [32] detectors and characterized with the SIFT [24; 25] descriptor. Informally spoken, the algorithm works as follows. Starting from a single feature match, the algorithm searches for additional matches in the neighborhood of this initial match. Each time an additional match is found, the estimated alignment model is updated and refined. These steps are iteratively repeated until convergence. The proposed algorithm achieves impressing results for a variety of image pairs. However, in the presence of large appearance variations such as physical changes due to different seasons, the algorithm shows significant performance deficiencies.

Obviously, achieving accurate corresponding locations between *multi-temporal* image sets is a challenging task, because the effects of temporal variations are manifold and tightly related with the captured image scene. Consequentially, it is essential to use local features in order to address the problem of multi-temporal image matching. Even though comprehensive evaluations of local features have been published [11; 12; 30; 33; 34; 52; 59], they are usually limited to certain domains and none of these works focused the performance evaluation in the presence of temporal variations.

The absence of evaluation results is a major drawback, because multi-temporal image matching is often required particularly in fields such as photogrammetry where the overall performance is tightly related with the accuracy of the image matching stage. For example, in aerial photogrammetry a block of images is processed in order obtain a digital model of the ground. In this case, a camera is mounted to an aircraft and temporal variations evolve at the time of image acquisition. During a flight mission, images of the same scene are captured from different viewpoints and at different times. Low time differences are based on the flight path over the surveyed area. They can vary between seconds in flight direction, several minutes when adjacent strips are captured strip-by-strip, or several hours for interleaved strip alignment. High time differences are the result if images of the same scene is captured by different flight missions. Often, this happens on a regular basis during different seasons of the year or during several years.

Figure 1.1 shows the occurrence of short-term temporal variations. The image shown in Figure 1.1a was captured about noon and the image in Figure 1.1b was captured the other day in the afternoon. Due to the different positions of the sun, the tower building casts a shadow on the parking cars (cf. green squares in the images). Furthermore, there is a significant change in the lineup of the parking cars in the entire parking area (highlighted with red rectangles). Note that cars usually have a high contrast to their surroundings, hence, they are typically targeted by feature detectors. Therefore, this kind of temporal variations yield a high number of feature detections that (a) cannot be re-detected in other images but (b) significantly increase the computational complexity.

In Figure 1.2, long-term temporal variations are illustrated. The image displayed in Figure 1.2a was captured in the late afternoon in winter, whereas Figure 1.2b shows the same image scene captured at noon about five months later in spring time. When these two images are compared, the following points have to be noticed. (a) There is a massive change of appearance caused by seasonal vegetation. For example, see the green rectangles in the images where the trees cover a large part of the road. (b) The low position of the sun causes large shadows that can cover significant image structures (cf. red rectangles). (c) Due to the withered vegetation in Figure 1.2a the shadows are highly textured and feature detections may evolve accidentally (e.g. orange rectangle).

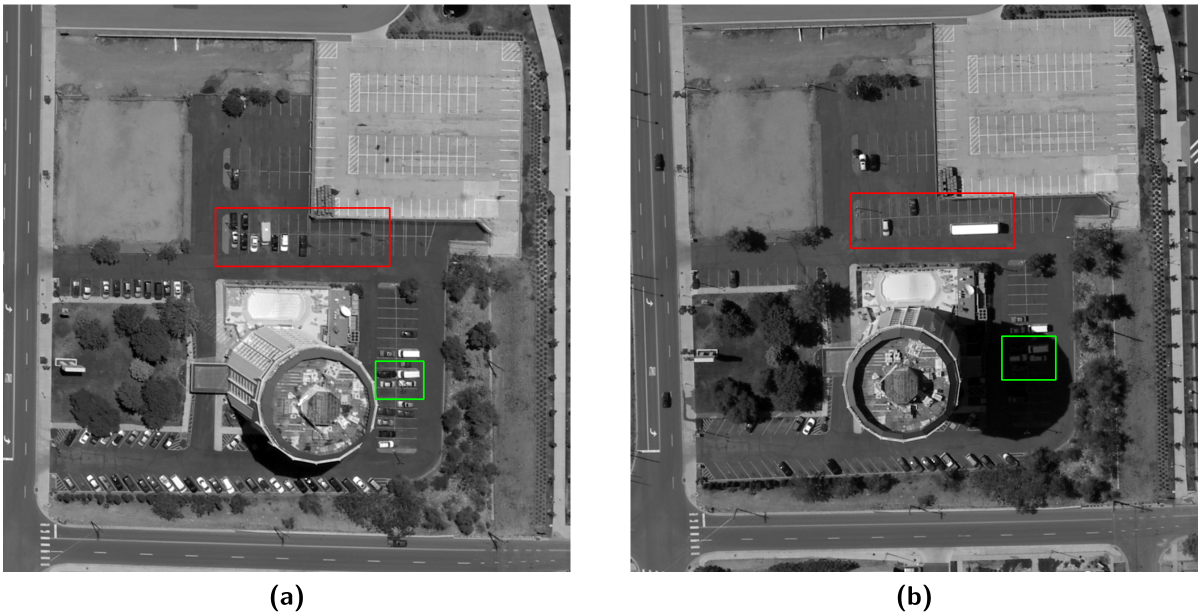


Figure 1.1: Illustration of short-term temporal variations.

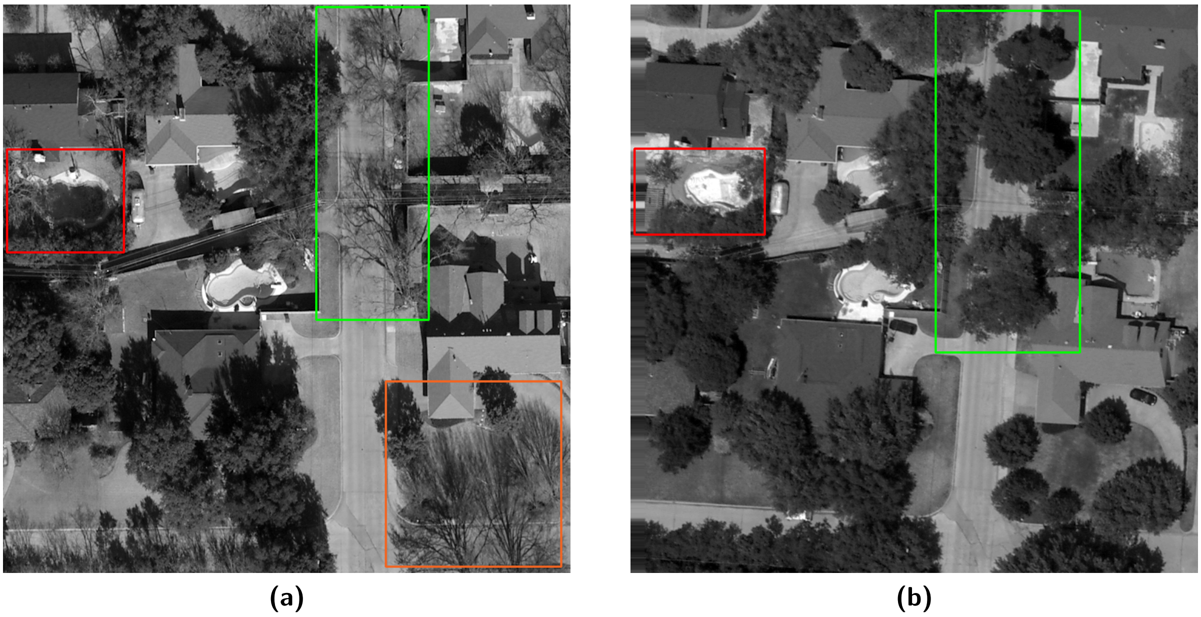


Figure 1.2: Illustration of long-term temporal variations.

1.2 Aims and Objectives of this Thesis

The goal of this work is to develop a temporal insensitive image matching workflow that is robust to temporal changes in the image scene and achieves a high accuracy with correspondence alignment errors as small as possible, preferably less than a pixel. As already mentioned before, it is difficult to develop such an algorithm, because there is an unlimited variety of possible image scenes and distortions. Hence, the focus of this thesis is put on the domain of *aerial* imagery and the range of temporal variations collected in three different data sets. Such a matching algorithm may serve as a fundamental component of a broad range of applications. For example, in many photogrammetric tasks, it is important to extract topographic information such as an object's size, shape, or position solely from a collection of aerial images. Hence, it is important to capture the same scene from different viewpoints in order to triangulate the relative positions of corresponding locations in each image. In current systems this process typically runs fully automated. However, if the images capture a substantial amount of temporal variations, a user is required to manually identify and select correspondences. In order to design an algorithm that solves this problem, the following objectives must be addressed.

1. **Identification of potential image matching methods**
   Local features are the key component for many image matching applications. Based on a review of current state-of-the-art image matching techniques, a subset of promising approaches is selected and evaluated.

2. **In-depth performance evaluation of local features**
   Literature on local features is vast and comprehensive evaluation papers compare the properties of different approaches. However, these evaluations give no information about performance regarding multi-temporal analysis of real-world imagery. Therefore, this thesis provides a systematic evaluation of local feature performances particularly focused on multi-temporal image matching.

3. **Design of a temporal insensitive image matching workflow**
   Based on the evaluation results, a temporal insensitive aerial image matching algorithm – which is actually a system of existing approaches – is proposed. This workflow should be robust to a variety of temporal variations captured in the collected data sets and match corresponding locations with high accuracy.

1.3 Outline

This work is organized as follows. Chapter 2 presents an overview of a typical photogrammetric workflow and reviews local features from the very beginning up to the most recent developments. Chapter 3 deals with the experimental setup and protocol. Local feature detectors and descriptors are selected in Section 3.1. Section 3.2 introduces the collected data sets and in Section 3.3 the evaluation metrics are discussed. Section 3.4 continues with a recapitulation of ground truth generation. Finally, Sections 3.5 to 3.7 deal with the systematic evaluation of feature detectors and descriptors. These evaluation results are used to develop a fully automated temporal insensitive image matching workflow in Chapter 4. Finally, Chapter 5 concludes this thesis with a brief discussion of future work.

Chapter 2

Related Research

Contents

This chapter presents a review of related research. Section 2.1 introduces the organization of a typical photogrammetric workflow and shows how the proposed matching algorithm is related with such a system. Section 2.2 presents a survey of approaches to feature extraction schemes proposed in the literature, starting from the very beginning up to the recently developed state-of-the-art methods. Finally, Section 2.3 gives a brief introduction to image pre-processing based on total variation.

2.1 A Typical Photogrammetric Workflow

The American Society for Photogrammetry and Remote Sensing defines [1] photogrammetry as “the art, science, and technology of obtaining reliable information about physical objects and the environment, through processes of recording, measuring, and interpreting images and patterns of electromagnetic radiant energy and other phenomena.”

A traditional and widespread application of photogrammetry is to extract topographic information from aerial images. Informally spoken, aerial photogrammetry allows the reconstruction of certain characteristics of an object – such as the object’s size, shape, or position – solely from aerial images. In order to achieve this, a photogrammetric model which accurately maps 2D image coordinates to the corresponding 3D world coordinates (and vice versa) must be established. This coordinate transfer is directly related with both the *interior* (i.e. focal length and position of the principal point) and *exterior* (i.e. spatial position and view direction) camera orientation of an image. While information about the interior orientation is a priori known for calibrated cameras, precise information about the exterior orientation is usually unknown. Fortunately,

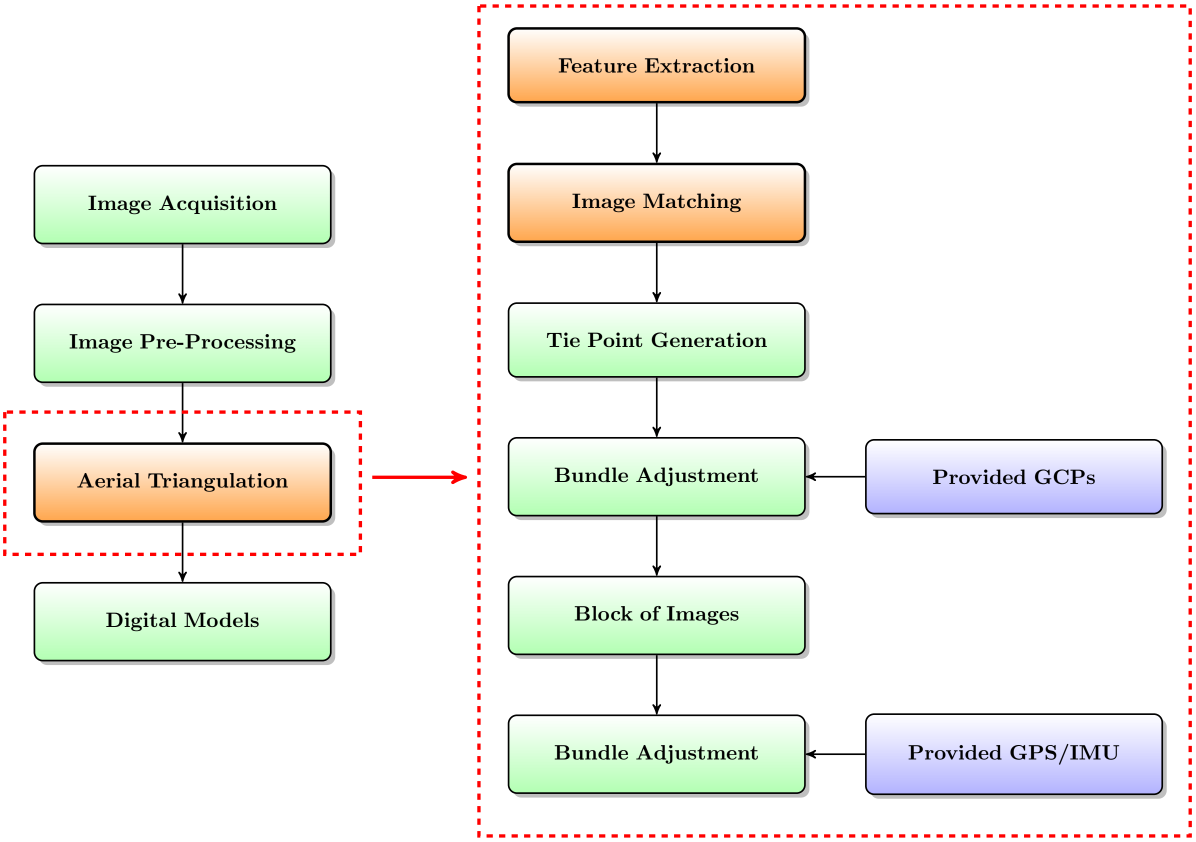


Figure 2.1: A typical photogrammetric workflow.

the elements of the exterior orientation can be obtained from *corresponding locations* that are found in two or more images.

Prior to the advances of computer-aided photogrammetry these corresponding locations were selected manually. In modern systems, the exterior orientation parameters are determined fully automated during *aerial triangulation* (AT). The AT computation stage is crucial for any photogrammetric application, because the performance of subsequent processing steps fully depends on the accuracy of this stage. Based on the image content, it might be difficult for the AT to determine correspondences between multiple views. This is particularly true, if a subset of images is heavily affected by temporal variations.

Typically, aerial triangulation consists of two steps, namely *relative* and *absolute* orientation of the image set. During relative orientation, adjacent images of an image set are linked together to form a block of images. The absolute orientation of the model computes an optimal fit of the image block to the ground coordinates. Figure 2.1 shows the block diagram of a typical photogrammetric workflow. The orange-colored blocks highlight processing stages that are directly related with multi-temporal image matching, whereas the blue-colored blocks indicate supplementary external input data. The remainder of this section gives a brief overview of the individual stages of the workflow.

Stage 1: Image Data Acquisition The collection of aerial imagery is based on a predefined flight plan. The details of the flight plan are influenced by several external conditions such as the solar altitude, the geographic spread, or the maneuverability of the airplane. The area under investigation is captured along parallel flight strips. Based on the intended purpose of the application, the required *redundancy* of the imagery denotes a major issue for flight planning. Typically, images within the same strip have an overlap of 80% (illustrated with the blue-colored area in Figure 2.2) and images from two adjacent strips have a sidelap of 60% (see green-colored area in Figure 2.2). This high redundancy provides an increased robustness and helps to automate the workflow. As already mentioned before, temporal variations are integrated in the imagery at this stage.

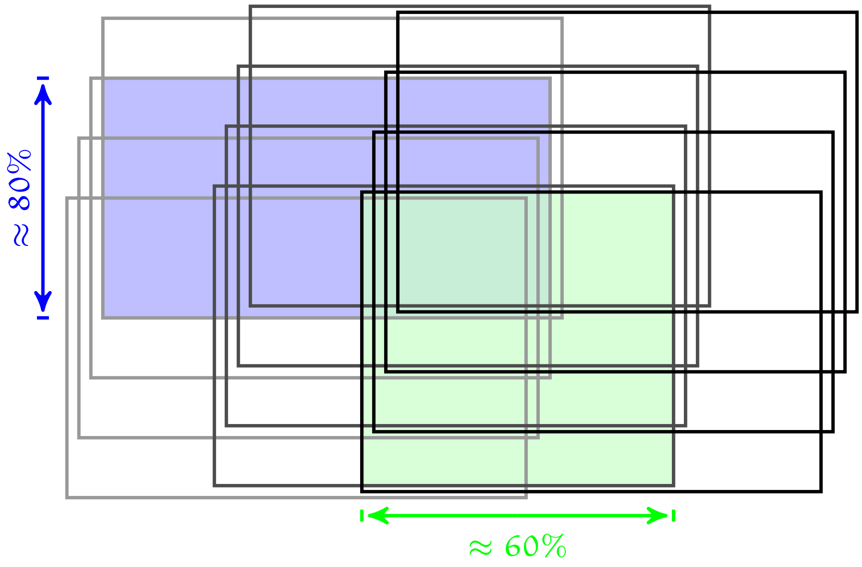


Figure 2.2: A block of overlapping images.

Stage 2: Image Pre-processing Modern digital cameras used for aerial image acquisition provide image formats up to 250 megapixels. Currently, no single CCD sensor with this size is available on the market, hence, a special concept that applies multiple sensors simultaneously is required. Due to this fact, there is a need for pre-processing of the raw aerial imagery. Basically, the pre-processing step eliminates radial or tangential distortions and resamples the individual images together to a high-resolution image.

Stage 3: Aerial Triangulation Aerial triangulation (AT) is a crucial component of any photogrammetric application. Informally spoken, AT correlates all images in a data set and aligns the correlated images to the ground. Since this thesis deals with the image correlation part of the AT, this processing stage is of special interest and will be discussed in more detail.

Stages 3.1 and 3.2: Feature Extraction and Matching The first step of AT computation is to establish correspondences between the images. Feature extraction is usually decomposed into feature detection and feature description. Due to the numerous types of distortions and the diversity of image scenes to be matched, the applied methods have to be specifically selected.

Once local features are detected for each image, they have to be matched in order to find corresponding locations in adjacent images.

Stage 3.3: Tie Point Generation and Ground Control Point Identification The term *tie point* refers to a feature correspondence between three or more images. The ground coordinates of tie points are unknown and computed in consecutive stages of the workflow. In contrast to tie points, ground control points (GCPs) are feature locations with known coordinates on the ground. Thus, GCPs establish a relationship between the collected imagery and the ground, whereas tie points link adjacent images to an image block.

Stages 3.4 to 3.6: Bundle Adjustment Typically, GPS (Global Positioning System) equipment which is attached to the aircraft gives only initial approximations to the exterior orientation parameters. Hence, the positions recorded during a flight mission are not accurate enough in order to exactly register the block of images to the ground. However, refined positions are available from Differential GPS (DGPS) and an inertial measurement unit (IMU). According to Triggs et al. [56], bundle adjustment can be defined as the problem of refining a visual reconstruction to simultaneously estimate 3D structure and exterior orientation parameters. Informally spoken, bundle adjustment solves a geometric parameter estimation problem by minimizing a cost function (e.g. the reprojection error) and refines both the exterior orientation parameters for each single image in the block and the ground coordinates of the tie points.

Stage 4: Digital Models After aerial triangulation, several digital models can be computed. A digital surface model (DSM) is a digital representation of the dense ground surface (including man-made structures), whereas a digital terrain model (DTM) is a digital representation of the terrain (i.e. the surface without man-made structures). Both models are the source for further digital representations like orthoimages or 3D maps that let users see 3D buildings which are textured using composites of aerial images [27].

2.2 Survey of Local Features

In general, the research work in the field of feature matching techniques can be divided into two main classes, namely, global and local methods. The former approach attempts to use all of the image information, while the latter strategy attempts to extract points of interest and use a small amount of local information to find matches. Since the focus of this work is put on local approaches they will be discussed in more detail in the remainder of this section.

Research literature on local approaches to feature extraction is vast and it is nearly impossible to discuss each single contribution. However, comprehensive reviews can be found in a series of survey papers by Fraundorfer and Bischof [11; 12], Mikolajczyk et al. [30; 31; 33; 34], Schmid et al. [52], or Tuytelaars et al. [59]. The discussion and explanations in this section are informal in the spirit of a review of the evolution of local approaches over the years rather than a rigorous discussion of the internals of each approach. Hence, the interested reader is pointed to the literature for further details.

2.2.1 Feature Detectors

Local methods use specific locations in the image, such as mountain peaks or building corners and extract significant information, e.g. gradient information, from their neighborhoods. This approach offers some desirable advantages since it both saves computational resources and improves robustness. However, the performance depends significantly on the reliability and accuracy with which corresponding points can be detected.

2.2.1.1 Early Work on Feature Detectors

Research on local feature-based correspondences dates back to the early years of stereo matching. Fundamental work into this direction was done by Moravec [35] and Beaudet [5] in the late 1970s.

Hessian Detector

Beaudet [5] developed a rotationally invariant blob detector based on the *Hessian matrix*

$$\mathcal{H} = \begin{bmatrix} I_{xx}(\mathbf{p}, \sigma_D) & I_{xy}(\mathbf{p}, \sigma_D) \\ I_{xy}(\mathbf{p}, \sigma_D) & I_{yy}(\mathbf{p}, \sigma_D) \end{bmatrix} \tag{2.1}$$

with second-order partial derivatives

$$I_{uv}(\mathbf{p}, \sigma_D) = \frac{\partial^2}{\partial u \partial v} g(\sigma_D) * I(\mathbf{p}) \tag{2.2}$$

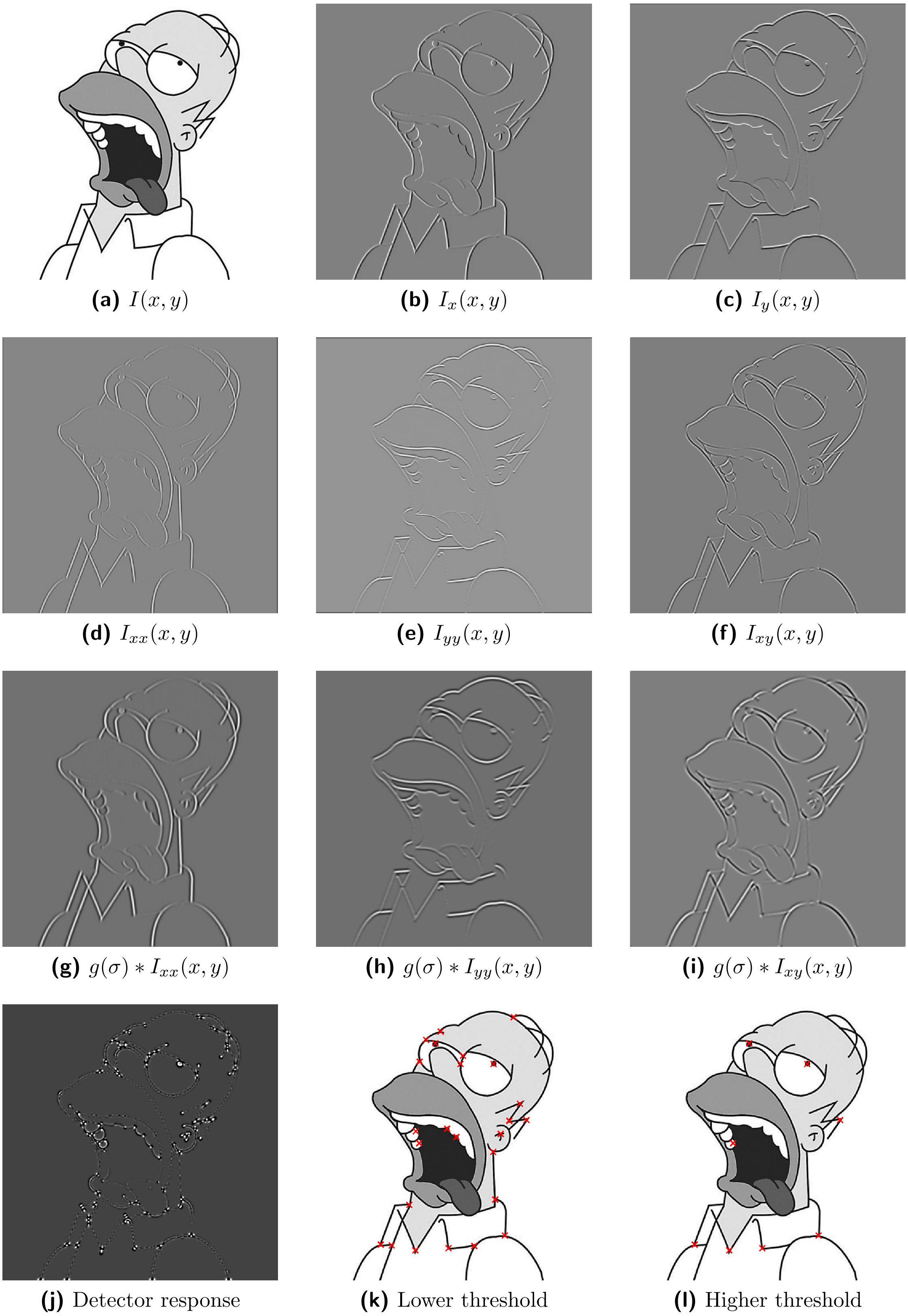
and Gaussian smoothing kernel

$$g(\sigma) = \frac{1}{2\pi\sigma^2}\; e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.3}$$

The Hessian matrix describes shape information in a local neighborhood of image point $\mathbf{p} = [x, y]^\top$ by the change of the normal to an isosurface. The detector response measure equals the determinant of the Hessian matrix

$$r_{hes} = \det(\mathcal{H}) = I_{xx} \cdot I_{yy} - I_{x,y}^2. \tag{2.4}$$

The feature detection procedure is illustrated in Figure 2.3. Given the input image (Figure 2.3a), the first step is to compute second-order Gaussian-smoothed image derivatives (Figures 2.3b – 2.3i). Each of these images contain different coefficients of $\mathcal{H}$. These values are combined to form the determinant of the Hessian matrix for each pixel (Figure 2.3j). Figures 2.3k and 2.3l show interest points detected with different thresholds of the determinant of the Hessian matrix.



(a) $I(x,y)$ **(b)** $I_x(x,y)$ **(c)** $I_y(x,y)$

(d) $I_{xx}(x,y)$ **(e)** $I_{yy}(x,y)$ **(f)** $I_{xy}(x,y)$

(g) $g(\sigma) * I_{xx}(x,y)$ **(h)** $g(\sigma) * I_{yy}(x,y)$ **(i)** $g(\sigma) * I_{xy}(x,y)$

(j) Detector response **(k)** Lower threshold **(l)** Higher threshold

Figure 2.3: Hessian blob detection scheme.

Moravec Detector

In the Moravec corner detection algorithm [35], each pixel is inspected to see whether it is a corner or not. This test is based on auto-correlation, i.e. measuring the gray-level difference of a patch centered on the pixel under inspection and nearby patches shifted in four directions parallel to the rows and columns of the image. The difference between the original patch and each shifted patch is expressed by the sum of squared differences

$$SSD(x,y) = \sum_{(i,j)\in W} \left(I(x+i,y+j) - I(i,j)\right)^2 . \tag{2.5}$$

If the SSD is low, there is only a low difference between the two patches (e.g. homogeneous region), otherwise, if the SSD is high, there is a significant variance in the gray-values of the patches. If the SSD is high for each shifted patch, the pixel is considered to be an interesting point and its strength is defined by the lowest of the four SSD values. In most cases such points are located on corners and edges. Subsequently, feature matching was based on correlation of small square image patches of specified size, centered on the detected interest points.

Obviously, the Moravec detector has a major limitation: Due to the discrete neighborhood shifts and the parallel shift directions, the detector is not isotropic. For example, the limited set of shifts does not allow to detect interest points located on edges that are oriented in other than the shift directions.

Harris Detector

Harris and Stevens [14] developed their popular corner detector using the *second moment matrix* (which is also known as auto-correlation matrix or structure tensor) instead of discrete shifts. The second moment matrix

$$\mathcal{M} = \left[\begin{array}{cc} I_x^2(\mathbf{p},\sigma_D) & I_x(\mathbf{p},\sigma_D)\cdot I_y(\mathbf{p},\sigma_D) \\ I_x(\mathbf{p},\sigma_D)\cdot I_y(\mathbf{p},\sigma_D) & I_y^2(\mathbf{p},\sigma_D) \end{array} \right] \tag{2.6}$$

with the partial derivatives

$$I_u(\mathbf{p},\sigma_D) = \frac{\partial}{\partial u} g(\sigma_D) * I(\mathbf{p}) \tag{2.7}$$

and Gaussian smoothing kernel

$$g(\sigma) = \frac{1}{2\pi\sigma^2}\, e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.8}$$

describes the gradient distribution in a local neighborhood of an image point $\mathbf{p} = [x,y]^\top$ by covering all possible shifts from the shift origin $\mathbf{p}$. The eigenvalues of $\mathcal{M}$ are proportional to the curvature of the local neighborhood of $\mathbf{p}$ and they provide a rotation invariant description of $\mathcal{M}$. There are three possibilities: (i) If both eigenvalues are low, the neighborhood of $\mathbf{p}$ has low intensity variations and is flat. (ii) If one value is high and the other is low, $\mathbf{p}$ is located on an edge because shifts along the edge cause small values whereas shifts perpendicular to the edge cause high values. (iii) If both eigenvalues are high then shifts in any direction cause high values and indicate a corner.

Harris and Stephens propose to use a cornerness measure based on the determinant (i.e. the product of the eigenvalues) and the trace (i.e. the sum of the eigenvalues) of matrix $\mathcal{M}$ in order to avoid explicit eigenvalue decomposition of the matrix. Let λ_1 and λ_2 be the eigenvalues of $\mathcal{M}$, they define the corner response function as

$$r_{har} = \det(\mathcal{M}) - \kappa \cdot Tr^2(\mathcal{M}) = \lambda_1 \cdot \lambda_2 - \kappa \cdot (\lambda_1 + \lambda_2)^2 \tag{2.9}$$

where κ denotes a sensitivity factor. A typical value for κ is 0.04.

Figure 2.4 illustrates the individual stages of the corner detection algorithm. Given the original image (Figure 2.4a), the first order derivatives in both x and y direction are computed (Figures 2.4b and 2.4c respectively). In the next step, the products of the partial derivatives are computed (Figures 2.4d, 2.4e, and 2.4f) and subsequently smoothed with a Gaussian kernel (Figures 2.4g, 2.4h, and 2.4i). Each of these images contain different elements of the second moment matrix. Combining these values gives a cornerness response value for each pixel (Figure 2.4j). Finally, corners are found at local maxima with response values above a certain threshold (Figures 2.4k and 2.4l).

Other Early Feature Detection Approaches

Working independently from Harris and Stevens, Förstner and Gülch [10] presented a double-stage workflow to detect interest points. During the first stage, interest points are detected with the auto-correlation matrix and at the second stage the localization accuracy is improved with a differential edge intersection technique

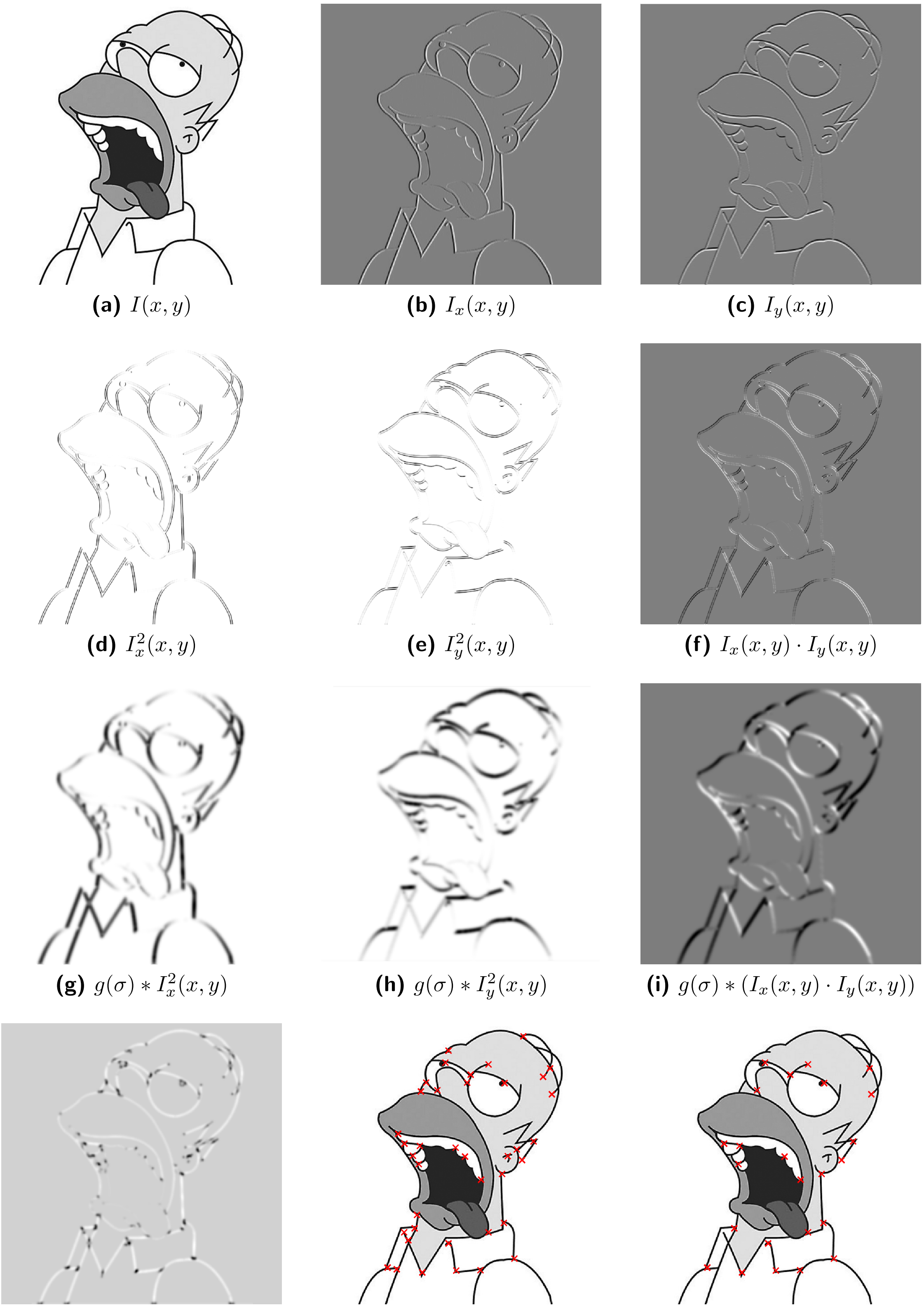
Tomasi and Kanade [55] used the concept of auto-correlation in the context of tracking. They search for interest points that can be well tracked and the proposed criterion is based on the smaller eigenvalue of their tracking matrix.

In [9] Förstner presents an interest point detector based on local image statistics. He used the auto-correlation matrix to classify pixels into one of three classes, namely region, contour, and interest point. Additionally, he showed that the analysis of local statistics allows to automatically find a threshold value for the interest point measure.

Summary

Basically, the main steps of auto-correlation based interest point extraction and matching can be summarized in the following way. (i) Compute the image derivatives and form the auto-correlation matrix. (ii) Compute a scalar measure for each pixel. (iii) Detect interest points as local maxima for measures above a certain threshold. (iv) Match image patches centered on interest points with cross-correlation.

However, one problem with these approaches is that they perform reasonably well only for *translational* changes, but many applications require interest points to be matched independently from the viewpoint. This limitation was subject to the development of interest point detection techniques that guarantee proper scale stability.



(a) $I(x,y)$ **(b)** $I_x(x,y)$ **(c)** $I_y(x,y)$

(d) $I_x^2(x,y)$ **(e)** $I_y^2(x,y)$ **(f)** $I_x(x,y) \cdot I_y(x,y)$

(g) $g(\sigma) * I_x^2(x,y)$ **(h)** $g(\sigma) * I_y^2(x,y)$ **(i)** $g(\sigma) * (I_x(x,y) \cdot I_y(x,y))$

(j) Detector response **(k)** Lower threshold **(l)** Higher threshold

Figure 2.4: Harris corner detection scheme.

2.2.1.2 Efforts toward Scale and Rotation Invariance

A first approach towards *scale* changes is fairly straight forward. So called multi-scale methods extract features over a pre-defined range of scales and use all features together to represent the image. These methods have a major drawback. If a local image characteristic occurs at several scales, multiple features are extracted with slightly changing localization and scale values. The high number of features increases the computational complexity and causes ambiguities related with feature matching. This disadvantage was addressed by the development of scale invariant approaches that automatically find a proper scale.

Influential research work on automatic scale selection was done by Lindeberg [23]. He uses circular patches of varying diameter in order to select maxima of the Laplacian-of-Gaussian (LoG) function as characteristic scales. Based on this technique, Mikolajczyk and Schmid [32] presented scale-adapted versions of both the Harris detector (i.e. Harris-Laplace) and the Hessian detector (i.e. Hessian-Laplace) where each method selects features by iteratively updating both position and scale until convergence. Note that the Laplacian-of-Gaussian function is circularly symmetric, hence, detectors based on this operator are *rotation invariant* by design.

Many scale-invariant detectors require the computation of more or less complex measures such as image derivatives or the second moment matrix. Since this has to be repeated for every single feature location (i.e. position and scale), this computation step might become computationally expensive soon. Thus, based on the work of Lindeberg, Lowe [24] used a set of Difference-of-Gaussian (DoG) filters in order to efficiently compute an approximation of the LoG.

Inspired by the DoG detector design, Bay et al. [3] developed a detector that uses box filters and integral images [61] in order to compute a fast approximation of the Hessian matrix. In this case, the determinant of the Hessian matrix is used for both selecting the spatial location and the scale.

Putting all the gathered feature information – such as spatial location and characteristic scale – together, the image patch for each feature location can be normalized to a unit circle and feature matching based on cross-correlation would work again.

2.2.1.3 Efforts toward Affine Invariance

In many computer vision applications, such as wide-baseline image matching, scale invariance and rotational invariance do not suffice. These applications may require invariance to *affine* transformations.

Maximally Stable Extremal Regions

Matas et al. [26] introduced a watershed-based segmentation algorithm termed Maximally Stable Extremal Regions (MSERs). MSERs are connected components in an image that are stable (i.e. they have a consistent size and shape) over a range of intensity threshold values.

In other words, MSERs are image regions where all pixels within the region boundary are either brighter or darker as the pixels surrounding the regions. This type of features has some favorable properties [26]. First, MSERs are invariant to affine intensity changes. Second, they are automatically detected at multiple scales. Third, MSERs are invariant to continuous geometric transformations. The latter property is probably most important, because it means that a continuous geometric transformation will transform a region into a region again. Hence, rotational changes, scale changes, or perspective transformations do not effect the repetitive detection of a region.

The detection of MSERs is implemented with a watershed-like thresholding algorithm: Consider all possible threshold values t_i for a gray-level image I. Thresholding image I with t_i creates a binary image where pixel intensity values less than t_i are considered to be "black" and values greater than or equal to t_i are considered to be "white". For the first threshold value t_0 the binary image is white. With an increasing t_i more and more black regions will appear. Some of these black regions will grow when t_i is further increased, however, some of them will hardly change during a series of threshold operations. These regions are of special interest and they will be selected as *maximally stable* if they satisfy the stability criterion

$$\Psi(R_i) = \frac{|R_{i+\Delta} \backslash R_{i-\Delta}|}{|R_i|} \tag{2.10}$$

where Δ is a free parameter, R_i is a region that is obtained by thresholding with gray value i, and $|\cdot|$ denotes cardinality. Figure 2.5 illustrates the stability criterion for the one-dimensional case. First, there are two distinct regions for threshold $t_{i-\Delta}$. When t_i further increases, the two regions merge and remain stable until threshold $t_{i+\Delta}$. Following this example, region R_i is clearly not an MSER, because its area differs significantly from the area of region $R_{i-\Delta}$.

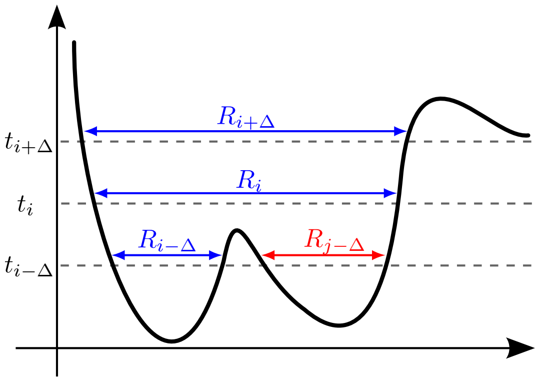


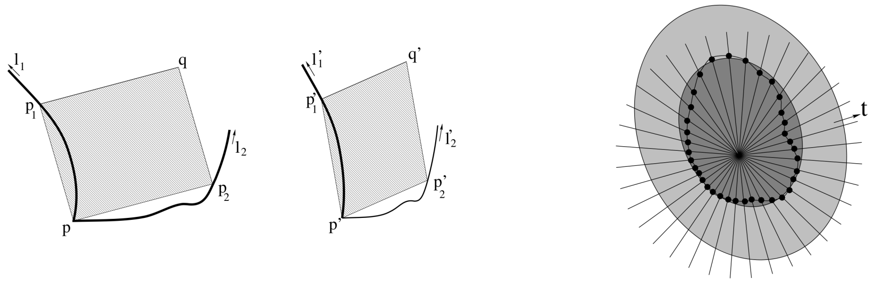
Figure 2.5: MSER stability criterion.

If it is preferable to have a single point location instead of a region one can compute the center of gravity of the region. In [38] Obdrzalek and Matas show that the centers of gravity of two regions are invariant to affine transformations.

Usually the watershed-like thresholding procedure is executed twice. The first run processes the original input image and detects dark regions (MSER+) while the second run uses the negative input image and detects bright regions (MSER-). The original algorithm proposed by Matas et al. [26] runs in $\mathcal{O}(n \ \log \ \log \ n)$ time. Recently, Nistér and Stewénius [37] developed an algorithm with a computational complexity of $\mathcal{O}(n)$.

Other Detectors

Tuytelaars and Van Gool [57] proposed Edge-based Regions (EBR). The construction of an EBR is illustrated in Figure 2.6a. Based on a Harris corner location $\mathbf{p}$, two points $\mathbf{p_1}$, $\mathbf{p_2}$ that move along intersecting edges in the neighborhood of this corner, are used to define an affine-invariant parallelogram. The points stop moving along the edges when a photometric measure of the texture covered by the parallelogram reaches an extremum.



(a) EBR (taken from [57]) **(b)** IBR (taken from [58])

Figure 2.6: Construction of Edge-based and Intensity-based Regions.

Similar to EBR, Tuytelaars and Van Gool [58] proposed another detector referred to as Intensity-based Regions (IBR). The construction of an IBR is illustrated in Figure 2.6b. Based on local intensity maxima, the intensity profiles along radially symmetric rays that emanate from each intensity maximum are evaluated. If the intensity profile changes significantly along a certain ray, a marker is placed at this location. All markers are connected to form a region of arbitrary shape and an ellipse is fitted to this region.

In [29; 34] Mikolajczyk and Schmid present affine adaptions of the Harris-Laplace and Hessian-Laplace detectors coined Harris-Affine and Hessian-Affine respectively. The affine extension is based on the shape estimation properties of the second moment matrix. The detectors are initialized with the detections from the multi-scale Harris and Hessian detectors and they determine the position, scale, and shape in order to obtain affine invariant regions.

Smith and Brady [53] developed a feature detector based on the Smallest Univalue Segment Assimilating Nucleus (SUSAN) principle. The SUSAN feature detection principle is illustrated in Figure 2.7 where a circular mask is shown at three different image positions. The center pixel of the circular mask is called nucleus (shown in red). The intensity value of each pixel within the mask is compared to the intensity value of the nucleus and an area of intensity values which are similar to the nucleus' value is defined. This area is called USAN. In Figure 2.7 the USANs are those parts of the circular masks that are overlapping with the dark rectangle. The USAN area is at a maximum when the circular mask is in a flat region (illustrated with the green circle). If the nucleus is located on a straight edge the USAN is halved (orange circle) and it is even further decreased when the nucleus is located near a corner point (blue circle). Thus, corners can be detected at locations where the USAN reaches a minimum (Smallest USAN = SUSAN).

Based on the SUSAN detector, Rosten and Drummond [45; 46] developed Features from Accelerated Segment Test (FAST). In contrast to SUSAN, only the 16 pixels that lie on a circle centered on the nucleus are compared (cf. Figure 2.8). If there are n contiguous pixels that are all even brighter or darker than the nucleus' intensity value, this point detected as corner. The authors use machine learning in order to compute a high-speed corner detector for any given n.

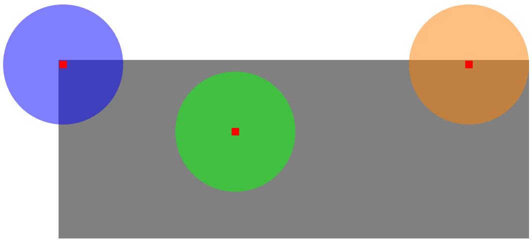


Figure 2.7: SUSAN feature detection principle.

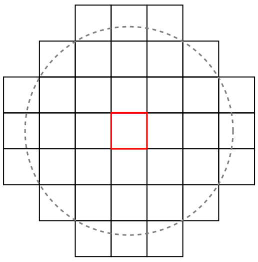


Figure 2.8: FAST segment test criterion.

2.2.2 Feature Descriptors

Given the detected interest points or regions, the next step is to determine corresponding locations in different images. Although simple measures such as the sum of squared differences or normalized cross-correlation can be used to directly compare the intensities of patches surrounding each interest point, it is usually preferable to use scale, orientation, and affine information to resample the patches. However, even after adjusting for such changes, the local appearance of the image patches is very likely to still differ from image to image. The key to this issue is the design of feature *descriptors*.

In their pioneering work, Schmid and Mohr [50] addressed the problem of matching images to a large image database. They proposed the use of local greyvalue characteristics, in their case differential greyvalue invariants introduced by Koenderink [19], to characterize detected interest points. This approach proved to achieve good results in the presence of occlusions or background clutter. Since the publication of this paper many new feature extraction schemes have been proposed.

Scale Invariant Feature Transform

The Scale Invariant Feature Transform (SIFT) [24; 25] is one of the most appealing feature extractors for practical uses because it is both distinctive and relatively fast. In the original work, SIFT refers to a combination of the Difference-of-Gaussian (DoG) detector and a distinctive feature descriptor. However, this section discusses solely the feature description stage which consists of two distinct tasks. First, a dominant orientation is assigned to each detected feature. Second, the feature descriptor is computed relatively to the assigned orientation.

Orientation Assignment The computation of a reproducible orientation makes the descriptor invariant to rotational changes. The descriptor orientation is obtained from gradient information in a local neighborhood of the feature. For each sample location $[x, y]^\top$ in the neighborhood both the gradient magnitude

$$m = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{2.11}$$

and the gradient orientation

$$\theta = \tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - I(x-1,y)}\right) \tag{2.12}$$

are computed. Here, $L(x, y) = g(x, y, \sigma) * I(x, y)$ is the selected Gaussian smoothed image where σ is closest to the detected scale s. The main orientation is obtained by creating a 36 bin histogram of oriented gradients. For each orientation θ the corresponding magnitude value m is first weighted with a Gaussian window and then inserted into the histogram. Finally, the highest peak in the orientation is selected as dominant orientation. Additionally, any other histogram peaks that are within 80% of the highest peak also denote a dominant orientation. Hence, multiple descriptors with differing orientation can occur for the same feature location.

Descriptor Formation Figure 2.9 illustrates the task of SIFT descriptor computation. First, the local neighborhood of the feature is divided into a 4×4 grid of subregions containing the Gaussian weighted gradients (cf. Figure 2.9a). For each subregion these samples are now accumulated into a 8 bin orientation histogram, giving a $4 \times 4 \times 8 = 128$ dimensional descriptor vector (cf. Figure 2.9b). In order to reduce the impact of linear and non-linear illumination changes, the vector is normalized to unit length, thresholded, and re-normalized again.

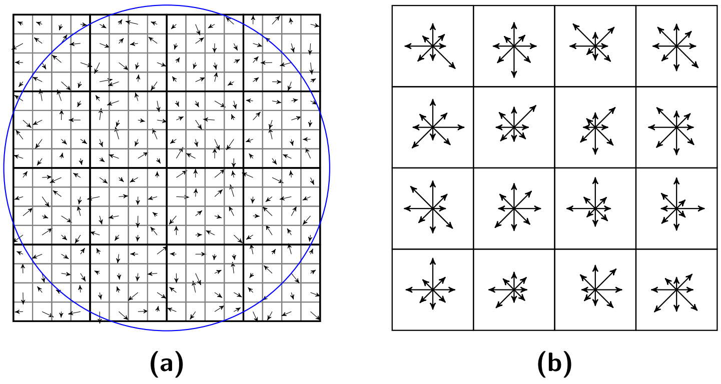


Figure 2.9: SIFT descriptor computation.

Speeded-up Robust Features

Similar to the SIFT descriptor, Bay et al. [3; 4] developed a highly efficient feature descriptor coined Speeded-up Robust Features (SURF). In contrast to SIFT, SURF is based on first-order Haar wavelet responses in both x and y direction instead of gradient information. This approach allows a high-speed implementation based on integral images [61].

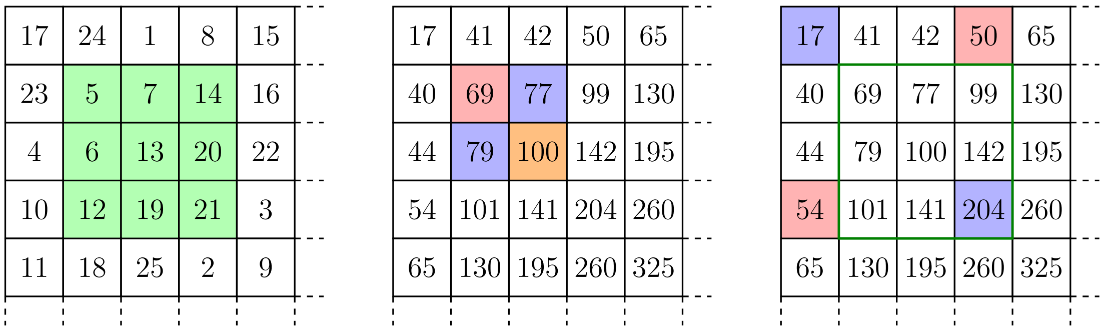
Figure 2.10 illustrates the basics of integral image computation. Each value in the integral image (highlighted orange in Figure 2.10b) is computed iteratively from the corresponding pixel value in the original image by adding (highlighted blue) and subtracting (highlighted red) its three adjacent neighbors in the integral image. More formally, the elements of an integral image $\mathcal{I}_\Sigma$ can be defined as

$$\mathcal{I}_\Sigma(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j). \tag{2.13}$$

Once the integral image is computed, the summed area

$$A = \mathcal{I}_\Sigma(x_1, y_1) - \mathcal{I}_\Sigma(x_1, y_1 - 1) - \mathcal{I}_\Sigma(x_1 - 1, y_1) + \mathcal{I}_\Sigma(x_0 - 1, y_0 - 1) \tag{2.14}$$

of a rectangle $[x_0, x_1] \times [y_0, y_1]$ is computed with four samples of the integral image only. For example, the sum of the highlighted area in Figure 2.10a equals 117 and the same result is obtained in Figure 2.10c by addition (highlighted blue) and subtraction (highlighted red) of the values at the rectangle corners. Note that this computation scheme allows to compute areas of arbitrary size in constant time.



17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

(a) Original image

17	41	42	50	65
40	69	77	99	130
44	79	100	142	195
54	101	141	204	260
65	130	195	260	325

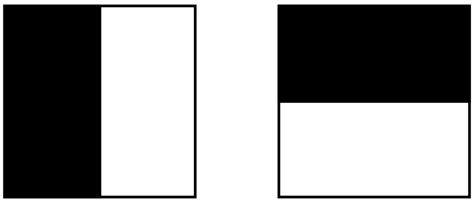
(b) Integral image

17	41	42	50	65
40	69	77	99	130
44	79	100	142	195
54	101	141	204	260
65	130	195	260	325

(c) Area computation

Figure 2.10: Integral image creation.

The concept of integral images is ideally suited to be combined with 2D Haar wavelets [39; 40]. These are simple box filters that encode the relationship between intensities of adjacent regions. The used filters are shown in Figure 2.11. In order to compute the wavelet response, the intensity values under the negative (i.e. black) area are averaged and subtracted from the averaged intensity value under the positive (i.e. white) area.



(a) Response in x direction **(b)** Response in y direction

Figure 2.11: 2D Haar wavelets used by the SURF descriptor.

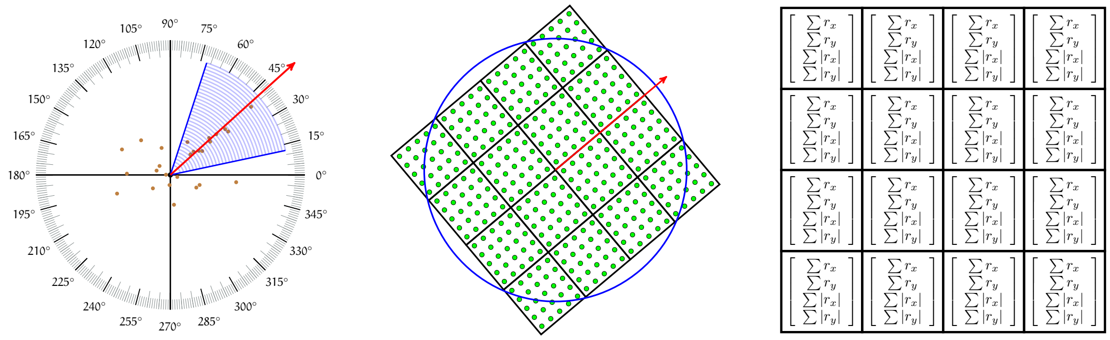
The SURF descriptor extraction task can be separated into two distinct stages, namely orientation assignment and descriptor formation.

Orientation Assignment In order to obtain rotationally invariant descriptors, a dominant orientation has to be found for each feature. Hence, points are regularly sampled with spacing s in a circular neighborhood of radius $6s$. Here, s denotes the scale at which the feature was detected. For each sample point the Haar wavelet responses r_x and r_y are computed and weighted with a Gaussian that is centered at the feature point. The response values are then represented as point (r_x, r_y) in a two-dimensional parameter space. Finally, the orientation is computed by rotating a sliding orientation window of size $\pi/3$. All points that are covered by the sliding window are used to form an orientation vector. The orientation of the longest vector out of all sliding windows is assigned to the feature. See Figure 2.12a for an illustration of the orientation assignment process.

Descriptor Formation The formation of a descriptor is based on square window of size $20s \times 20s$ that is relatively oriented according to the selected orientation. In order to encode spatial information the square region is divided into a regular grid of size 4×4. For each subregion, Haar wavelet responses r_x and r_y are computed for a 5×5 regular grid (cf. Figure 2.12b). Note that responses in x and y directions are relative to the selected feature orientation and each response is weighted with a Gaussian that is centered at the feature point. Based on the wavelet responses, each subregion is then characterized by a four-dimensional descriptor vector

$$\mathbf{v}_s = \begin{bmatrix} \sum r_x \\ \sum r_y \\ \sum |r_x| \\ \sum |r_y| \end{bmatrix} \tag{2.15}$$

Combining all subregion vectors builds a descriptor vector of dimension $4 \times 4 \times 4 = 64$ (cf. Figure 2.12c). In order to reduce the impact of illumination changes the descriptors are normalized to unit length.



(a) (b) (c)

Figure 2.12: SURF orientation assignment and vector formation.

Steerable Filters

Nearly all of the previously mentioned feature descriptors are distribution-based (i.e. these methods use histograms in order to represent certain characteristics of appearance or shape). In contrast to these methods, filter-based techniques approximate the neighborhood of a feature position with local derivatives up to a given order.

The Steerable Filter [13] feature descriptors are based on high-order derivatives of the Gaussian function. Basically, this approach evolves from the Taylor expansion

$$f(x_0 + x, y_0 + y) = f(x_0, y_0) + \left[\sum_{n=1}^{N} \frac{1}{n!} \left(x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y} \right)^n f(x_0, y_0) \right] + \mathcal{R}_N \tag{2.16}$$

with Lagrange remainder $\mathcal{R}_N$, which can be used to describe a small image region in the neighborhood of a feature location $[x_0, y_0]^\top$ by using a series of local derivatives up to a given order N. The responses of the derivatives are then combined to form a descriptor vector.

Note that the applicability of Steerable Filters is not limited to feature description. They can be used in a variety of computer vision tasks that involve oriented filters. Basically, Steerable Filters compute filter responses for a few pre-defined orientations only and subsequently interpolate the responses for *arbitrary* orientations. In [13] Freeman and Adelson provide the following illustrative example. Consider a two-dimensional Gaussian function

$$g(x, y) = \exp^{-(x+y)} \tag{2.17}$$

where scaling and normalization terms are set to 1 for convenience. The first-order partial derivatives of g are

$$g_x = -2x \exp^{-(x+y)} \qquad \text{and} \qquad g_y = -2y \exp^{-(x+y)} \tag{2.18}$$

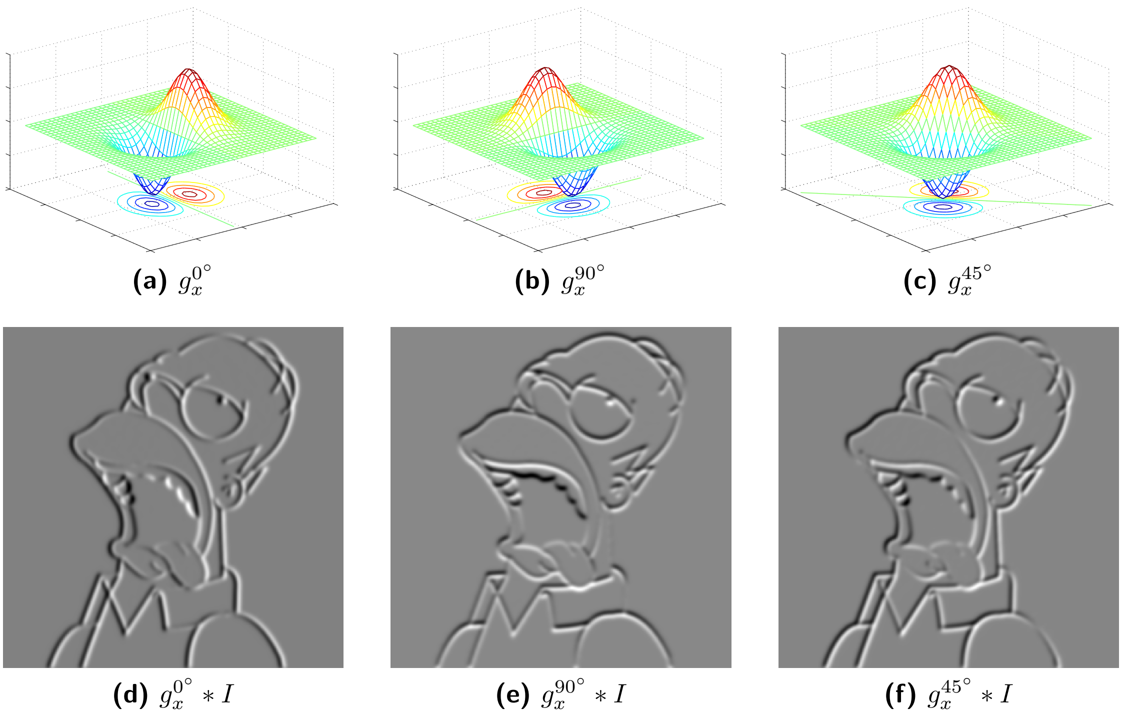
respectively. Since the Gaussian function is isotropic, g_y equals g_x rotated by 90° (i.e. $g_y = g_x^{90^\circ}$). Hence, it can be shown that any orientation θ can be computed by the linear combination of g_x and g_y:

$$g_x^\theta = \cos(\theta) g_x + \sin(\theta) g_x^{90^\circ} \tag{2.19}$$

This is illustrated in Figure 2.13, where a filter with orientation $\theta = 45^\circ$ (cf. Figure 2.13c) is interpolated from the base filters g_x and g_y. Remember that the convolution of two signals is a linear operation. Hence, analogous to Equation 2.19, it is straight forward to filter an image $I(x, y)$ at arbitrary rotation θ from the linear combination of images filtered with the base filters:

$$I^\theta = \cos(\theta)(g_x * I) + \sin(\theta)(g_y * I) = \cos(\theta) L_x + \sin(\theta) L_y \tag{2.20}$$

In general, the representation of a filter of n^{th} order requires the computation of $(n + 1)$ derivatives. The orientation of each derivative is given by



(a) $g_x^{0^\circ}$ **(b)** $g_x^{90^\circ}$ **(c)** $g_x^{45^\circ}$

(d) $g_x^{0^\circ} * I$ **(e)** $g_x^{90^\circ} * I$ **(f)** $g_x^{45^\circ} * I$

Figure 2.13: Illustration of a steerable filter.

$$\theta_{n,i} = \frac{i \cdot \pi}{n+1} + \theta_g \qquad i \in \{0, \ldots, n\} \tag{2.21}$$

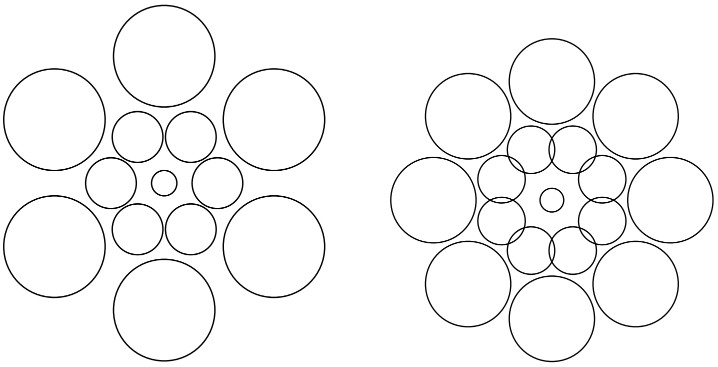
where θ_g is directly related with the local image structure. In [33] Mikolajczyk and Schmid use steerable filters up to the fourth order in order to compute low-dimensional descriptor vectors given by

$$\mathbf{v} = \begin{bmatrix} L_x \cos(\theta_{1,0}) \\ L_y \sin(\theta_{1,1}) \\ L_{xx} \cos^2(\theta_{2,0}) \\ 2 \cdot L_{xy} \sin(\theta_{2,1}) \cos(\theta_{2,1}) \\ L_{yy} \sin^2(\theta_{2,2}) \\ L_{xxx} \cos^3(\theta_{3,0}) \\ 3 \cdot L_{xxy} \sin(\theta_{3,1}) \cos^2(\theta_{3,1}) \\ L_{xyy} \sin^2(\theta_{3,2}) \cos(\theta_{3,2}) \\ L_{yyy} \sin^3(\theta_{3,3}) \\ L_{xxxx} \cos^4(\theta_{4,0}) \\ 4 \cdot L_{xxxy} \sin(\theta_{4,1}) \cos^3(\theta_{4,1}) \\ 6 \cdot L_{xxyy} \sin^2(\theta_{4,2}) \cos^2(\theta_{4,2}) \\ 4 \cdot L_{xyyy} \sin^3(\theta_{4,3}) \cos(\theta_{4,3}) \\ L_{yyyy} \sin^4(\theta_{4,4}) \end{bmatrix} . \tag{2.22}$$

In contrast to Mikolajczyk and Schmid, Winder et al. [63; 64] use this concept in order to develop a high-dimensional descriptor. In their approach they compute the even and odd responses of second-order quadrature Steerable Filters for $\eta = 6$ orientations. For each location within the patch, the descriptor vector

$$\mathbf{v} = \begin{bmatrix} pos(odd_j) \\ pos(-odd_j) \\ pos(even_j) \\ pos(-even_j) \end{bmatrix} \qquad pos(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.23}$$

is computed for orientation $j \in \{1, 2, \cdots, \eta\}$ giving a vector length of $k = 4\eta$. These vectors are then used to compute a Gaussian-weighted histogram into ξ spatial bins where each bin has dimension k. Figure 2.14 shows typical lineups of Gaussian-weighted regions with two concentric rings consisting of six segments (i.e. $\xi = 13$, see Figure 2.14a) and eight segments (i.e. $\xi = 17$, see Figure 2.14b) respectively. Each circle indicates a region where the radius equals a standard deviation of 1. Hence, the final descriptor vector has a dimension of $\xi \times k$.



(a) 2 Rings with 6 Segments **(b)** 2 Rings with 8 Segments

Figure 2.14: Gaussian summation regions for Steerable Filters.

Other Descriptors

Inspired by SIFT, Ke and Sukthankar [18] proposed an alternate, more compact representation of SIFT features. The proposed method accepts the same input – i.e. location, scale, and orientation – as the original SIFT descriptor. However, instead of using weighted orientation histograms, they sample the gradient region at 39×39 and apply Principal Component Analysis (PCA) the normalized gradient patch. This significantly reduces the dimensionality of the descriptor (from 3042 to 36) vector and improves matching speed.

Another extension of SIFT, namely gradient location and orientation histogram (GLOH), was developed Mikolajczyk and Schmid [33]. Instead of the 4×4 histogram array, this descriptor uses a log-polar binning structure (cf. Figure 2.15)with subsequent dimensionality reduction based on PCA.

In [22] Lazebnik et al. present a descriptor for texture matching applications based on spin images [17]. Each column of an intensity domain spin image (i.e. a two-dimensional histogram of intensity values) encodes the histogram of pixel intensities i with the same distance d to the center of the image patch. This is illustrated in Figure 2.16 where three sample points in the left image map three different locations in the spin image on the right.

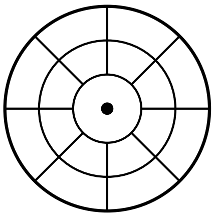


Figure 2.15: GLOH log-polar binning scheme.

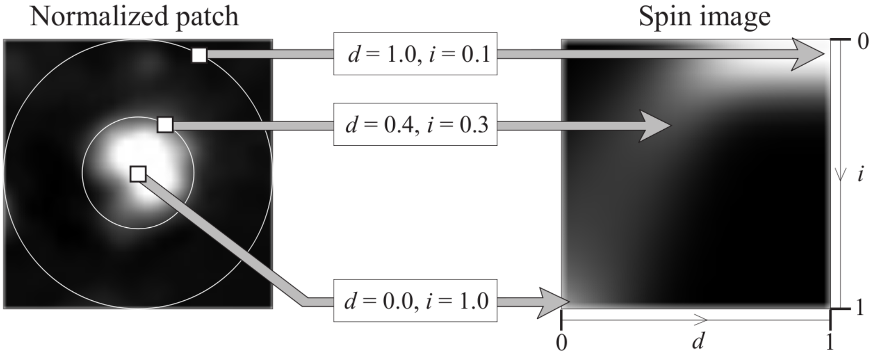


Figure 2.16: Construction of spin images (taken from [22]).

Another class of filter types is based on complex-valued coefficients. Schaffalitzky and Zisserman [49] use such a filter to develop an algorithm that organizes a set of unordered images. They use a bank of linear filters derived from the family

$$K_{m,n}(x,y) = (x+iy)^m(x-iy)^n G(x,y) \tag{2.24}$$

and compute a total of 16 rotational invariants for each image patch in order to establish an indexing scheme.

2.2.3 Summary

Concluding this review of local feature approaches it is important to note that: (i) The very nature of local feature detectors and descriptors is tightly related. For example, if the descriptor provides invariance to a particular type of transformation, the interest point detector can be less sensitive with respect to this transformation. (ii) There is no perfect combination of a particular feature detector and descriptor. Both detectors and descriptors must be carefully selected specifically for the addressed problem in order to achieve best results.

Local features proved to be well suited to many computer vision tasks like matching or recognition. Basically, the main challenge is to detect locations which are robust to various transformations and thus can be reliably matched under different viewing conditions. Different techniques have been proposed to extract such interesting locations and there exist a number of comprehensive performance evaluation papers [11; 12; 30; 33; 34; 52; 59].

2.3 Pre-processing based on Total Variation

The key component of any image matching algorithm is to detect *repeatable* features while ignoring non-relevant ones. The remainder of this section describes pre-processing based on total variation which is used to filter out irrelevant image content before features are extracted.

Variational methods have been successfully applied to solve a number of computer vision tasks and they have seen rapid progress in recent years. Basically, these methods minimize an energy functional that is specifically designed to address a certain problem. Especially, the total variation (TV) norm is of great interest for many computer vision problems due to its ability to preserve sharp discontinuities. In [48] Rudin et al. were the first who introduced TV methods in the field of computer vision. In their original formulation of the ROF model (named after the authors Rudin, Osher, and Fatemi)

$$\min_u \left\{ \int_\Omega |\nabla u| d\Omega \right\} \quad s.t. \quad \int_\Omega (u-f)^2 d\Omega = \sigma^2 \tag{2.25}$$

they applied the TV-L^2 norm for edge-preserving image denoising. In Equation 2.25, Ω denotes the image domain, u is the true image, and

$$f(x,y) = u(x,y) + n(x,y) \quad \text{with} \quad n(x,y) \sim \mathcal{N}(0,\sigma) \tag{2.26}$$

is the observed image degraded by zero-mean white Gaussian noise of variance σ^2. Since then, TV-based methods were successfully applied to solve many other problems such as 3D reconstruction [20], medical image registration [43], or face recognition [7].

In this work, the goal is to partition an image

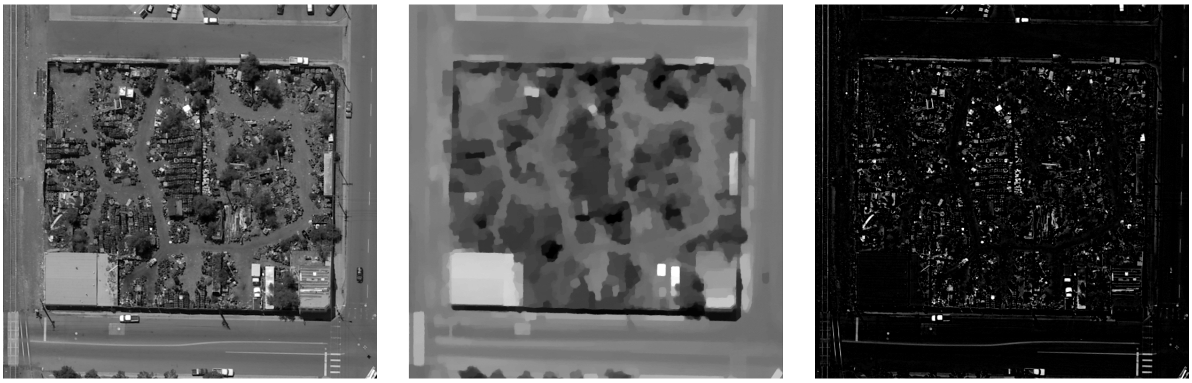
$$f(x,y) = u(x,y) + v(x,y) \tag{2.27}$$

into an image $u(x,y)$ that contains the structural part (i.e. large objects) of $f(x,y)$ and another image $v(x,y)$ which contains the textural information (i.e. fine details) and noise of $f(x,y)$. Figure 2.17 shows an illustrative example of the results of the image decomposition. This can be computed by solving the variational problem of the TV-L^1 model [36]

$$\min_u \left\{ \int_\Omega |\nabla u| d\Omega + \lambda \int_\Omega |u-f| d\Omega \right\} \tag{2.28}$$

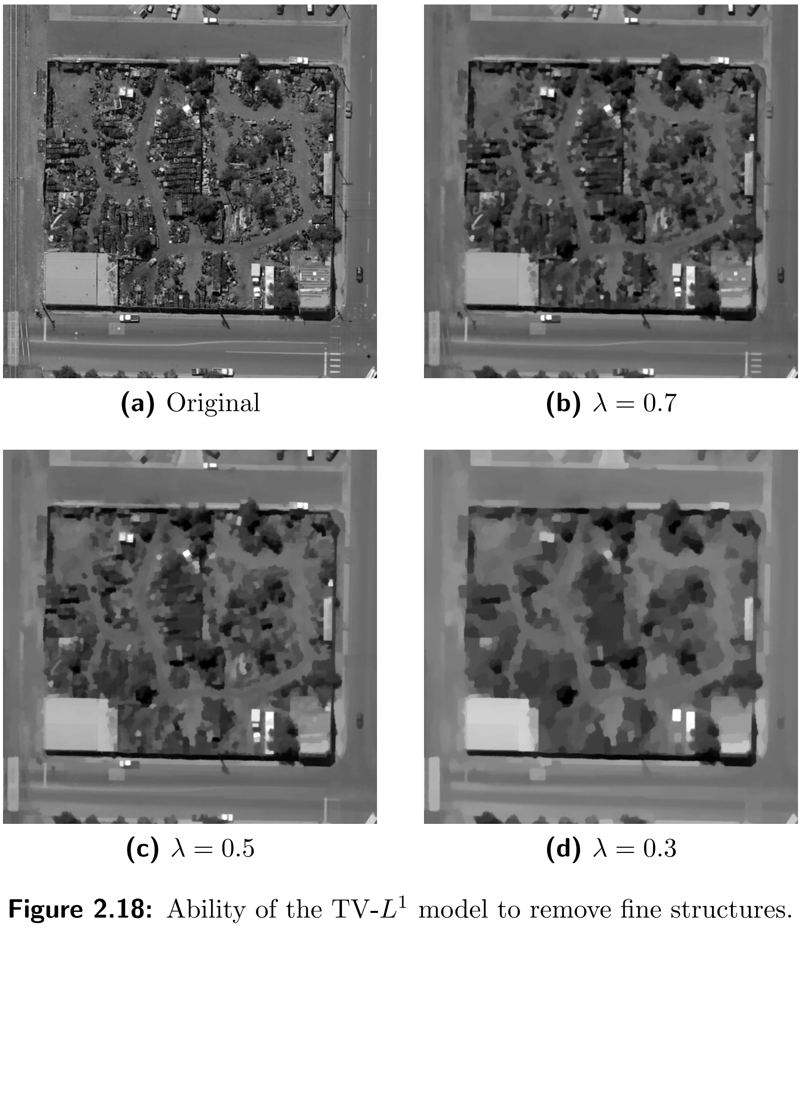
where Ω is the image domain and λ denotes a free parameter. Computing the solution of this variational model is not an easy task, because the L^1 norm is not differentiable at zero. However, there exist several numerical methods to find a solution (see [41; 42] for a review).

As Figure 2.17 indicates, the TV-L^1 model has a strong geometric capability [2; 41] which makes it particularly interesting for feature detection based on the structural part of the image scene. Figure 2.18 displays solutions of the TV-L^1 model for different values of λ. Note that with a decreasing value for λ more and more structures move from the structural image to the image that contains the details.



(a) $f(x,y)$ **(b)** $u(x,y)$ **(c)** $v(x,y)$

Figure 2.17: Image decomposition with the TV-L^1 model.



(a) Original **(b)** $\lambda = 0.7$

(c) $\lambda = 0.5$ **(d)** $\lambda = 0.3$

Figure 2.18: Ability of the TV-L^1 model to remove fine structures.

Chapter 3

Experimental Setup and Protocol

Contents

3.1 Selected Methods for Feature Extraction

Due to the numerous types of image degradation and the diversity of images to be matched, it is still impossible to develop a general-purpose algorithm that is applicable to all matching tasks. Different features respond to different types of image structures and provide different levels of invariance. A naive approach would always pursue the highest level of invariance, however, an increased level of invariance usually implies a decreased level of discriminative power. As a rule of thumb, it is recommended to use the level of invariance that exactly meets the specific application scenario.

3.1.1 Feature Detectors

Table 3.1 summarizes the feature detectors presented in Section 2.2.1. The detectors are grouped according to their level of invariance: rotation, scale, or affine. Additionally, the most important properties (i.e. repeatability and accuracy) are compared. In the application scenario addressed in this thesis, there is no special requirement for any kind of invariance since meta information (such as flight direction, flying altitude, average ground level, etc.) is available for every single

	Category			Invariance			Localization	
Detector	Corner	Blob	Region	Rotation	Scale	Affine	Repeatability	Accuracy
Harris	☒	□	□	☒	□	□	☒☒☒	☒☒☒
Hessian	□	☒	□	☒	□	□	☒☒□	☒☒□
SUSAN	☒	□	□	☒	□	□	☒☒□	☒☒□
Harris-Lap.	☒	□	□	☒	☒	□	☒☒☒	☒☒☒
Hessian-Lap.	□	☒	□	☒	☒	□	☒☒☒	☒☒☒
DoG	□	☒	□	☒	☒	□	☒☒□	☒☒□
Fast-Hessian	□	☒	□	☒	☒	□	☒☒□	☒☒□
Harris-Aff.	☒	□	□	☒	☒	☒	☒☒☒	☒☒☒
Hessian-Aff.	□	☒	□	☒	☒	☒	☒☒☒	☒☒☒
EBR	☒	□	□	☒	☒	☒	☒☒☒	☒☒☒
MSER	□	□	☒	☒	☒	☒	☒☒☒	☒☒☒
IBR	□	□	☒	☒	☒	☒	☒☒□	☒☒□

Table 3.1: Summary of feature detectors (adapted from [59]).

image. This information can be used, for example, to pre-order the image sets (i.e. eliminate the need of rotation invariance), or to determine the ratio of flying altitudes (i.e. eliminate the need of scale invariance). The following feature detectors have been selected for multi-temporal performance evaluations.

Harris Detector The Harris detector [14] is one of the most popular and widely used interest point detectors. This detector searches corner-like structures and it has been shown [52] that it achieves stable results and a good spatial localization of the feature points.

Hessian Detector Beside corners, blobs are another intuitive image feature and blob detectors are widely used in numerous computer vision applications. Since there are no special requirements regarding invariance, the Hessian detector [5] is an obvious choice in this category.

MSER Detector Maximally Stable Extremal Regions (MSERs) [26] have proven to be highly repeatable [34]. Additionally, they are effectively detected with a watershed-like segmentation algorithm that runs at reasonable computational costs.

The selected detectors are more or less complementary, which means that they may be easily combined to incorporate different image structures. Thus, the image may be better covered with features and the overall performance is less dependent on the image content.

3.1.2 Feature Descriptors

Given the detected feature locations, the next step is to describe a region in the neighborhood of this location. A naive approach is to characterize the region patch by a vector of pixel intensities. On the one hand, this yields a high-dimensional (i.e. number of patch samples) vector which is a drawback for the computational complexity of the subsequent descriptor matching task. On the other hand, the local appearance of the patches is very likely to differ from image to image. Hence, it is preferable to build alternative descriptions of image patches. Different approaches

	Category			Characteristics	
Descriptor	Distribution	Filter	Other	Dimensionality	Performance
GLOH	☒	□	□	☒ ☒ ☒	☒ ☒ ☒
PCA-SIFT	☒	□	□	☒ ☒ □	☒ ☒ ☒
SIFT	☒	□	□	☒ ☒ ☒	☒ ☒ ☒
Spin Images	☒	□	□	☒ ☒ □	☒ ☒ □
SURF	☒	□	□	☒ ☒ □	☒ ☒ ☒
Complex Filters	□	☒	□	☒ □ □	☒ □ □
Diff. Invariants	□	☒	□	☒ □ □	☒ □ □
Steerable Filters[1]	□	☒	□	☒ ☒ ☒	☒ ☒ ☒
Cross Correlation	□	□	☒	☒ ☒ ☒	☒ ☒ □
Gradient Moments	□	□	☒	☒ □ □	☒ ☒ □

Table 3.2: Summary of feature descriptors (adapted from [47]).

have been proposed to describe image patches and their performances were compared, e.g., in [30; 33]. Table 3.2 summarizes a set of well-known feature descriptors. The table shows the assigned category along with important characteristics such as the dimensionality and a rating of the descriptor performance. The following feature descriptors have been selected for multi-temporal performance evaluations.

Scale Invariant Feature Transform (SIFT) The SIFT descriptor [25] is one of the most popular and widely used feature descriptors and it has been shown [30; 33] that it achieves state-of-the-art performances. The descriptor accumulates local gradient information in the neighborhood of the feature location into a 128 dimensional descriptor vector.

Speeded-up Robust Features (SURF) In [3] it is shown that Speeded-up Robust Features achieve highly competitive performances. Due to the use of integral images, the descriptor computation can be executed faster than for other state-of-the-art methods and due to the relatively low descriptor dimensionality (i.e. 64 dimensions) the time for feature matching is reduced. These properties make SURF features an appealing alternative to SIFT.

Steerable Filters Winder and Brown [63; 64] studied Steerable Filter[1] feature descriptors for image matching and 3D reconstruction. They used an automated learning algorithm in order to get reasonable parameter settings. In their work Steerable Filters achieved remarkable results with significantly lower error rates compared to SIFT.

[1] Note that Mikolajczyk and Schmid [33] used Steerable Filters up to the fourth order as low-dimensional descriptors. Here, Steerable Filters are used to compute high-dimensional histogrammed descriptors as proposed by Winder and Brown [63; 64].

3.2 Image Data Sets

All images are high resolution (i.e. 11500 × 7500 pixels) and the image sets capture a substantial amount of temporal variations. In each set, the image overlap is about 80% in flight direction (i.e. viewpoint change of about 10 degrees) and 60% for adjacent flight strips (i.e. viewpoint change of about 30 degrees). The flying altitude is about 1700 meters above the ground level. Both the position of the camera center (in GPS coordinates) and the orientation are known a priori, thus, the image sets are already ordered.

The test images are divided into the following three image sets which consist of four flight strips with four images per strip. Since images in the same strip are captured one after another there is no temporal impact for any given image pair. However, all cross-strip image pairs contain a varying amount of temporal variation.

The `DEN1` Image Set All images in this set were captured about noon in spring-time. The captured scene contains an industrial area (characterized by large buildings, spacious parking areas, a four-lane motorway network, etc.) and a residential area (characterized by small houses with front gardens and trees). For cross-strip image pairs, the changes in flying altitude are up to 40 meters. Three strip sequences were captured at the same day and they contain a cross-strip temporal variation of about 20 minutes. The fourth image strip was captured the other day and with two hours delay with respect to the other images. The main challenges regarding temporal variations are (i) moving objects in the industrial area and (ii) different drop shadow directions due to the position of the sun.

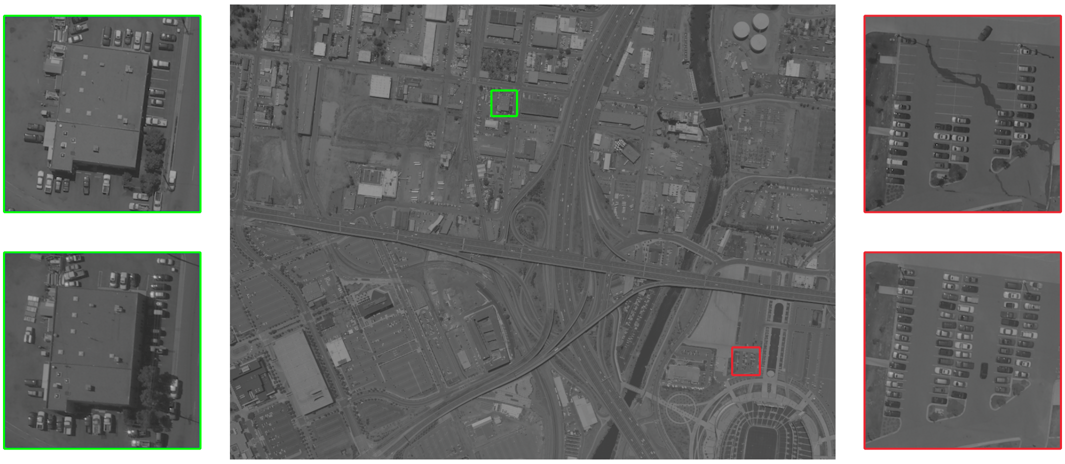


Figure 3.1: The `DEN1` image set.

The `DAL1` Image Set This image set captures a residential area mainly characterized by small houses, dense ground vegetation, and small parking areas. For cross-strip image pairs, the flying altitude changes are up to 30 meters. Three image strips of this set were captured in the afternoon of the same day in spring-time. These images contain temporal variations of up to 30 minutes. The fourth strip sequence was captured three days later in the morning. The main challenge of this data set is characterized by substantial appearance changes. The low position of the sun in the morning causes different shadow directions.

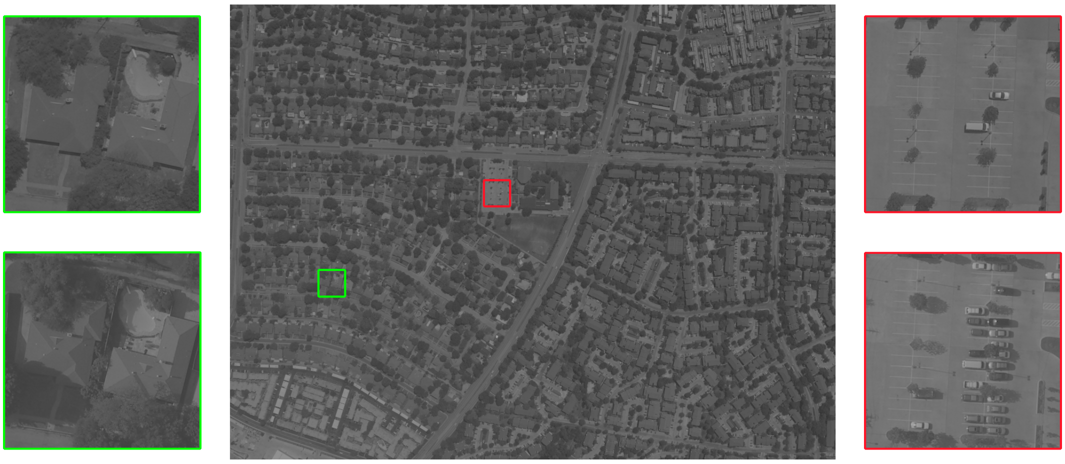


Figure 3.2: The DAL1 image set.

The DAL2 Image Set As the name indicates, this image set basically captures the same scene than the DAL1 set with only few large buildings and parking areas. For cross-strip image pairs, the flying altitude differences are up to 70 meters. The four image strips can be grouped into two pairs where each pair shows temporal variations of a few minutes only. However, the temporal variation between these two pairs is about 5 months (December to May). Subsequently, the main characteristic of this image set is a massive change in appearance due to (i) seasonal vegetation changes and (ii) drop shadows caused by the low position of the sun.

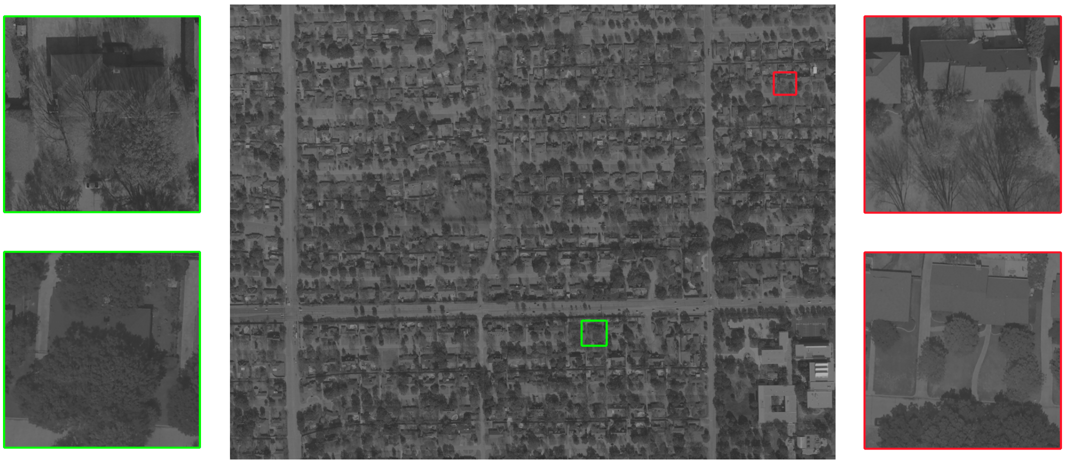


Figure 3.3: The DAL2 image set.

3.3 Performance Measures

This section introduces performance measures that will be used to quantitatively analyze the applied local detectors and descriptors. A lot of influential work regarding performance evaluations has been done by Mikolajczyk and Schmid [30; 31; 33; 34]. In order to be comparable with their evaluation work, the same measures are applied here. Additionally, the total number of corresponding features is computed. This value might be of interest because a large number of correspondences usually increases the accuracy of geometry estimation tasks.

3.3.1 Detector Evaluation

3.3.1.1 Repeatability Score

The basic property of any feature detector is its ability to detect features in a repeatable manner particularly in the presence of geometric or photometric changes. This property is measured with the repeatability score [51]

$$r_{ij} = \frac{|C_{ij}|}{\min(D_i, D_j)} \tag{3.1}$$

where C_{ij} denotes the set of correspondences between image I_i and image I_j, D_i and D_j represent the set of detections in I_i and I_j respectively, and $|\cdot|$ denotes cardinality.

According to Equation 3.1, the repeatability score defines the ratio of the number of corresponding detections to the minimum number of detections in either image. However, finding the number of correspondences is not an easy task and requires ground truth information for the given image pair (cf. Section 3.4).

In the case of planar image scenes a homography H (i.e. point-to-point transfer) can be used to compute correspondences. In [32] the correspondence of two points p_i, p_j is defined as

$$\|(H \cdot p_i) - p_j\| < 1.5 \ [pixels] \tag{3.2}$$

and two regions r_i, r_j are deemed to correspond if the overlap error

$$1 - \frac{r_i' \cap r_j}{r_i' \cup r_j} < 50 \ [\%] \tag{3.3}$$

where r_i' is r_i transferred into image I_j. For non-planar image scenes the ground truth geometry can be characterized with a fundamental matrix F. However, the fundamental matrix defines a point-to-line transfer between images (cf. Figure 3.6). Thus, a point correspondence for non-planar scenes is established if the Euclidean distance

$$dist(F \cdot p_i, p_j) < d_c \ [pixels] \tag{3.4}$$

from point p_j to the epipolar line $l = F \cdot p_i$ is less than d_c pixels.

Due to the large image dimensions (i.e. 11500 × 7500 pixels, cf. Section 3.2) it is not sufficient to search for correspondences along the epipolar line because typically there will occur more than one detection close to it. Since parameters like the camera position, orientation, and flying altitude are known for each image, it is possible to narrow down the search area along the epipolar line by simply projecting point p_i from one image to the other.

In order to obtain a reasonable restriction of the search area, an approximate estimation of the minimum and maximum projection height is required for each image pair. This is illustrated in Figure 3.4 where the search area along the epipolar line (blue) is significantly reduced (indicated by the yellow bounds). The values for the minimum and maximum projection height are collected by manually verifying the projection of selected points.

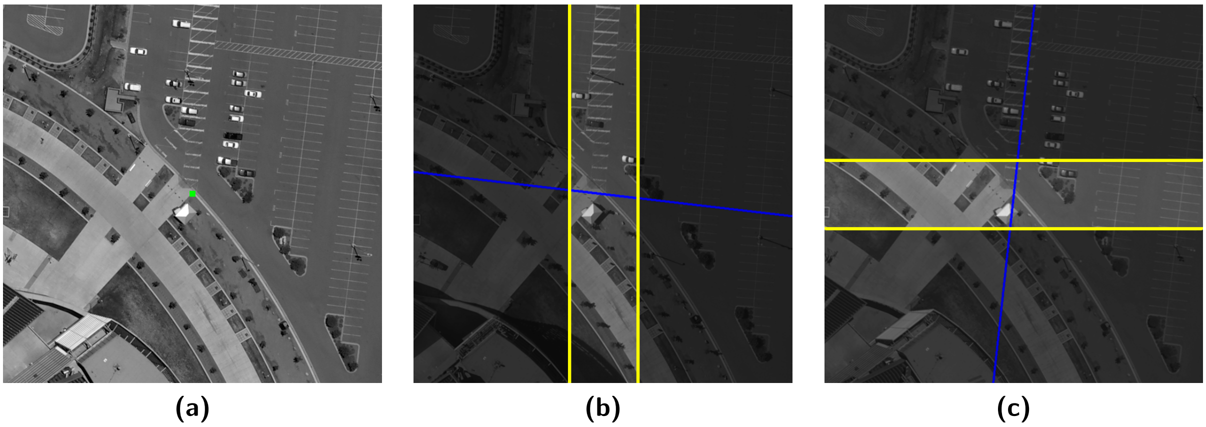
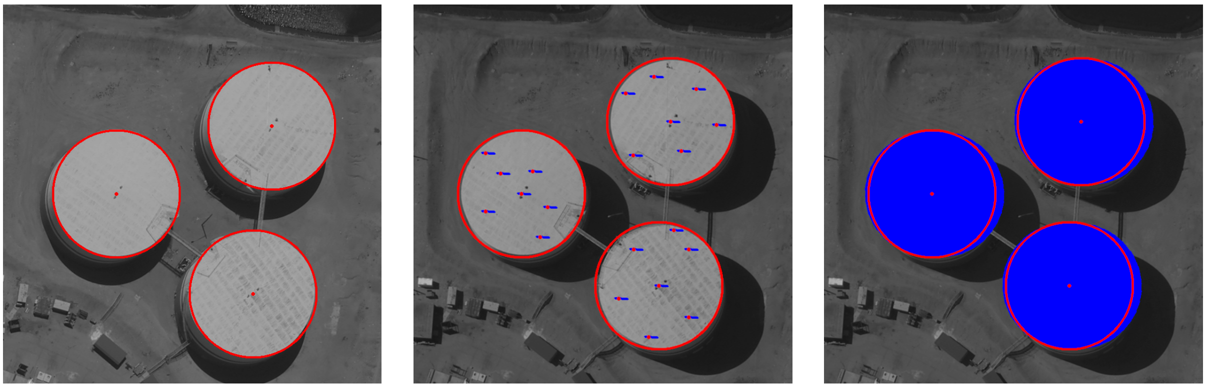


Figure 3.4: Restricted search area along epipolar line.

The transfer of elliptical regions between images pose more problems. First, the point-to-line transfer of every single region point gives a pencil of epipolar lines and does not allow to determine the location of the transferred region (assuming the same region shape). Second, the transfer is tightly related with the 3D image scene. If the ellipse represents a planar part of the image scene then the transfer yields an elliptical structure too, otherwise, the shape of the transferred region is arbitrary.

The first issue can be addressed again by applying projective constraints. Analogous to the point transfer, each pixel in region r_i is used to compute the corresponding epipolar line and the length of each line is limited by the minimum and maximum projection height (cf. Figure 3.5b). Subsequently, the union of all line segments represents the transferred region r_i' (i.e. the blue regions in Figure 3.5b). Since there are only translational and rotational changes between image views, the second issue is negligible because the lack of projective accuracy has a much stronger impact on the computed overlap error than the underlying 3D scene.



(a) Regions in Image I_i **(b)** Samples with restricted epipolar line **(c)** Transferred regions

Figure 3.5: Region transfer based on epipolar geometry.

3.3.2 Descriptor Evaluation

3.3.2.1 Matching Strategies

Descriptor evaluation is the task of verifying descriptor matches between two images. The definition of a match depends on the matching strategy.

Threshold-based Matching Two regions $R_\mathbf{A}$ and $R_\mathbf{B}$ are matched if the distance d between the corresponding descriptors $D_\mathbf{A}$ and $D_\mathbf{B}$ is below a threshold t_d. The definition of this matching strategy allows a descriptor to have several correct matches.

Nearest Neighbor Matching In this case, regions $R_\mathbf{A}$ and $R_\mathbf{B}$ are matched if the distance d between the corresponding descriptors $D_\mathbf{A}$ and $D_\mathbf{B}$ is below a threshold t_d and if $D_\mathbf{B}$ is the nearest neighbor of $D_\mathbf{A}$. This definition allows a descriptor to have only a single correct match.

Nearest Neighbor Matching with Distance Ratio This approach is closely related to the nearest neighbor matching. The only difference is that thresholding is applied to the distance ratio between the two nearest neighbors $D_\mathbf{B}$ and $D_\mathbf{C}$. Thus, two regions $R_\mathbf{A}$ and $R_\mathbf{B}$ are matched if the distance ratio $\|D_\mathbf{A} - D_\mathbf{B}\| / \|D_\mathbf{A} - D_\mathbf{C}\|$ is below a threshold t_r. Again, this definition allows a descriptor to have only a single correct match.

Mikolajczyk and Schmid [33] show that the ranking of the descriptors is quite similar across all three matching strategies. However, the approaches based on nearest neighbor matching select only the best match which yields more correct matches than simple thresholding.

3.3.2.2 Recall vs. 1-Precision

The performance evaluation is derived from the number of true and false matches for any given image pair and it is strongly associated with the evaluation criteria used by Ke and Sukthankar [18] and Mikolajczyk and Schmid [33]. The final performance score is expressed with recall vs. 1-precision graphs which are generated as follows. For each descriptor in the reference image nearest neighbor matching is applied and the total number of candidate matches is counted. Subsequently, these initial matches are verified using epipolar geometry. A candidate match is correct if the nearest neighbor key point does not deviate more than d_{eg} pixels from the corresponding epipolar line. This verification step divides the number of candidate matches into the final number of true and false matches. Since it is not possible to determine the true number of correspondences, this value is approximated by the number of correspondences that is computed with the repeatability score. With the number of true and false matches, the number of candidate matches, and the approximated number of correspondences, is is possible to compute recall vs. 1-precision values as

$$recall = \frac{\#\ true\ matches}{\#\ correspondences} \tag{3.5}$$

and

$$1 - precision = \frac{\#\ false\ matches}{\#\ candidate\ matches}. \tag{3.6}$$

In order to generate a curve the nearest neighbor distance threshold value t_d is varied. Recall measures the fraction of true matches out of all possible matches, while 1-precision measures the fraction of false matches out of all obtained matches. Thus, it is desirable to both maximize recall and minimize 1-precision.

3.4 Ground Truth Generation

The objective of the evaluations discussed below is to measure the performance of both local detectors and descriptors *independently*. In the latter case, performance analysis can be done with (robust) descriptor matching. However, matching is not appropriate for the automatic evaluation of feature detectors, because the final score is not fully decoupled from the descriptor. Therefore, the evaluation of any feature detector requires ground truth (GT) information on the geometric constraints of the captured image scene.

Ground truth data is generated by establishing epipolar constraints between views [15]. For each detected feature in the first image, nearest neighbor matching based on the distance ratio test (cf. Section 3.3.2) is applied. Two features are selected as a candidate match, if the distance ratio between the nearest and the second nearest feature is less than 30%. The set of potential matches is then verified by a robust estimation of the epipolar constraints based on the RANdom SAmple Consensus (RANSAC) algorithm [8]. A candidate match is correct if the locations of the features do not deviate more than 0.5 pixels from the estimated epipolar line. This verification step divides the set of candidate matches into subsets of true and false matches. In order to further improve the estimation quality RANSAC is applied to the set of true matches to finally estimate the fundamental matrix.

The quality of the ground truth estimation procedure directly effects the evaluation results. Hence, the accuracy of the established correspondences is crucial. The initial set of features is extracted with the rotation-variant version of the SURF detector/descriptor scheme [3] for the following reasons. First, the goal is to find highly accurate correspondences, thus, a detector that offers sub-pixel accuracy is necessary. Second, the procedure should not necessitate any assumptions regarding image content or detection scale. Therefore, a scale-invariant detector is preferable. Third, SURF descriptors can be computed and matched at lower computational costs compared to, e.g., SIFT.

The automatic ground truth estimation procedure completely failed for *all* cross-strip image pairs with seasonal variations in the DAL2 image set (cf. Section 3.2). In this case the correspondences were established by manually finding at least 128 corresponding locations in adjacent images. Figure 3.6 visualizes the computed ground truth epipolar geometry for two different image pairs.

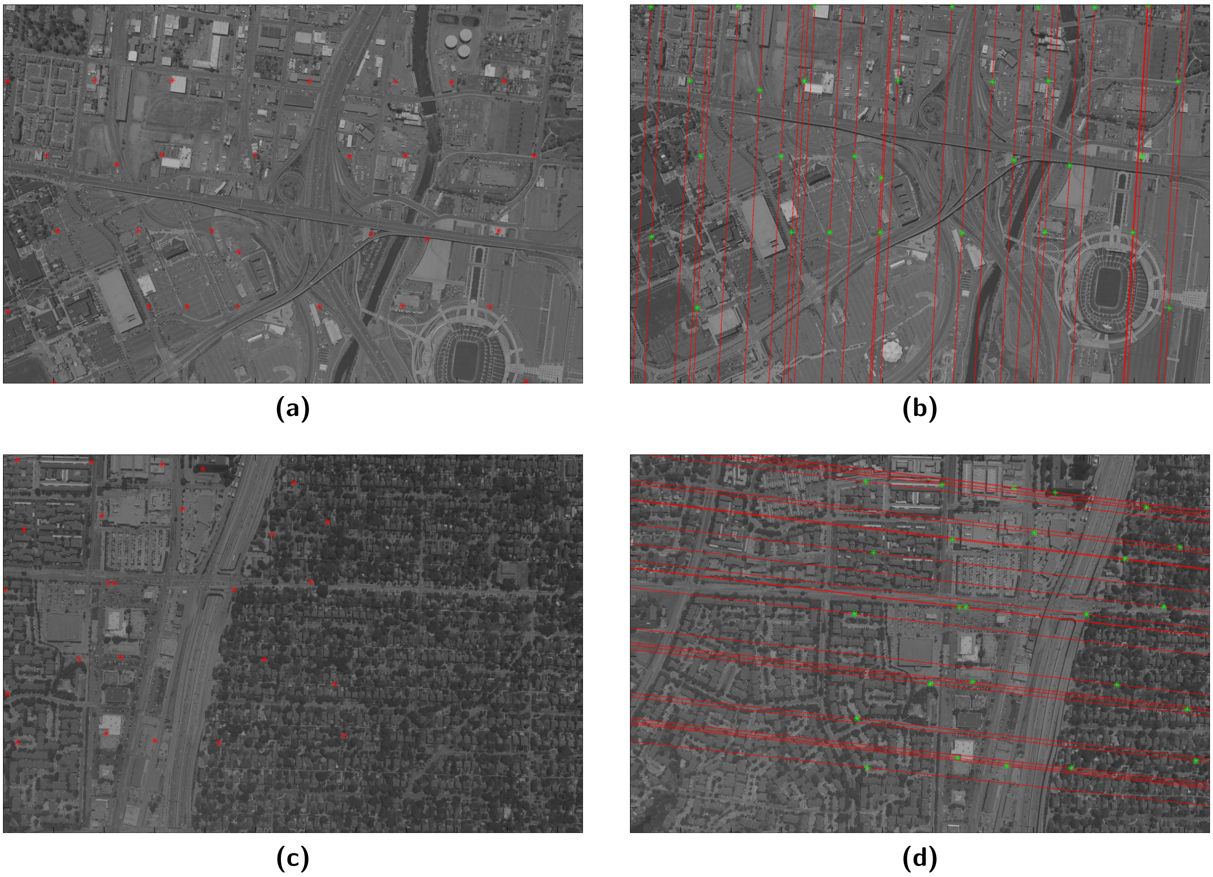


Figure 3.6: Visualization of ground truth epipolar geometry.

3.5 Detector Evaluation

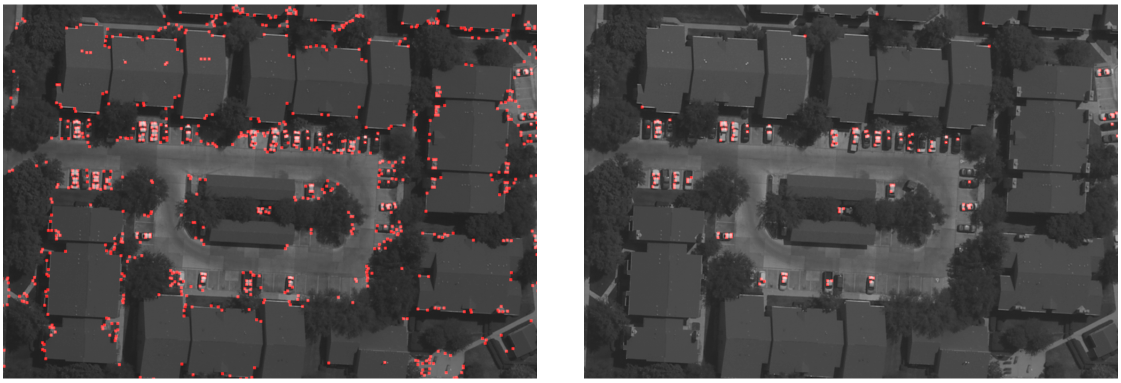
The feature detectors evaluated in this section are Harris corners [14], Hessian detector [5], and Maximally Stable Extremal Regions [26]. For all detectors the binaries provided by Mikolajczyk [28] are used.

3.5.1 General Detection Trends

This first set of evaluations aims at measuring the basic trends for different detector settings particularly regarding response threshold value, detection scales, and absolute number of detections.

3.5.1.1 Harris Detector

The detections for scale $\sigma_1 = 1.4$ and different cornerness thresholds are visualized in Figure 3.7. In general, the results depend on the captured image scene. In all cases, the strongest detections are generated by vehicles. Nearly all detections with a corner strength greater than 4096 appear either at the bodywork, the windshield, or the drop shadow of the vehicle. Only a small portion of the detections appear on "static" objects like rooftops or swimming pools. Decreasing the cornerness threshold value usually raises the number of detections. The detections added this way increasingly arise from corners of houses and from shadows.



(a) Low threshold **(b)** High threshold

Figure 3.7: Harris detections with different cornerness thresholds.

The previous tests with different cornerness threshold values are based on a detection scale σ_1 of 1.4. Now, the objective is to analyze the feature locations for increasing detection scales $\sigma_n = s^n$, where the scale factor s is set to 1.4 and $n \in \{0, 1, 2, \ldots, 10\}$. For detection scales up to σ_3, the feature locations do not significantly change with respect to the observations presented above. However, starting with σ_4 fewer detections come from vehicles and more features arise near corners of buildings. Tables 3.3 to 3.5 show the minimum, maximum, and mean number of detected corners for any combination of cornerness thresholds and detection scales.

Detection	Threshold 1024			Threshold 2048			Threshold 4096		
Scale	min	max	mean	min	max	mean	min	max	mean
$\sigma_0 = 1.00$	26726	37976	32249	14121	23143	18622	7271	13262	10145
$\sigma_1 = 1.40$	19096	26915	23082	10884	17155	13871	5896	10966	8274
$\sigma_2 = 1.96$	11824	16269	14186	6681	10473	8423	3513	6798	5075
$\sigma_3 = 2.74$	7247	9919	8696	4008	6064	4992	2020	3653	2786
$\sigma_4 = 3.84$	4624	6146	5464	2288	3319	2812	989	1566	1264
$\sigma_5 = 5.38$	2520	3387	2991	1117	1526	1312	420	562	498
$\sigma_6 = 7.53$	1273	1672	1449	539	709	600	194	245	221
$\sigma_7 = 10.54$	662	897	735	265	341	293	92	140	113
$\sigma_8 = 14.76$	313	547	390	132	206	166	41	80	63
$\sigma_9 = 20.66$	150	292	208	66	118	95	18	45	33
$\sigma_{10} = 28.93$	112	164	138	41	84	61	7	31	19

Table 3.3: Number of Harris detections for `DEN1`.

Detection	Threshold 1024			Threshold 2048			Threshold 4096		
Scale	min	max	mean	min	max	mean	min	max	mean
$\sigma_0 = 1.00$	8771	20516	15848	4537	12224	9189	1900	7041	4870
$\sigma_1 = 1.40$	7482	16698	13064	4223	10136	7947	2077	6080	4529
$\sigma_2 = 1.96$	5445	11026	8935	3053	6335	5268	1625	3899	3079
$\sigma_3 = 2.74$	3701	6953	5705	1972	3723	3094	1014	2035	1651
$\sigma_4 = 3.84$	2438	4304	3593	1249	2122	1752	566	877	727
$\sigma_5 = 5.38$	1601	2422	2068	747	945	797	234	370	295
$\sigma_6 = 7.53$	969	1293	1070	326	519	405	117	218	171
$\sigma_7 = 10.54$	517	723	587	183	292	233	77	140	110
$\sigma_8 = 14.76$	302	448	363	105	181	140	41	81	61
$\sigma_9 = 20.66$	158	249	190	57	89	67	17	31	25
$\sigma_{10} = 28.93$	86	125	102	31	62	43	4	39	20

Table 3.4: Number of Harris detections for `DAL1`.

Detection	Threshold 1024			Threshold 2048			Threshold 4096		
Scale	min	max	mean	min	max	mean	min	max	mean
$\sigma_0 = 1.00$	6599	15853	10981	3130	7355	5079	1340	3329	2179
$\sigma_1 = 1.40$	5071	12986	8910	2572	6001	4191	1208	2708	1868
$\sigma_2 = 1.96$	3380	9595	6352	1623	4290	2915	716	1825	1263
$\sigma_3 = 2.74$	2200	6909	4440	977	2881	1926	404	1143	776
$\sigma_4 = 3.84$	1496	4905	3145	627	1984	1290	235	700	461
$\sigma_5 = 5.38$	1063	3501	2204	382	1314	829	137	414	268
$\sigma_6 = 7.53$	770	2471	1544	263	905	549	85	255	165
$\sigma_7 = 10.54$	528	1593	1011	181	556	352	43	147	92
$\sigma_8 = 14.76$	349	1022	633	116	345	209	21	93	51
$\sigma_9 = 20.66$	184	567	360	50	160	102	8	40	24
$\sigma_{10} = 28.93$	87	325	199	24	85	53	3	23	13

Table 3.5: Number of Harris detections for `DAL2`.

3.5.1.2 Hessian Detector

Again, the detection scales are defined by $\sigma_n = s^n$, where the scale factor s is set to 1.4, and $n \in \{0, 1, 2, \ldots, 10\}$. For scale values from σ_0 to σ_2 a large part of the detections is caused by high-frequency distortions (e.g. reflections of the sunlight on water surface, or patches of high contrast on concrete ceilings). With an increasing scale (σ_2 to σ_6) more and more detections arise from vehicles, flue-like structures on top of buildings, and shrubbery. Starting with σ_7, the majority of detections is based on rooftops, shrubbery, and shadows. Tables 3.6, 3.7, and 3.8 list the minimum, maximum, and mean number of detections for several combinations of response strength thresholds and detection scales.

3.5.1.3 MSER Detector

The Maximally Stable Extremal Regions (MSER) detector is mainly driven by the following two parameters.

Minimum Region Size A small value for the minimum region size (MS) achieves a relatively large number of detections. However, very often such small regions evolve from high contrast patches on concrete ceilings or road markings. Cars are detected up to parameter values of 256 (though, dark colored cars combined with shadows can still appear for region sizes up to 768). Regions with sizes starting from 512 are typically detected at swimming pools, pavements, or rooftops.

Minimum Margin The minimum margin (MM) parameter Δ (cf. Section 2.2.1) controls the stability of the regions. This parameter has a strong impact on the number of detected regions. Low values detect a higher number (but less stable) of regions, whereas high values detect only a few regions that are very stable.

Moreover, considering only bright extremal regions (i.e. MSER-) increases the repeatability rate significantly, because usually a large portion of dark regions arise from shadowed regions. The repeatability for bright regions systematically outperforms the scores achieved with all (bright and dark) regions. Therefore, all repeatability graphs presented in this section show performances for bright regions only.

The number of detected regions is closely related with the captured image scene, the specified minimum region size, and the defined minimum margin threshold. Tables 3.9 to 3.11 show the number of detected MSERs for several combinations of MS and MM values.

Detection	Threshold 64			Threshold 128			Threshold 256		
Scale	min	max	mean	min	max	mean	min	max	mean
$\sigma_0 = 1.00$	7327	30399	17175	3726	6750	5336	1084	1837	1468
$\sigma_1 = 1.40$	10880	37071	22412	5855	9978	7925	1251	2366	1832
$\sigma_2 = 1.96$	9154	27146	17432	5490	9597	7463	919	2304	1617
$\sigma_3 = 2.74$	5702	16853	10973	3700	6843	5158	954	2276	1607
$\sigma_4 = 3.84$	3943	11177	7408	2485	4630	3509	643	1517	1078
$\sigma_5 = 5.38$	2391	7196	4655	1414	2466	1914	225	431	337
$\sigma_6 = 7.53$	1178	4216	2546	638	963	810	76	138	100
$\sigma_7 = 10.54$	525	2393	1304	277	380	322	20	36	28
$\sigma_8 = 14.76$	234	1088	609	109	152	128	4	14	8
$\sigma_9 = 20.66$	113	628	316	55	81	66	2	8	4
$\sigma_{10} = 28.93$	72	386	182	30	49	39	1	7	3

Table 3.6: Number of Hessian detections for `DEN1`.

Detection	Threshold 64			Threshold 128			Threshold 256		
Scale	min	max	mean	min	max	mean	min	max	mean
$\sigma_0 = 1.00$	1823	13919	7151	1043	4174	2615	218	1683	921
$\sigma_1 = 1.40$	3148	18509	10425	1696	5376	3676	256	1888	1035
$\sigma_2 = 1.96$	3342	15597	9436	1878	5169	3798	220	1603	880
$\sigma_3 = 2.74$	2642	10729	6781	1677	4221	3223	205	1315	759
$\sigma_4 = 3.84$	1869	7771	4756	1221	2832	2237	207	1012	642
$\sigma_5 = 5.38$	1190	5130	3109	740	1454	1222	106	333	215
$\sigma_6 = 7.53$	704	2510	1598	357	477	434	23	54	37
$\sigma_7 = 10.54$	337	1216	764	133	231	176	6	17	12
$\sigma_8 = 14.76$	195	617	406	91	165	125	4	11	8
$\sigma_9 = 20.66$	130	380	263	53	110	82	2	8	6
$\sigma_{10} = 28.93$	83	249	164	32	62	43	4	10	7

Table 3.7: Number of Hessian detections for `DAL1`.

Detection	Threshold 64			Threshold 128			Threshold 256		
Scale	min	max	mean	min	max	mean	min	max	mean
$\sigma_0 = 1.00$	1652	12548	5485	856	2204	1446	167	355	250
$\sigma_1 = 1.40$	2532	15828	7384	1341	3228	2168	220	475	346
$\sigma_2 = 1.96$	2271	12155	5921	1200	2708	1905	178	454	307
$\sigma_3 = 2.74$	1591	7977	3982	911	1911	1383	153	328	236
$\sigma_4 = 3.84$	1087	5500	2728	564	1266	917	121	245	181
$\sigma_5 = 5.38$	684	4048	1915	310	888	600	44	127	85
$\sigma_6 = 7.53$	392	2936	1313	161	514	337	16	66	40
$\sigma_7 = 10.54$	234	2041	899	115	335	221	8	43	25
$\sigma_8 = 14.76$	175	1359	628	87	223	149	5	34	17
$\sigma_9 = 20.66$	131	932	454	67	181	113	4	27	13
$\sigma_{10} = 28.93$	101	593	304	47	99	66	0	10	5

Table 3.8: Number of Hessian detections for `DAL2`.

Minimum	**Minimum Size 128**			**Minimum Size 256**			**Minimum Size 512**		
Margin	**min**	**max**	**mean**	**min**	**max**	**mean**	**min**	**max**	**mean**
$\Delta = 8$	4367	5373	4762	2019	2541	2248	1010	1407	1193
$\Delta = 12$	1914	2458	2122	852	1056	946	373	574	472
$\Delta = 16$	873	1146	993	360	511	432	169	283	215
$\Delta = 20$	399	531	469	178	246	213	77	144	106

Table 3.9: Number of MSER detections for `DEN1`.

Minimum	**Minimum Size 128**			**Minimum Size 256**			**Minimum Size 512**		
Margin	**min**	**max**	**mean**	**min**	**max**	**mean**	**min**	**max**	**mean**
$\Delta = 8$	2060	3176	2760	1136	1517	1334	621	769	655
$\Delta = 12$	780	1512	1191	404	558	487	173	276	215
$\Delta = 16$	366	761	566	175	262	219	73	146	101
$\Delta = 20$	188	383	281	88	130	104	43	75	56

Table 3.10: Number of MSER detections for `DAL1`.

Minimum	**Minimum Size 128**			**Minimum Size 256**			**Minimum Size 512**		
Margin	**min**	**max**	**mean**	**min**	**max**	**mean**	**min**	**max**	**mean**
$\Delta = 8$	1591	2410	2036	1024	1409	1226	622	761	697
$\Delta = 12$	588	893	731	347	479	409	189	227	210
$\Delta = 16$	233	376	298	124	190	158	63	86	73
$\Delta = 20$	94	182	129	47	88	67	21	33	29

Table 3.11: Number of MSER detections for `DAL2`.

3.5.2 Repeatability Scores

The number of detected features has a direct impact on the repeatability score of the detector. If there is a huge number of detections, the image may be cluttered with features such that correspondences will be established accidentally rather than intentionally. In order to avoid this, non-maximum suppression will be used in a pre-processing step for the Harris and Hessian detector. The suppression radius is set to 16 pixels

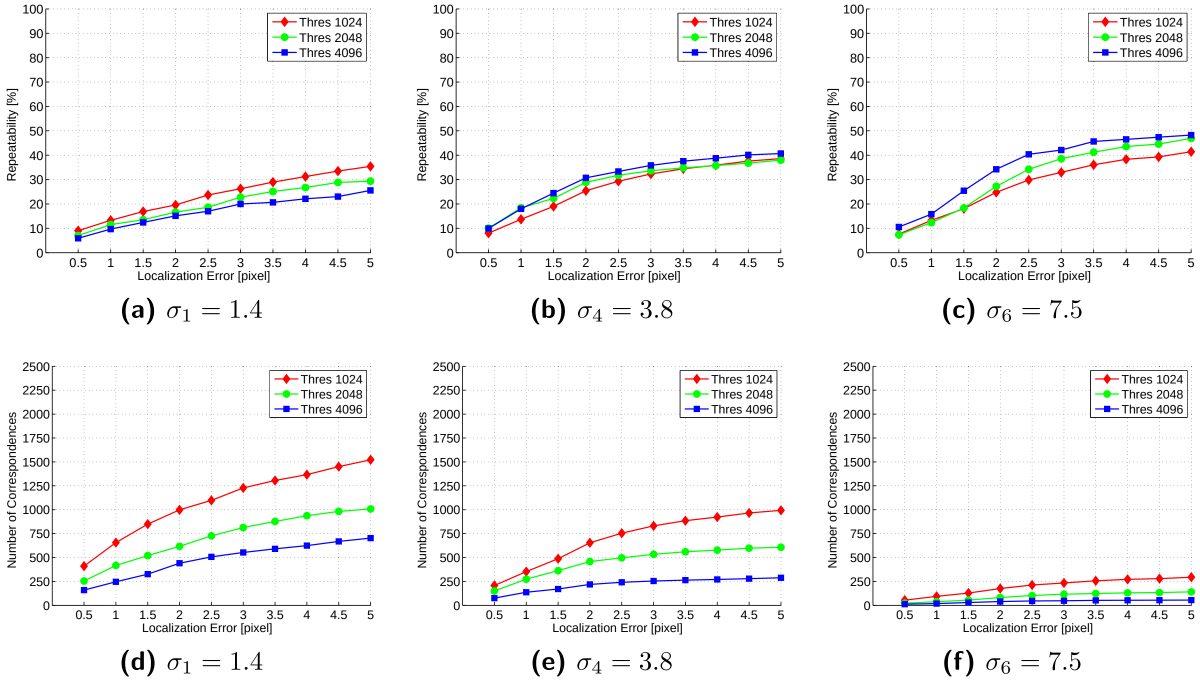
3.5.2.1 The `DEN1` Image Set

The temporal variations in this data set come rather from moving objects than from illumination or shadow changes.

Harris Detector The repeatability rate ranges from around 7% for a localization error of 0.5 pixel to 50% for a localization error of 5 pixels. The achieved repeatability rate in previous evaluations [11; 12] lies between 20% and 30% for a viewpoint angle of 30 degrees and a localization error of 1.5 pixels. These performances are comparable to the results presented in Figure 3.8. For the detection scale σ_1 (Figure 3.8a), the highest detector response threshold (i.e. 4096) achieves the worst repeatability rate. This exactly reflects the observation from Section 3.5.1, where the strongest detections are generated by dynamic structures such as moving vehicles. For the increased detection scale σ_6 in Figure 3.8c the ranking is reversed and the highest response threshold achieves the best repeatability rates. However, the total number of correspondences decreases significantly with an increasing detection scale (cf. Figures 3.8d to 3.8f).

Hessian Detector The repeatability scores range from around 3% for a localization error of 0.5 pixel to 50% for a localization error of 5 pixels. From previous evaluations [11; 12] a repeatability rate of 25% to 30% may be expected for a localization error of 1.5 pixels. This is the case for the detection scales $\sigma_4 = 3.8$ (Figure 3.9b) and $\sigma_6 = 7.5$ (Figure 3.9c). However for $\sigma_1 = 1.4$ the repeatability is considerably lower (10%). The reason for this is that a large part of the detections is caused by high-frequency distortions such as reflections of the sunlight on water surface. Just like for the Harris detector, the number of correspondences decreases with an increasing detection scale.

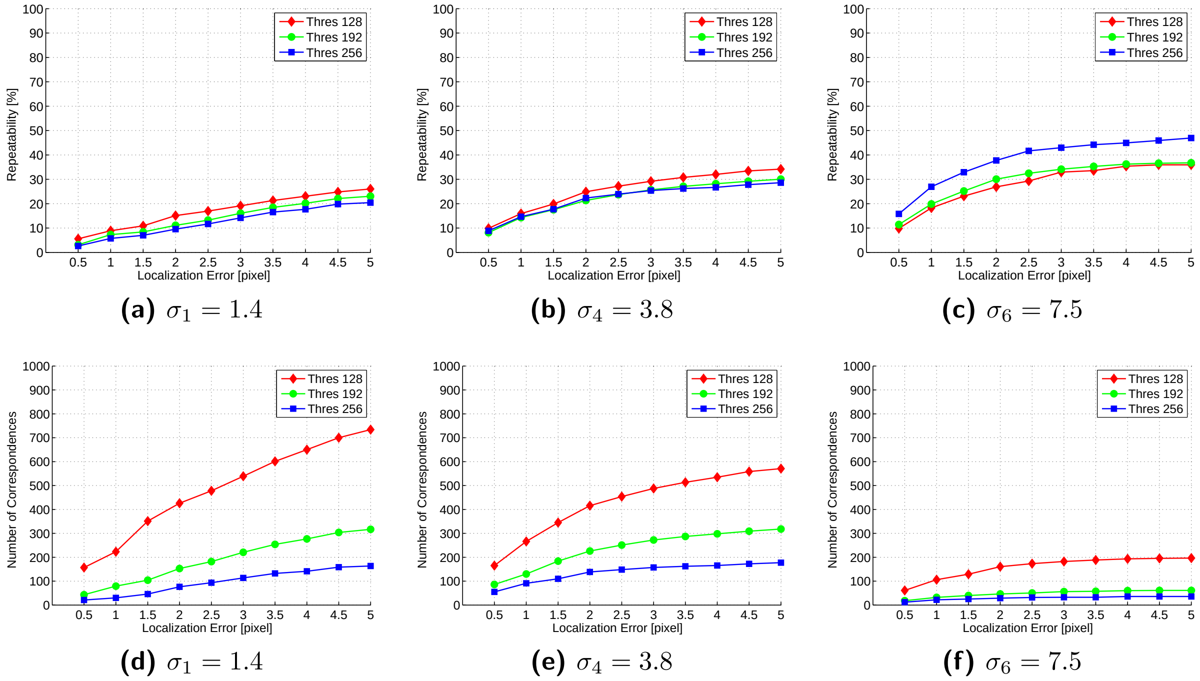
MSER Detector Figure 3.10 shows the repeatability score and the total number of correspondences for different parameter settings of the MSER detector. With respect to different overlap errors, the repeatability score ranges from 10% for small errors to 60% for large overlap errors. The repeatability scores in previous evaluations [11; 12] range from 25% to 30% for a viewpoint angle of 30 degrees and an overlap error of 50%. Surprisingly, these performances are below the results presented in Figure 3.10 where the achieved repeatability scores range from 45% to 55%. Generally, MSERs with a minimum region size of 1024 pixels achieve the best repeatability rates. The stability parameter Δ is clearly sensitive to the captured image scene. When Δ is increased from 8 to 12, the repeatability rates rise about 4% in average. However, further increasing Δ does not improve repeatability. The best repeatability values are measured with a minimum region size of 1024 and the minimum margin parameter Δ set to 12.



(a) $\sigma_1 = 1.4$ **(b)** $\sigma_4 = 3.8$ **(c)** $\sigma_6 = 7.5$

(d) $\sigma_1 = 1.4$ **(e)** $\sigma_4 = 3.8$ **(f)** $\sigma_6 = 7.5$

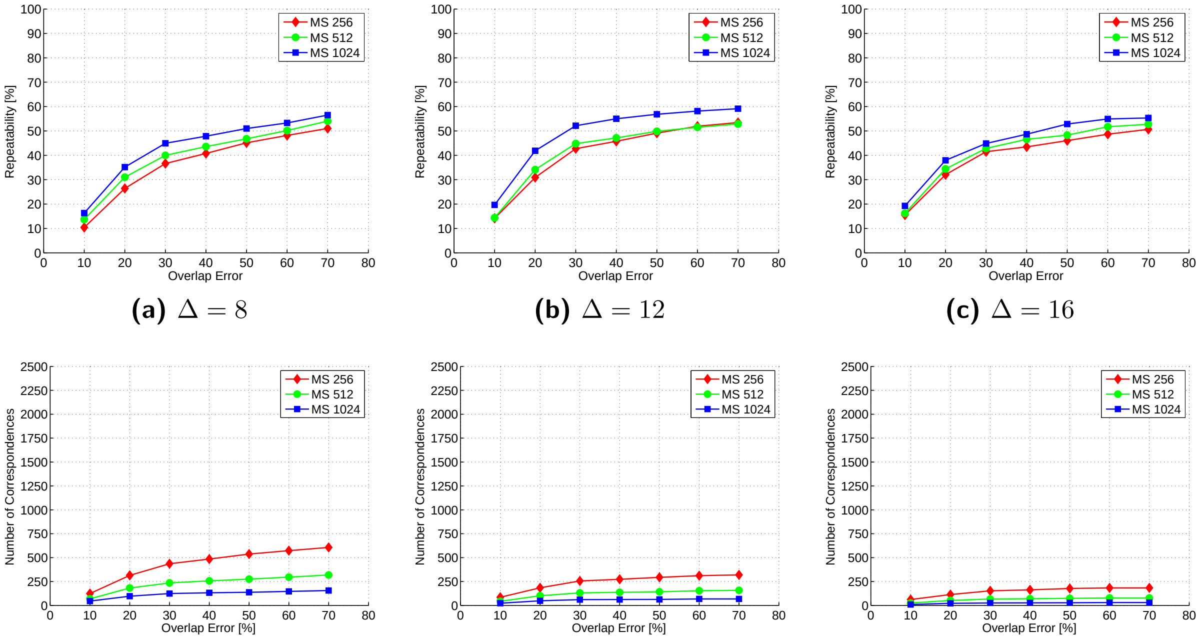
Figure 3.8: Harris repeatability rates for `DEN1`.



(a) $\sigma_1 = 1.4$ **(b)** $\sigma_4 = 3.8$ **(c)** $\sigma_6 = 7.5$

(d) $\sigma_1 = 1.4$ **(e)** $\sigma_4 = 3.8$ **(f)** $\sigma_6 = 7.5$

Figure 3.9: Hessian repeatability rates for `DEN1`.



(a) $\Delta = 8$ **(b)** $\Delta = 12$ **(c)** $\Delta = 16$

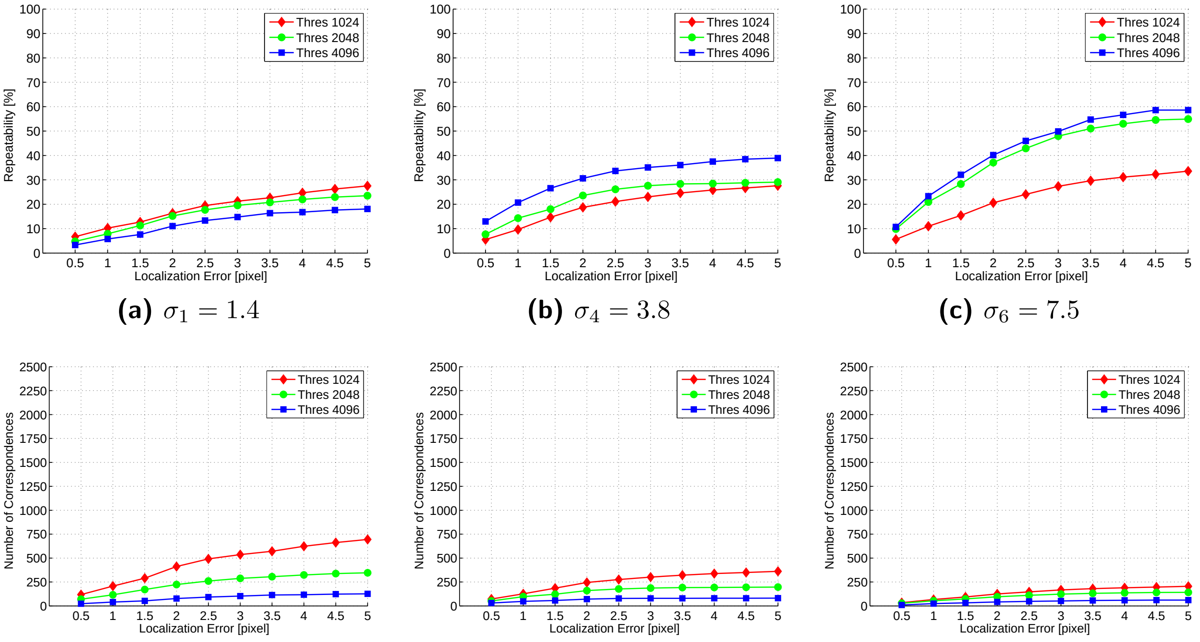
(d) $\Delta = 8$ **(e)** $\Delta = 12$ **(f)** $\Delta = 16$

Figure 3.10: MSER repeatability rates for `DEN1`.

3.5.2.2 The DAL1 Image Set

This data set is dominated by significant changes in appearance of shadowed regions. Large parts that are clearly visible in one image are covered with shadow in the other image.

Harris Detector The first point to notice is that there is a massive difference regarding the number of detected correspondences compared to the DEN1 set. However, this is not a big surprise since there are rather different numbers of detections (cf. Tables 3.3 and 3.4 respectively). In terms of repeatability, the trend is not so clear. For detection scale $\sigma_1 = 1.4$ repeatability decreases by 5%. For higher scales repeatability increases especially for higher response thresholds. The best repeatability curve is measured with detection scale $\sigma_6 = 7.5$ and the response strength threshold set to 4096. Figure 3.11 lists the repeatability scores and number of correspondences computed for this image set.



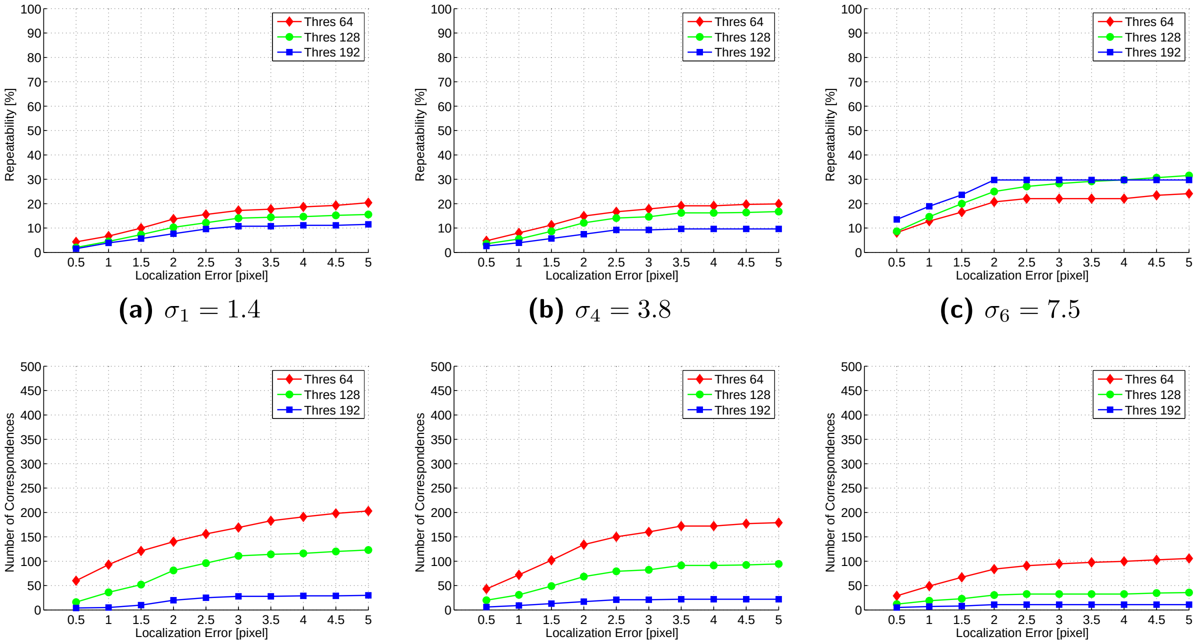
(a) $\sigma_1 = 1.4$ **(b)** $\sigma_4 = 3.8$ **(c)** $\sigma_6 = 7.5$

(d) $\sigma_1 = 1.4$ **(e)** $\sigma_4 = 3.8$ **(f)** $\sigma_6 = 7.5$

Figure 3.11: Harris repeatability rates for DAL1.

Hessian Detector Basically, the previous observations for the Harris detector also hold for the Hessian. The number of correspondences is considerably lower than in the DEN1 set. However, in this case the computed repeatability scores are lower for all detection scales. The results are listed in Figure 3.12. The best repeatability curve is measured with a detector response strength of at least 192 at scale $\sigma_6 = 7.5$.

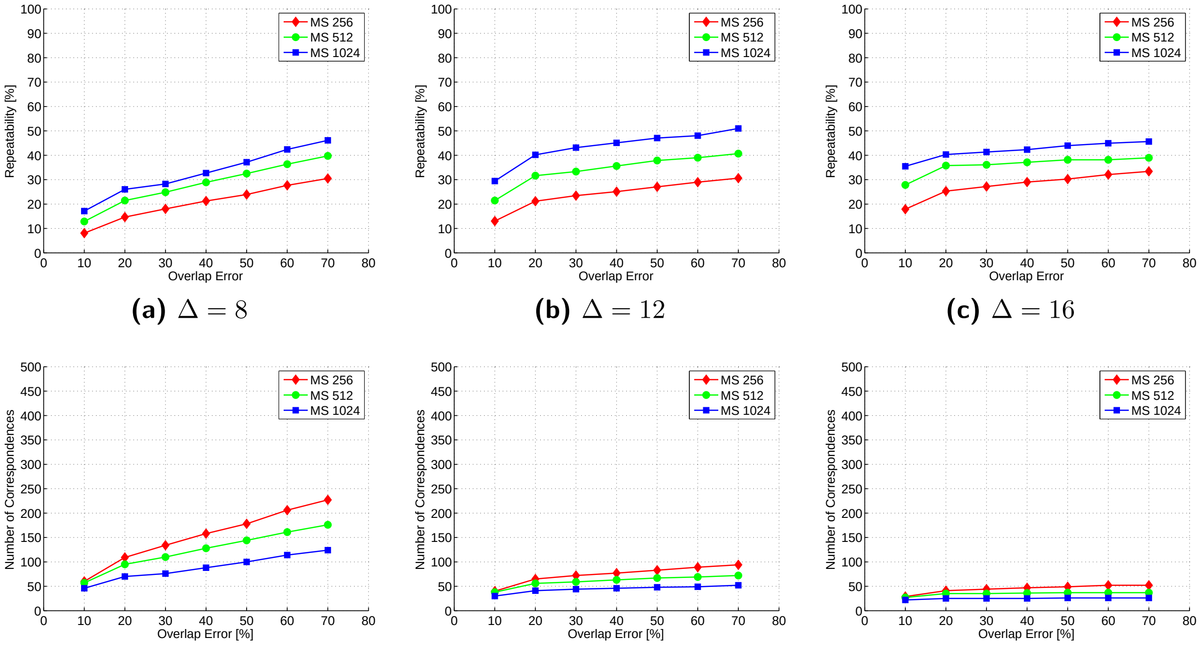
MSER Detector The repeatability rates are a bit lower than the values computed for the DEN1 set. The number of correspondences decreases relatively by 50% (e.g. from 250 to 120 for a minimum region size of 256 pixels and $\Delta = 8$). Nevertheless, MSERs achieve the most promising results among all detectors for this image set. Again, the best repeatability curves are measured with MSERs with a minimum size of 1024 pixels and Δ set to 12.



(a) $\sigma_1 = 1.4$ **(b)** $\sigma_4 = 3.8$ **(c)** $\sigma_6 = 7.5$

(d) $\sigma_1 = 1.4$ **(e)** $\sigma_4 = 3.8$ **(f)** $\sigma_6 = 7.5$

Figure 3.12: Hessian repeatability rates for `DAL1`.



(a) $\Delta = 8$ **(b)** $\Delta = 12$ **(c)** $\Delta = 16$

(d) $\Delta = 8$ **(e)** $\Delta = 12$ **(f)** $\Delta = 16$

Figure 3.13: MSER repeatability rates for `DAL1`.

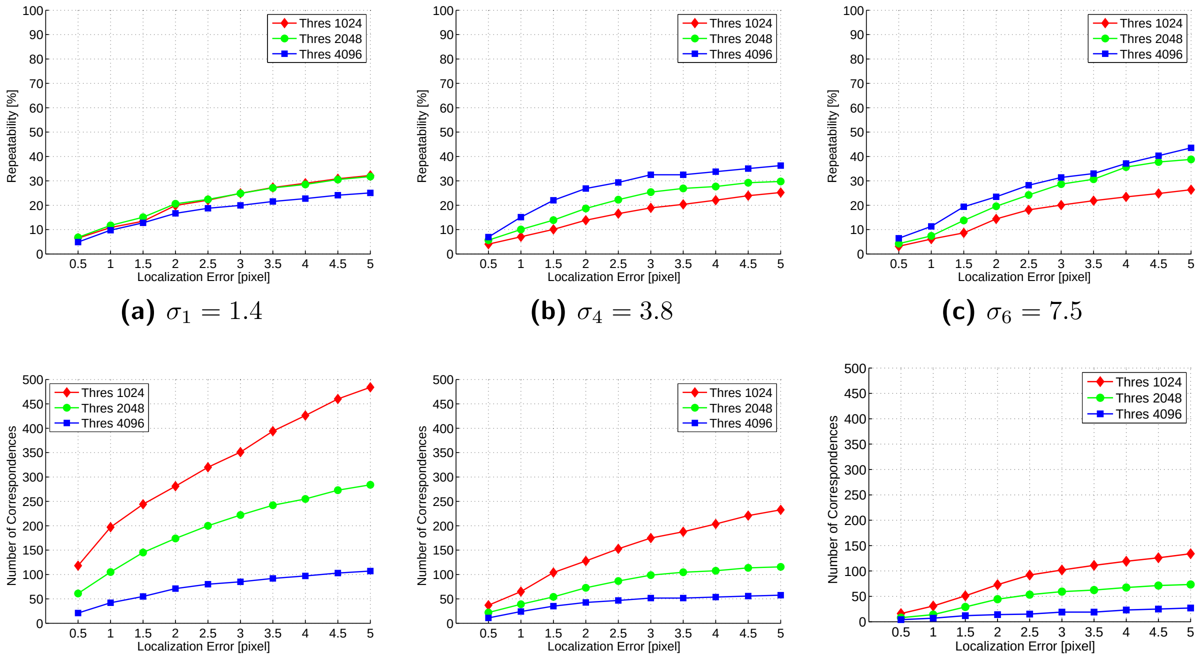
3.5.2.3 The DAL2 Image Set

This data set captures temporal variations of about 5 months. Hence, it is characterized by massive appearance changes due to seasonal vegetation and shadows caused by the low position of the sun.

Harris Detector Figure 3.14 presents the repeatability scores along with the number of correspondences detected with Harris. Compared to the previous evaluation results for the DAL1 image set, the achieved repeatability scores are generally lower (except for detection scale $\sigma_1 = 1.4$ where the results are similar). Since these two image sets basically capture the same image scene, the loss of repeatability is caused be the temporal variations of the DAL2 set. In contrast to the results of the DAL1 image set, the performance does not improve for higher detection scales. Also note that the total number of precise correspondences vanishes for higher detection scales.

Hessian Detector With respect to different localization errors, the repeatability scores range from 5% to 30%. In contrast to the Harris detector the achieved repeatability rates are similar compared to the results for the DAL1 set. Especially for the higher detection scales $\sigma_4 = 3.8$ and $\sigma_6 = 7.5$ the seasonal variations have less impact to this detector than to the Harris detector. However, in this case the number of precise correspondences (i.e. localization error ≤ 1.0 pixels) drops below 100.

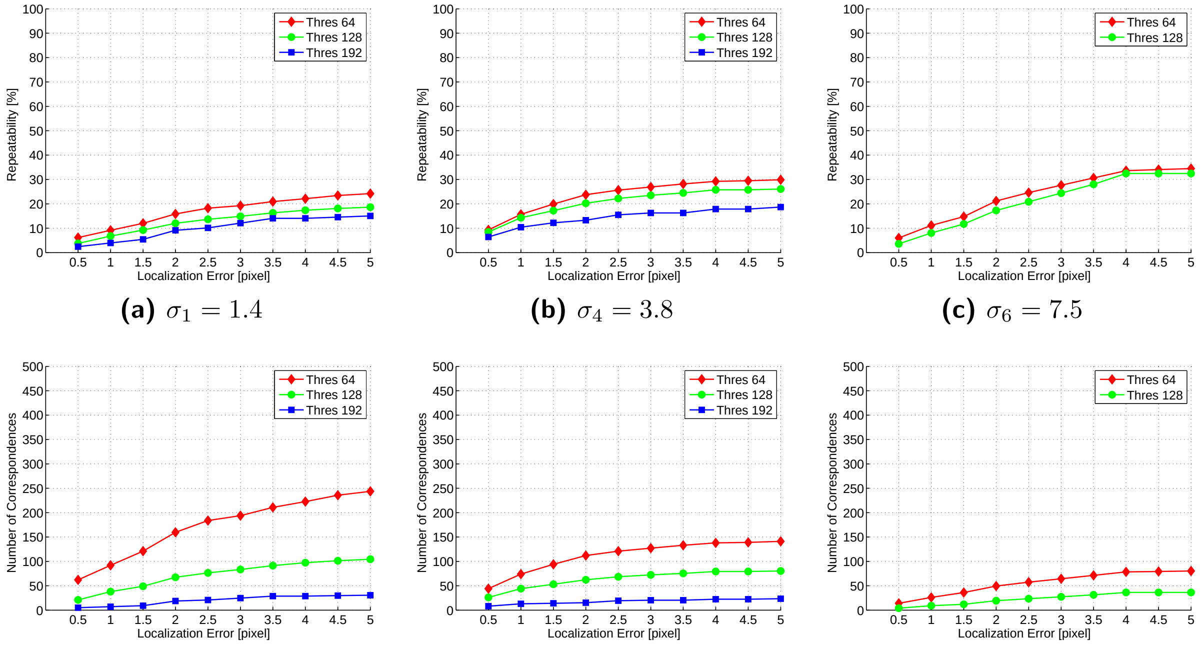
MSER Detector Figure 3.16 presents the repeatability score for different parameter combinations of the MSER detector. At first glance it becomes clear that this detector is heavily affected by the seasonal variations. The repeatability rates for regions with a low overlap error (i.e. $\leq 20\%$) decrease by 20% compared to the results for the DAL1 set and the number of correspondences drops to at most 50. Tables 3.12 to 3.14 show the distances of the regions' centers of gravity to the ground truth epipolar geometry. For the DAL2 set regions with an overlap error as low as 20%, the centers of gravity deviate more than 2 pixels from the corresponding epipolar lines while the distances for the DAL1 data set (which basically captures the same image scene) deviate less than 1 pixel for overlap errors up to 30%.



(a) $\sigma_1 = 1.4$ **(b)** $\sigma_4 = 3.8$ **(c)** $\sigma_6 = 7.5$

(d) $\sigma_1 = 1.4$ **(e)** $\sigma_4 = 3.8$ **(f)** $\sigma_6 = 7.5$

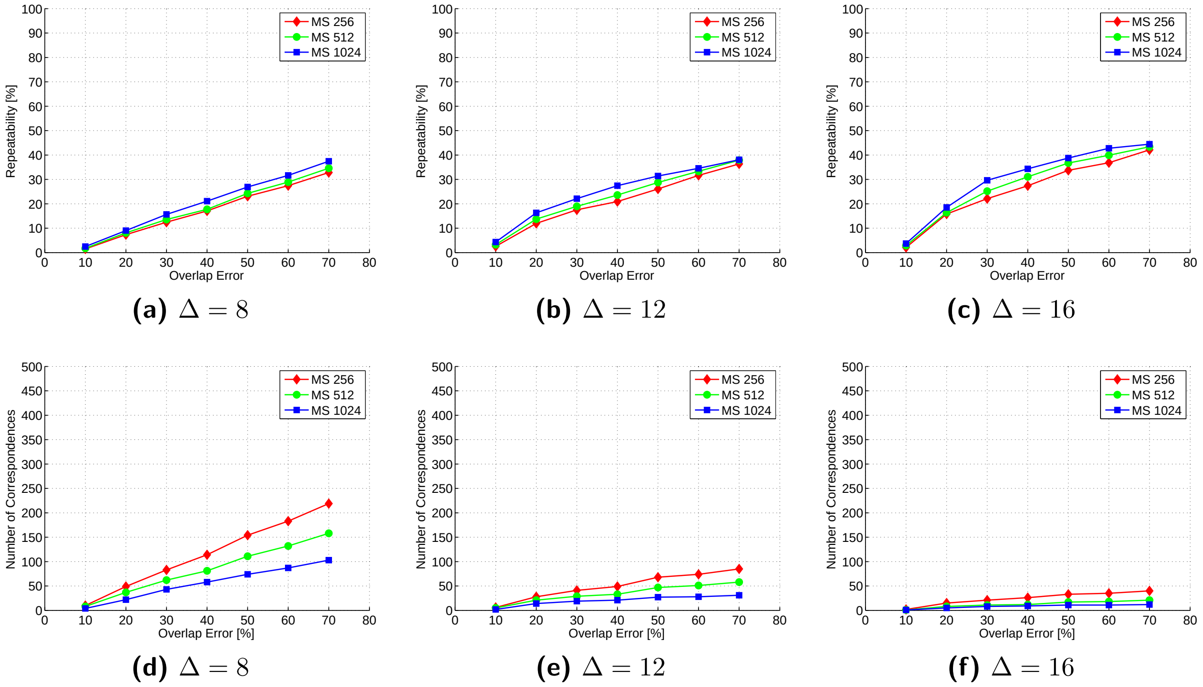
Figure 3.14: Harris repeatability rates for `DAL2`.



(a) $\sigma_1 = 1.4$ **(b)** $\sigma_4 = 3.8$ **(c)** $\sigma_6 = 7.5$

(d) $\sigma_1 = 1.4$ **(e)** $\sigma_4 = 3.8$ **(f)** $\sigma_6 = 7.5$

Figure 3.15: Hessian repeatability rates for `DAL2`.



(a) $\Delta = 8$ **(b)** $\Delta = 12$ **(c)** $\Delta = 16$

(d) $\Delta = 8$ **(e)** $\Delta = 12$ **(f)** $\Delta = 16$

Figure 3.16: MSER repeatability rates for `DAL2`.

Minimum Margin	Minimum Size 256			Minimum Size 512			Minimum Size 1024		
	OE 10	OE 20	OE 30	OE 10	OE 20	OE 30	OE 10	OE 20	OE 30
$\Delta = 8$	0.39	0.65	0.82	0.53	0.83	1.02	0.66	0.98	1.26
$\Delta = 12$	0.34	0.49	0.61	0.43	0.61	0.68	0.66	0.74	0.80
$\Delta = 16$	0.27	0.47	0.58	0.40	0.65	0.69	0.68	0.86	0.81

Table 3.12: Distances of MSER centers of gravity to the ground truth for `DEN1`.

Minimum Margin	Minimum Size 256			Minimum Size 512			Minimum Size 1024		
	OE 10	OE 20	OE 30	OE 10	OE 20	OE 30	OE 10	OE 20	OE 30
$\Delta = 8$	0.60	0.78	0.93	0.60	0.83	0.94	0.65	0.88	0.89
$\Delta = 12$	0.65	0.75	0.81	0.65	0.79	0.83	0.71	0.82	0.83
$\Delta = 16$	0.64	0.66	0.70	0.62	0.67	0.67	0.72	0.74	0.74

Table 3.13: Distances of MSER centers of gravity to the ground truth for `DAL1`.

Minimum Margin	Minimum Size 256			Minimum Size 512			Minimum Size 1024		
	OE 10	OE 20	OE 30	OE 10	OE 20	OE 30	OE 10	OE 20	OE 30
$\Delta = 8$	0.84	2.17	2.90	0.87	2.62	3.44	0.87	3.51	4.16
$\Delta = 12$	0.47	2.23	2.78	0.34	2.67	3.52	0.46	3.66	4.68
$\Delta = 16$	0.62	0.84	2.41	0.71	0.88	3.85	0.14	0.78	2.00

Table 3.14: Distances of MSER centers of gravity to the ground truth for `DAL2`.

3.5.3 TV-L^1 Pre-processing

It was shown in Section 3.5.1 that for both Harris and Hessian many detections evolve from fine image structures. However, these detections are hardly repeatable in other images and they can be considered as irrelevant. Hence, TV-L^1 pre-processing is used to decompose the input image into an image that contains fine details and another image that contains the structural part of the input image (cf. Section 2.3). The latter image is then used for feature detection.

This approach can be used to improve the repeatability rates. Tables 3.15 to 3.17 display the repeatability rates for localization errors up to 1.5 pixels for both the original images and TV-L^1 pre-processed images where the parameter λ was set to 0.7 and 0.5 respectively. Note that with a decreasing value for λ more and more structures move from the structural image to the image that contains the details. For detection scales σ_1 and σ_4 the *relative* improvement of the repeatability rates is up to 50% for $\lambda = 0.5$. With a further increasing detection scale, the impact of the pre-processing is reduced. Also note that pre-processing decreases total number of correspondences significantly.

Detection	**TV-L^1**	**Repeatability**			**Correspondences**		
Scale	λ	≤ 0.5	≤ 1.0	≤ 1.5	≤ 0.5	≤ 1.0	≤ 1.5
$\sigma_x = 1.40$	N/A	9.03	13.27	16.86	409	601	764
$\sigma_x = 3.84$	N/A	8.05	13.69	16.99	207	352	437
$\sigma_x = 7.53$	N/A	7.62	13.26	18.05	54	94	128
$\sigma_x = 1.40$	0.7	11.71	16.63	21.03	255	362	458
$\sigma_x = 3.84$	0.7	9.02	15.55	19.46	203	350	438
$\sigma_x = 7.53$	0.7	7.02	12.89	17.77	49	90	124
$\sigma_x = 1.40$	0.5	13.49	19.97	24.11	188	278	336
$\sigma_x = 3.84$	0.5	10.07	16.08	20.08	184	294	367
$\sigma_x = 7.53$	0.5	8.14	14.18	18.40	54	94	122

Table 3.15: TV-L^1 pre-processed Harris repeatability for DEN1.

Detection	**TV-L^1**	**Repeatability**			**Correspondences**		
Scale	λ	≤ 0.5	≤ 1.0	≤ 1.5	≤ 0.5	≤ 1.0	≤ 1.5
$\sigma_x = 1.40$	N/A	6.67	10.19	12.69	118	180	225
$\sigma_x = 3.84$	N/A	5.50	9.63	14.68	72	126	192
$\sigma_x = 7.53$	N/A	5.57	10.98	15.41	34	67	94
$\sigma_x = 1.40$	0.7	7.59	12.25	15.55	66	107	135
$\sigma_x = 3.84$	0.7	5.72	10.06	15.62	66	116	180
$\sigma_x = 7.53$	0.7	6.45	11.13	15.13	40	69	94
$\sigma_x = 1.40$	0.5	9.59	15.24	19.55	55	87	112
$\sigma_x = 3.84$	0.5	8.08	13.19	18.18	79	129	178
$\sigma_x = 7.53$	0.5	6.31	11.09	16.04	37	65	94

Table 3.16: TV-L^1 pre-processed Harris repeatability for DAL1.

Detection	**TV-L^1**	**Repeatability**			**Correspondences**		
Scale	λ	≤ 0.5	≤ 1.0	≤ 1.5	≤ 0.5	≤ 1.0	≤ 1.5
$\sigma_x = 1.40$	N/A	6.50	10.85	13.44	118	197	244
$\sigma_x = 3.84$	N/A	4.01	7.04	10.10	37	65	93
$\sigma_x = 7.53$	N/A	3.15	6.10	8.66	16	31	44
$\sigma_x = 1.40$	0.7	7.00	11.71	15.43	63	105	139
$\sigma_x = 3.84$	0.7	4.12	7.95	10.57	32	62	82
$\sigma_x = 7.53$	0.7	3.02	6.25	7.78	15	31	39
$\sigma_x = 1.40$	0.5	9.92	18.12	26.14	39	71	103
$\sigma_x = 3.84$	0.5	4.49	7.49	12.48	30	50	83
$\sigma_x = 7.53$	0.5	2.47	5.15	6.79	12	25	33

Table 3.17: TV-L^1 pre-processed Harris repeatability for DAL2.

3.5.4 Discussion

The goal of this section was to find the best detector for each image set. Section 3.5.2 showed that parameter settings which yield a higher number of correspondences usually tend to have lower repeatability scores. Particularly the number of highly precise correspondences (e.g. with a localization error less than a pixel or with an region overlap error less than 20%) is usually higher for these settings. In this context, Section 3.5.3 discussed the impact of TV-L^1 pre-processing. Tables 3.15 to 3.17 show that it is possible to improve the repeatability rate while preserving a considerable amount of correspondences. Altogether, it is necessary to find a reasonable trade-off between the requirements regarding repeatability rates and absolute number of correspondences.

For the (short-term) temporal variations in the DEN1 and DAL1 image sets, the MSER detector accomplishes the best performances. For the images in these sets, the repeated regions have an adequate accuracy which means that the distances of their centers of gravity to the ground truth epipolar lines are less than a pixel (cf. Tables 3.12 and 3.13 respectively). However, compared to, e.g., the Harris detector the total number of correspondences is considerably low. Nevertheless, for these types of temporal variations, the MSER detector is the best choice.

This rating changes for the seasonal variations in the DAL2 set. Although MSERs have the best repeatability rates for DAL2 it turns out that the centers of gravity of corresponding regions are not adequately repeatable (see Table 3.14). The main reason for this is the massive appearance change: Contrary to the DAL1 set which captures significant appearance changes too, the temporal changes in the DAL2 set change the shape of the detected MSERs (while the shadows in the DAL1 set occlude entire regions) and this causes a negative impact on the repeatability of the centers of gravity.

In contrast to MSERs, the Harris corner detector proved to achieve reasonable results for *all* temporal variations in the image sets. Hence, this detector is not only the best choice for the DAL2 set, it is also proposed to use the Harris detector for the matching algorithm that is developed in Chapter 4.

Note that it has been shown that the repeatability scores obtained with Harris and Hessian are comparable, however, Harris corners generally tend to have better repeatability scores while simultaneously finding a lot more precise correspondences. Thus, the Harris detector is preferred to the Hessian detector.

3.6 Descriptor Evaluation

This section presents the evaluation of SIFT [24; 25], SURF [4; 3], and Steerable Filters [13] feature descriptors. SIFT is computed with the VLFeat library [60], SURF descriptors are computed with the original binary [3], and Steerable Filter descriptors are based on a MATLAB implementation following the results from Winder et al. [63; 64]. Since the image sets are already ordered, all descriptors are computed in a rotationally variant manner. Following the conclusions in Section 3.5.4 the descriptors are evaluated with the proposed detectors (i.e. Harris and MSER).

3.6.1 General Description Trends

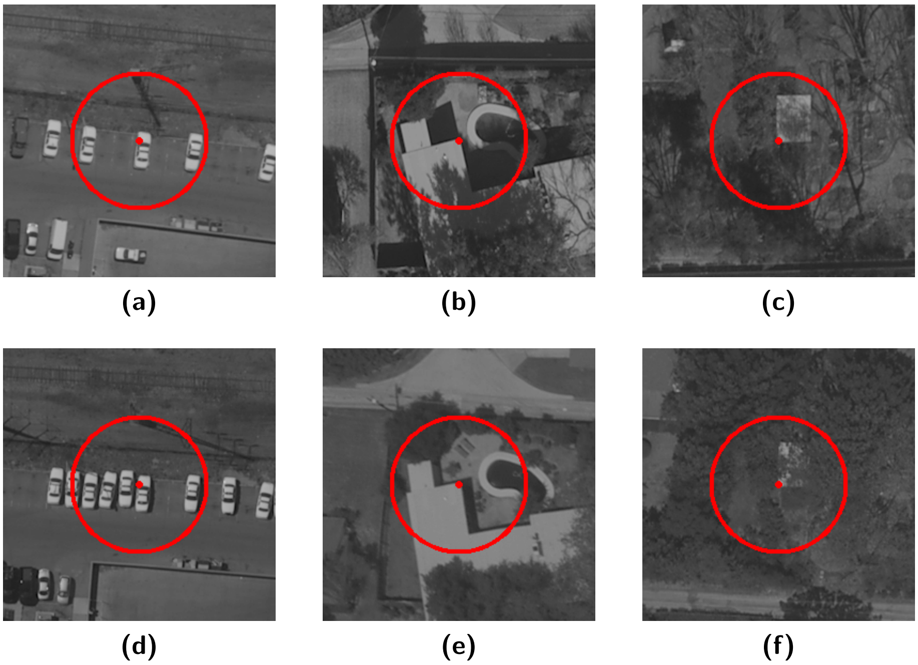
The aim of this section is to illustrate the impact of temporal variations to feature descriptors and to analyze the basic trends for different descriptor settings. The evaluation puts the focus on two points: (a) To find the best footprint patch size for each descriptor and (b) to evaluate the basic trends for feature matching with different settings.

Figure 3.17 visualizes temporal variations in descriptor patches. In each example, the temporal changes have a negative impact on the descriptor performance and none of these samples are successfully identified as nearest neighbors.

Tables 3.18 to 3.20 quantify this observation by measuring the impact of temporal changes on the Euclidean descriptor distances between correspondences. These tables list the minimum, maximum, and mean (along with the standard deviation) descriptor distances for correspondences with a localization error less then 0.5 pixels. The descriptor distances are computed for Steerable Filter descriptor vectors with footprint patches of size 41×41, 63×63, and 127×127 pixels. For the sake of completeness, all descriptor distances are additionally computed for in-strip image pairs.

The impact of temporal variations is measured by comparing the distances for cross-strip correspondences. For example, the mean distance for a descriptor footprint patch size of 127×127 pixels is 785.235 for `DEN1`, 836.384 for `DAL1`, and 989.154 for `DAL2`. Compared to `DEN1`, the mean distances increase by 6% for `DAL1` and by 26% for `DAL2`. The same is true for the minimum distance where there is a relative increase by 260% between `DEN1` (198.464) and `DAL2` (512.866). Note that for each data set, the mean descriptor distances decrease for increasing footprint patch sizes.

Summarizing the results from tables 3.18 to 3.20 it becomes clear that temporal changes have a negative impact on the distinctiveness of the descriptors. It was shown that particularly seasonal changes pose a problem for the task of descriptor matching.



(a) (b) (c)

(d) (e) (f)

Figure 3.17: Descriptor patches containing temporal variations.

Descriptor Distance	41 × 41 in-strip	41 × 41 cross-strip	63 × 63 in-strip	63 × 63 cross-strip	127 × 127 in-strip	127 × 127 cross-strip
minimum	69.886	209.652	58.378	164.548	75.406	198.464
maximum	2004.439	2151.467	1609.216	1785.496	1067.445	1454.729
mean	484.842	945.545	396.815	884.047	301.729	785.235
std	231.731	364.241	197.765	320.613	165.141	264.491

Table 3.18: Steerable Filter descriptor distances for correspondences in `DEN1`.

Descriptor Distance	41 × 41 in-strip	41 × 41 cross-strip	63 × 63 in-strip	63 × 63 cross-strip	127 × 127 in-strip	127 × 127 cross-strip
minimum	106.639	234.448	107.981	265.028	131.719	238.394
maximum	896.252	2215.513	848.129	1596.414	646.407	1267.678
mean	469.268	1041.247	384.769	972.088	273.726	836.384
std	174.552	365.591	150.821	270.807	99.791	214.541

Table 3.19: Steerable Filter descriptor distances for correspondences in `DAL1`.

Descriptor Distance	41 × 41 in-strip	41 × 41 cross-strip	63 × 63 in-strip	63 × 63 cross-strip	127 × 127 in-strip	127 × 127 cross-strip
minimum	155.377	291.279	166.655	453.078	109.425	512.866
maximum	1419.253	1968.425	1087.488	1924.589	816.661	1436.999
mean	515.571	1133.113	430.757	1097.026	323.756	989.154
std	226.997	347.453	189.516	278.993	131.046	196.112

Table 3.20: Steerable Filter descriptor distances for correspondences in `DAL2`.

Previously, the correspondences computed in Section 3.5.2 was used to analyze the impact of temporal variations on the Euclidean distances between corresponding descriptor vectors. Now, this information is used to measure the number of correspondences which are successfully identified as nearest neighbors (NN) for any given detector/descriptor combination. In order to achieve this, each reference feature in one image is first matched with all detected features in the adjacent image and then the nearest neighbor of the reference feature is computed and compared with the corresponding feature. The results are displayed in tables 3.21 to 3.32.

Harris Detector Each descriptor vector is computed for footprint patches of size 41 × 41, 63 × 63, and 127 × 127 pixels. For each combination of detection scale and detector response threshold, the best recall scores are achieved with a descriptor footprint patch size of 127 × 127 pixels and the recall rates of all descriptors are within 95% of the top results. In this evaluation SURF achieves the best scores for `DEN1` and `DAL1` whereas Steerable filters perform best for the `DAL2` image set. As an illustration, Tables 3.21 to 3.23 show the recall rates along with the number of matched correspondences (values in brackets) for the Harris detector combined with the SURF descriptor. For the `DEN1` set about 75% to 90% of the correspondences are correctly found as nearest neighbors. For `DAL1` recall ranges from 65% for detection scale σ_1 to 85% for σ_6 and for `DAL2` recall varies between 50% and 75% respectively. Especially for `DAL2` the performance considerably decreased, compared to the other data sets. There are two main reasons for this. First, the image scene has more repetitive motifs than the scene captured by the `DEN1` set. Second, the seasonal appearance changes in the `DAL2` set have a strong effect to the descriptor performance.

MSER Detector For MSERs the best recall scores are achieved with a measurement region that is three times larger than the extremal region (i.e. the ellipse scale is 3). For this detector the best recall rates are computed with Steerable Filters. Tables 3.27 to 3.29 list the computed recall scores together with the number of matched correspondences (values in brackets). For `DEN1` and `DAL1` recall ranges from 75% to 90% and the results are similar than for the Harris and Hessian detectors. However, for the `DAL2` set recall rates drop to 30% to 60%. This is consistent with the observation from Section 3.5.2, where it was shown that the MSER repeatability is heavily affected by this kind of changes.

Generally, the low recall rates reported in the tables pose a problem for the matching procedure. The descriptor vectors are not sufficiently distinctive, thus, actually corresponding descriptors can not be successfully matched. In order to improve the recall scores the matching task needs to be enhanced with a geometric constraint. Basically, the same technique was already used to compute the repeatability measure (cf. Section 3.3.1) where an approximate estimation of the minimum and maximum projection height was used to restrict the search area along the ground truth epipolar line. This time, the projection is used to restrict the area where nearest neighbors are searched. Note that in this context the estimated values for the minimum and maximum projection height can be less precise than for the repeatability measure. Even for search areas with a size of 512 × 512 pixels, the recall increases for the `DAL2` set to at least 85% for Harris and 50% for MSERs. Tables 3.24 to 3.26 list the recall rates along with the number of matched correspondences (values in brackets) for the Harris detector combined with the SURF descriptor when this geometric constraint is used. Confer tables 3.30 to 3.32 for the improved results of the MSER detector in combination with Steerable Filters.

Detection Scale	Threshold	41 × 41	63 × 63	127 × 127
$\sigma_1 = 1.40$	1024	65.28% (267)	72.62% (297)	78.97% (323)
$\sigma_1 = 1.40$	2048	65.35% (166)	72.05% (183)	80.71% (205)
$\sigma_1 = 1.40$	4096	68.55% (109)	72.96% (116)	80.50% (128)
$\sigma_4 = 3.84$	1024	65.22% (135)	72.46% (150)	83.09% (172)
$\sigma_4 = 3.84$	2048	62.67% (94)	73.33% (110)	81.33% (122)
$\sigma_4 = 3.84$	4096	68.00% (51)	82.67% (62)	82.67% (62)
$\sigma_6 = 7.53$	1024	79.63% (43)	85.19% (46)	92.59% (50)
$\sigma_6 = 7.53$	2048	72.73% (16)	77.27% (17)	81.82% (18)
$\sigma_6 = 7.53$	4096	66.67% (8)	66.67% (8)	75.00% (9)

Table 3.21: Harris correspondences of the DEN1 set found as NN with SURF.

Detection Scale	Threshold	41 × 41	63 × 63	127 × 127
$\sigma_1 = 1.40$	1024	43.22% (51)	56.78% (67)	67.80% (80)
$\sigma_1 = 1.40$	2048	45.71% (32)	60.00% (42)	68.57% (48)
$\sigma_1 = 1.40$	4096	52.17% (12)	65.22% (15)	78.26% (18)
$\sigma_4 = 3.84$	1024	48.61% (35)	63.89% (46)	76.39% (55)
$\sigma_4 = 3.84$	2048	55.77% (29)	71.15% (37)	84.62% (44)
$\sigma_4 = 3.84$	4096	63.33% (19)	76.67% (23)	90.00% (27)
$\sigma_6 = 7.53$	1024	55.88% (19)	58.82% (20)	85.29% (29)
$\sigma_6 = 7.53$	2048	53.57% (15)	53.57% (15)	82.14% (23)
$\sigma_6 = 7.53$	4096	70.00% (7)	80.00% (8)	100.00% (10)

Table 3.22: Harris correspondences of the DAL1 set found as NN with SURF.

Detection Scale	Threshold	41 × 41	63 × 63	127 × 127
$\sigma_1 = 1.40$	1024	39.83% (47)	52.54% (62)	51.69% (61)
$\sigma_1 = 1.40$	2048	45.90% (28)	59.02% (36)	59.02% (36)
$\sigma_1 = 1.40$	4096	57.14% (12)	61.90% (13)	61.90% (13)
$\sigma_4 = 3.84$	1024	48.65% (18)	48.65% (18)	62.16% (23)
$\sigma_4 = 3.84$	2048	50.00% (11)	63.64% (14)	81.82% (18)
$\sigma_4 = 3.84$	4096	36.36% (4)	45.45% (5)	90.91% (10)
$\sigma_6 = 7.53$	1024	37.50% (6)	37.50% (6)	62.50% (10)
$\sigma_6 = 7.53$	2048	37.50% (3)	62.50% (5)	75.00% (6)
$\sigma_6 = 7.53$	4096	50.00% (2)	75.00% (3)	75.00% (3)

Table 3.23: Harris correspondences of the DAL2 set found as NN with SURF.

Detection Scale	Threshold	41×41	63×63	127×127
$\sigma_1 = 1.40$	1024	87.78% (359)	89.24% (365)	91.44% (374)
$\sigma_1 = 1.40$	2048	90.55% (230)	90.16% (229)	91.73% (233)
$\sigma_1 = 1.40$	4096	92.45% (147)	90.57% (144)	91.19% (145)
$\sigma_4 = 3.84$	1024	92.27% (191)	93.24% (193)	95.65% (198)
$\sigma_4 = 3.84$	2048	92.00% (138)	94.00% (141)	96.00% (144)
$\sigma_4 = 3.84$	4096	94.67% (71)	98.67% (74)	100.00% (75)
$\sigma_6 = 7.53$	1024	100.00% (54)	98.15% (53)	100.00% (54)
$\sigma_6 = 7.53$	2048	100.00% (22)	100.00% (22)	100.00% (22)
$\sigma_6 = 7.53$	4096	100.00% (12)	100.00% (12)	100.00% (12)

Table 3.24: Harris correspondences of the `DEN1` set found as “projective” NN with SURF.

Detection Scale	Threshold	41×41	63×63	127×127
$\sigma_1 = 1.40$	1024	83.05% (98)	83.05% (98)	92.37% (109)
$\sigma_1 = 1.40$	2048	84.29% (59)	85.71% (60)	92.86% (65)
$\sigma_1 = 1.40$	4096	82.61% (19)	91.30% (21)	95.65% (22)
$\sigma_4 = 3.84$	1024	81.94% (59)	93.06% (67)	97.22% (70)
$\sigma_4 = 3.84$	2048	80.77% (42)	94.23% (49)	98.08% (51)
$\sigma_4 = 3.84$	4096	86.67% (26)	96.67% (29)	100.00% (30)
$\sigma_6 = 7.53$	1024	88.24% (30)	85.29% (29)	97.06% (33)
$\sigma_6 = 7.53$	2048	85.71% (24)	82.14% (23)	96.43% (27)
$\sigma_6 = 7.53$	4096	100.00% (10)	100.00% (10)	100.00% (10)

Table 3.25: Harris correspondences of the `DAL1` set found as “projective” NN with SURF.

Detection Scale	Threshold	41×41	63×63	127×127
$\sigma_1 = 1.40$	1024	86.44% (102)	90.68% (107)	87.29% (103)
$\sigma_1 = 1.40$	2048	83.61% (51)	91.80% (56)	91.80% (56)
$\sigma_1 = 1.40$	4096	90.48% (19)	100.00% (21)	100.00% (21)
$\sigma_4 = 3.84$	1024	83.78% (31)	81.08% (30)	91.89% (34)
$\sigma_4 = 3.84$	2048	90.91% (20)	81.82% (18)	100.00% (22)
$\sigma_4 = 3.84$	4096	81.82% (9)	72.73% (8)	100.00% (11)
$\sigma_6 = 7.53$	1024	93.75% (15)	93.75% (15)	87.50% (14)
$\sigma_6 = 7.53$	2048	87.50% (7)	87.50% (7)	87.50% (7)
$\sigma_6 = 7.53$	4096	100.00% (4)	100.00% (4)	100.00% (4)

Table 3.26: Harris correspondences of the `DAL2` set found as “projective” NN with SURF.

MM	MS	ES 1	ES 2	ES 3
$\Delta = 8$	256	69.75% (219)	68.47% (215)	73.89% (232)
$\Delta = 8$	512	72.68% (133)	72.13% (132)	78.69% (144)
$\Delta = 8$	1024	80.41% (78)	83.51% (81)	87.63% (85)
$\Delta = 12$	256	70.81% (131)	65.41% (121)	69.19% (128)
$\Delta = 12$	512	79.41% (81)	70.59% (72)	76.47% (78)
$\Delta = 12$	1024	85.71% (42)	85.71% (42)	81.63% (40)
$\Delta = 16$	256	72.81% (83)	66.67% (76)	69.30% (79)
$\Delta = 16$	512	83.02% (44)	73.58% (39)	77.36% (41)
$\Delta = 16$	1024	81.82% (18)	90.91% (20)	86.36% (19)

Table 3.27: MSER correspondences of the `DEN1` set found as NN with Steerable Filters.

MM	MS	ES 1	ES 2	ES 3
$\Delta = 8$	256	61.47% (67)	70.64% (77)	74.31% (81)
$\Delta = 8$	512	63.16% (60)	73.68% (70)	80.00% (76)
$\Delta = 8$	1024	68.57% (48)	75.71% (53)	85.71% (60)
$\Delta = 12$	256	66.15% (43)	75.38% (49)	80.00% (52)
$\Delta = 12$	512	75.00% (42)	78.57% (44)	85.71% (48)
$\Delta = 12$	1024	80.49% (33)	80.49% (33)	92.68% (38)
$\Delta = 16$	256	65.85% (27)	80.49% (33)	80.49% (33)
$\Delta = 16$	512	74.29% (26)	82.86% (29)	85.71% (30)
$\Delta = 16$	1024	88.00% (22)	92.00% (23)	96.00% (24)

Table 3.28: MSER correspondences of the `DAL1` set found as NN with Steerable Filters.

MM	MS	ES 1	ES 2	ES 3
$\Delta = 8$	256	32.65% (16)	38.78% (19)	34.69% (17)
$\Delta = 8$	512	29.73% (11)	43.24% (16)	29.73% (11)
$\Delta = 8$	1024	27.27% (6)	45.45% (10)	40.91% (9)
$\Delta = 12$	256	39.29% (11)	46.43% (13)	35.71% (10)
$\Delta = 12$	512	38.10% (8)	57.14% (12)	38.10% (8)
$\Delta = 12$	1024	28.57% (4)	57.14% (8)	42.86% (6)
$\Delta = 16$	256	60.00% (9)	33.33% (5)	46.67% (7)
$\Delta = 16$	512	50.00% (4)	62.50% (5)	62.50% (5)
$\Delta = 16$	1024	40.00% (2)	60.00% (3)	60.00% (3)

Table 3.29: MSER correspondences of the `DAL2` set found as NN with Steerable Filters.

MM	MS	ES 1	ES 2	ES 3
$\Delta = 8$	256	81.85% (257)	84.71% (266)	86.31% (271)
$\Delta = 8$	512	84.15% (154)	85.25% (156)	87.98% (161)
$\Delta = 8$	1024	88.66% (86)	89.69% (87)	89.69% (87)
$\Delta = 12$	256	85.41% (158)	85.95% (159)	84.86% (157)
$\Delta = 12$	512	89.22% (91)	88.24% (90)	89.22% (91)
$\Delta = 12$	1024	89.80% (44)	87.76% (43)	85.71% (42)
$\Delta = 16$	256	86.84% (99)	87.72% (100)	86.84% (99)
$\Delta = 16$	512	92.45% (49)	92.45% (49)	92.45% (49)
$\Delta = 16$	1024	86.36% (19)	90.91% (20)	90.91% (20)

Table 3.30: MSER correspondences of the DEN1 set found as "projective" NN with Steerable Filters.

MM	MS	ES 1	ES 2	ES 3
$\Delta = 8$	256	78.90% (86)	87.16% (95)	88.07% (96)
$\Delta = 8$	512	80.00% (76)	88.42% (84)	86.32% (82)
$\Delta = 8$	1024	85.71% (60)	90.00% (63)	88.57% (62)
$\Delta = 12$	256	86.15% (56)	90.77% (59)	92.31% (60)
$\Delta = 12$	512	87.50% (49)	92.86% (52)	91.07% (51)
$\Delta = 12$	1024	90.24% (37)	92.68% (38)	92.68% (38)
$\Delta = 16$	256	85.37% (35)	90.24% (37)	90.24% (37)
$\Delta = 16$	512	85.71% (30)	91.43% (32)	88.57% (31)
$\Delta = 16$	1024	92.00% (23)	96.00% (24)	96.00% (24)

Table 3.31: MSER correspondences of the DAL1 set found as "projective" NN with Steerable Filters.

MM	MS	ES 1	ES 2	ES 3
$\Delta = 8$	256	59.18% (29)	65.31% (32)	55.10% (27)
$\Delta = 8$	512	54.05% (20)	59.46% (22)	48.65% (18)
$\Delta = 8$	1024	54.55% (12)	54.55% (12)	59.09% (13)
$\Delta = 12$	256	64.29% (18)	75.00% (21)	60.71% (17)
$\Delta = 12$	512	61.90% (13)	71.43% (15)	57.14% (12)
$\Delta = 12$	1024	57.14% (8)	64.29% (9)	64.29% (9)
$\Delta = 16$	256	66.67% (10)	73.33% (11)	66.67% (10)
$\Delta = 16$	512	62.50% (5)	62.50% (5)	62.50% (5)
$\Delta = 16$	1024	60.00% (3)	60.00% (3)	60.00% (3)

Table 3.32: MSER correspondences of the DAL2 set found as "projective" NN with Steerable Filters.

3.6.2 Recall vs. 1-Precision

The objective of this section is to find the most suitable detector/descriptor combination for each image set. The performances of the descriptors are measured in terms of recall vs. 1-precision. Recall is the number of correct matches with respect to the number of correspondences and 1-precision is the number of false matches with respect to the number of putative matches. Two features are matched if the Euclidean distance between the reference descriptor vector and its nearest neighbor is below an arbitrary threshold value. In order to generate a curve this threshold is varied.

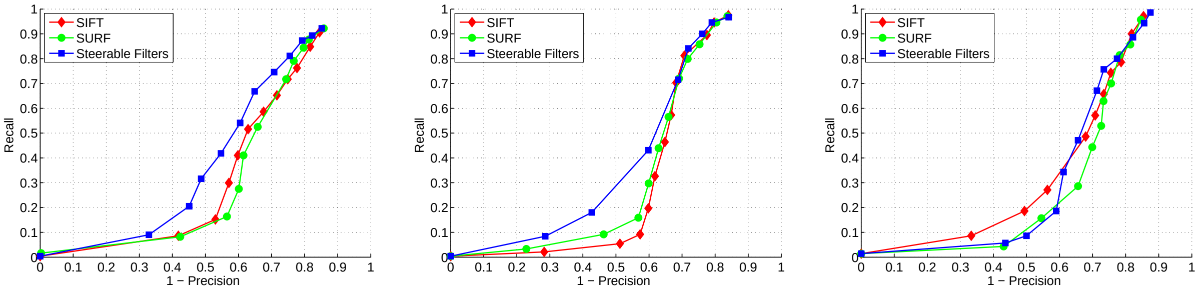
For all evaluations, the search area for nearest neighbors is restricted by the geometric constraint introduced in the previous section. The descriptor footprint patches are fixed to a size of 127×127 pixels and TV-L^1 pre-processing is used (the parameter λ is fixed to 0.5) for the Harris detector. For MSERs the footprint patches are three times larger than the detected regions.

3.6.2.1 The DEN1 Image Set

The temporal variations in this data set come rather from moving objects than from illumination or shadow changes. Figures 3.18 and 3.19 show the recall vs. 1-precision graphs for the Harris and MSER detectors respectively. The performances of SIFT and SURF are similar and Steerable Filters achieve the best scores for both Harris corners and MSERs.

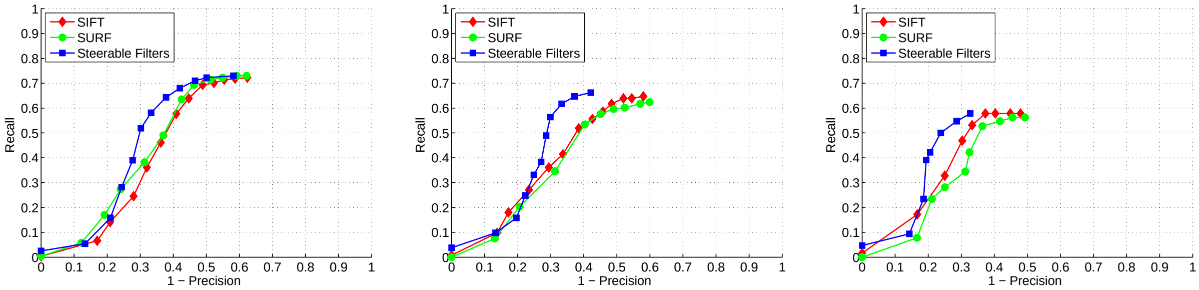
The recall rate ranges from around 10% for a 1-precision value of 0.3 pixel to 98% for a 1-precision score of 0.85 and the performance slightly decreases for an increasing detection scale σ. Compared to the results in previous descriptor evaluations [30; 33], it is evident that the obtained recall vs. 1-precision graphs significantly differ from the reported results. This can be explained by the different complexities of the image sets.

Note the superior performance of descriptors computed for MSERs. Even though the maximum recall rate is limited to 73% these rates are obtained for considerably lower 1-precision scores. For example, for a precision score of 0.5 the recall is 70% (i.e. 220 correct matches) for MSERs compared the recall of 30% (i.e. 60 correct matches) for Harris.



(a) $\sigma_1 = 1.4,\ \lambda = 0.5$ **(b)** $\sigma_4 = 3.8,\ \lambda = 0.5$ **(c)** $\sigma_6 = 7.5,\ \lambda = 0.5$

Figure 3.18: Harris recall vs. 1-precision rates for DEN1.



(a) MS 256 **(b)** MS 512 **(c)** MS 1024

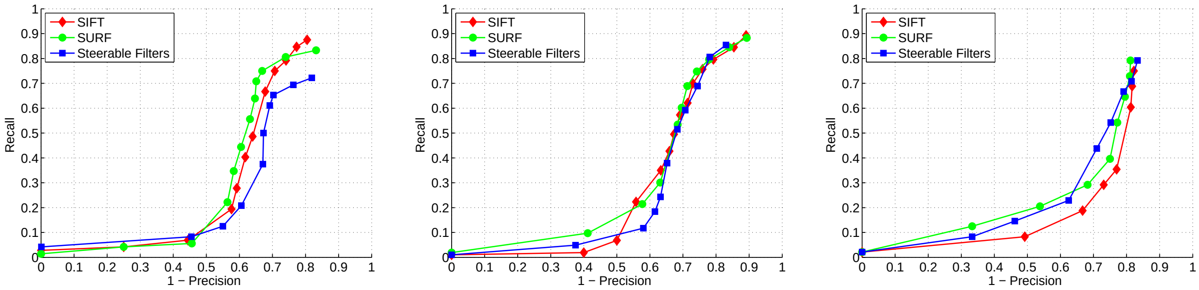
Figure 3.19: MSER recall vs. 1-precision rates for DEN1.

3.6.2.2 The DAL1 Image Set

This data set is dominated by significant appearance changes caused by shadowed regions. Large parts that are clearly visible in one image are covered with shadow in the other image. The descriptor performances for this data set are listed in Figure 3.20 for the Harris detector and Figure 3.21 for MSERs respectively. The recall scores range from 2% to 90% for Harris and up to 65% for MSERs. Again, the maximum recall rate for MSERs is lower than for Harris, however, these rates are obtained for considerably lower 1-precision scores.

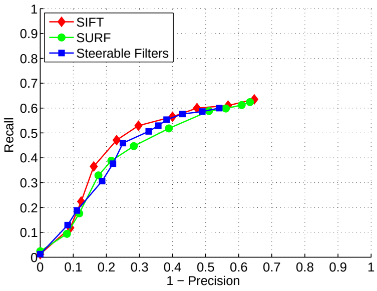
In contrast to the DEN1 set, where Steerable Filters clearly achieved the best results there is no big difference here. The individual recall vs. 1-precision graphs are rather similar to each other. For example, starting with a 1-precision score of 0.5 the recall rate increases drastically for all curves in Figure 3.20. However, at a second glance it becomes clear that particularly for the curve in Figure 3.20a SURF outperforms the other descriptors. For example, the recall rate for a 1-precision score of 0.6 is 20% for Steerable Filters, 30% for SIFT, and 42% for SURF descriptors.

Compared to the results for the DEN1 image set, the performances are generally lower. This can be explained with the different image scenes. The DAL1 set contains more repetitive structures (e.g. swimming pools) which have a higher chance to be mismatched.



(a) $\sigma_1 = 1.4, \lambda = 0.5$ **(b)** $\sigma_4 = 3.8, \lambda = 0.5$ **(c)** $\sigma_6 = 7.5, \lambda = 0.5$

Figure 3.20: Harris recall vs. 1-precision rates for DAL1.



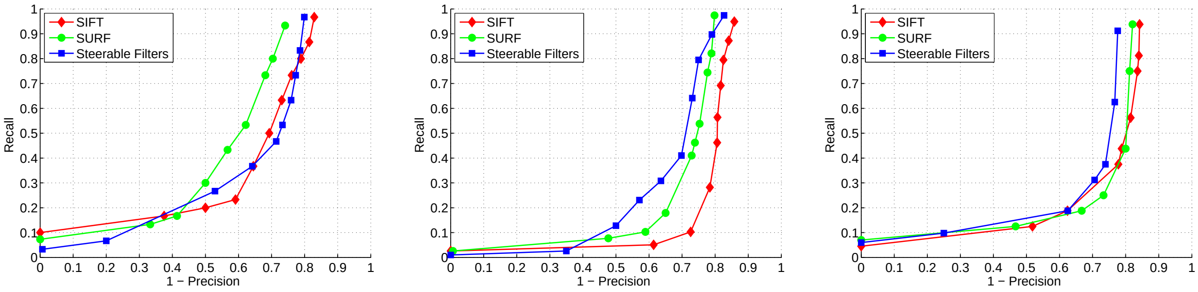
(a) MS 256 **(b)** MS 512 **(c)** MS 1024

Figure 3.21: MSER recall vs. 1-precision rates for DAL1

3.6.2.3 The DAL2 Image Set

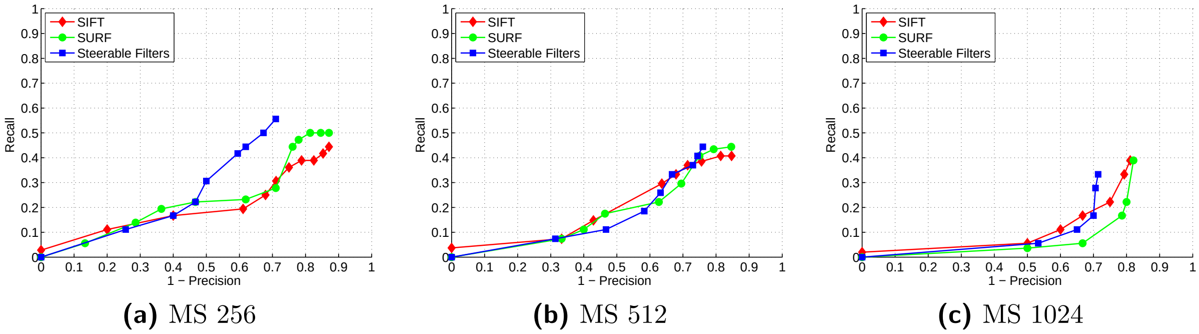
This data set captures temporal variations of about 5 months. Hence, it is characterized by massive appearance changes due to seasonal vegetation and shadows caused by the low position of the sun. Section 3.5.2 already concluded that the MSER detector is not applicable to this image set. However, for the sake of completeness recall vs. 1-precision scores are presented in Figure 3.23. The displayed graphs clearly validate these observations particularly for the increasing minimum region size in Figures 3.23b and 3.23c.

For the recall vs. 1-precision graphs displayed in Figure 3.22 there is generally a huge difference for the performances of the evaluated descriptors. It is remarkable that the performance of SURF descriptors is far beyond the performance of SIFT. For example, the recall rate for a 1-precision score of 0.55 is 21% for SIFT and 43% for SURF (cf. Figure 3.22a).



(a) $\sigma_1 = 1.4$, $\lambda = 0.5$ **(b)** $\sigma_4 = 3.8$, $\lambda = 0.5$ **(c)** $\sigma_6 = 7.5$, $\lambda = 0.5$

Figure 3.22: Harris recall vs. 1-precision rates for DAL2.



(a) MS 256 **(b)** MS 512 **(c)** MS 1024

Figure 3.23: MSER recall vs. 1-precision rates for DAL2

3.6.3 Discussion

This section presented an experimental evaluation of feature descriptors in the presence of temporal variations. The goal was to compare the descriptor performances for the previously selected feature detectors and to find the best detector/descriptor combinations for each image set.

It has been shown that the descriptor performances are tightly related with the image scene as well as the captured temporal variations. Thus, the feature matching stage is crucial for the performance and needs to be enhanced with a geometric constraint which limits the area for nearest neighbor search.

For the short-term temporal variations in the DEN1 set, Steerable Filters achieve the best performances. Particularly the combination of the MSER detector with Steerable Filters is appealing. It has been shown that this combination achieve high recall rates at a significantly lower false positive rate. Hence, for these types of temporal variations, this detector/descriptor combination is the best choice.

This recommendation changes for the variations in the DAL1 and DAL2 sets. In contrast to the DEN1 set, where Steerable Filters clearly achieved the best results, SURF outperforms the other descriptors for these image sets. Furthermore, SURF descriptors are more compact (i.e. 64-dimensional vectors) than SIFT descriptors (i.e. 128-dimensional vectors) or Steerable Filters (i.e. 312 dimensions) and due to the use of integral images they can be computed with less computational costs. Thus, the SURF descriptor is the best choice not only for the DAL1 and DAL2 sets, but also for a more general setup.

3.7 Conclusion

This chapter presented an experimental evaluation of local feature detectors and descriptors. For the detectors it is important to find a reasonable trade-off between a high repeatability rate and a large number of correspondences. As it was shown in Section 3.5.2 the number of correspondences on the one hand and the repeatability rate at the other hand are competing properties and they cannot be settled simultaneously. Hence, the choice depends on the actual application and the expected image degradations.

Detectors Maximally Stable Extremal Regions (MSERs) achieve the best repeatability rates for all image sets. Harris and Hessian perform similar, though, Harris corners tend to offer a better trade-off between repeatability and the total number of correspondences. However, it does not suffice to analyze the repeatability scores on their own: It turns out that MSERs are not applicable to the seasonal variations of the `DAL2` image set. Although they achieve the highest repeatability in terms of region overlap, their centers of gravity may not be accurately repeated. Furthermore, it has been shown that applying TV-L^1 pre-processing before using the Harris detector significantly increases the repeatability rates while decreasing the absolute number of correspondences. According to Section 3.5.4 the following detectors are proposed (cf. Table 3.33). For the `DEN1` and `DAL1` image sets the first choice is the MSER detector with minimum region size of 256 pixels and the stability parameter Δ set to 8. For the `DAL2` image set and a more general setup, the Harris detector (with detection scale σ set to 1.4 and a cornerness threshold of 1024) is proposed be used with TV-L^1 pre-processed data (where the parameter λ is fixed to 0.5) in order to achieve reasonable results.

Descriptors and Matching Strategy The evaluation results in Section 3.6 make two points clear: First, with increasing temporal variations image matching becomes more and more both a detection *and* a description problem. Second, the feature matching strategy plays a major role in the image matching workflow. A naive approach would use nearest neighbor matching with tight thresholds in order to compute putative feature matches. However, it may happen that *corresponding* features are not even identified as nearest neighbors. In order to overcome this problem, the matching strategy is required to incorporate a kind of geometric restriction constraint that limits the area for nearest neighbors search. It has been shown, that even weak restrictions to a search area of 512 × 512 pixels improve the performance. Following the discussion of Section 3.6.3 is is proposed to use Steerable Filters for the `DEN1` set and the SURF descriptor for `DAL1`, `DAL2`, and more general setup. Tables 3.33 and 3.34 summarize the resulting detector/descriptor combinations with specific parameter settings.

Image Set	Detector	Parameter Settings
DEN1	MSER	MS = 256, $\Delta = 8$
DAL1	MSER	MS = 256, $\Delta = 8$
DAL2	Harris	$\sigma = 1.4$, $t = 1024$, $\lambda = 0.5$
General	Harris	$\sigma = 1.4$, $t = 1024$, $\lambda = 0.5$

Table 3.33: Proposed feature detectors.

Image Set	Descriptor	Parameter Settings
DEN1	Steerable Filters	ellipse scale = 3
DAL1	SURF	127 × 127 pixels footprint
DAL2	SURF	127 × 127 pixels footprint
General	SURF	127 × 127 pixels footprint

Table 3.34: Proposed feature descriptors.

Chapter 4

A Temporal Insensitive Aerial Image Matching Workflow

Contents

The previous chapter presented a detailed performance evaluation of local features. Based on these evaluation results, a fully automated temporal insensitive image matching workflow – which actually a system of existing approaches – is developed in this chapter. Section 4.1 gives a brief introduction to the components of the proposed algorithm and in Section 4.2 the performance of the algorithm is measured by comparing the estimated fundamental matrices with the ground truth.

4.1 Processing Stages

Figure 4.1 shows the block diagram of the proposed workflow. The gray box indicates that this stage is either optional or not used for all settings. The remainder of this section gives a brief overview of the individual stages of the workflow.

Stage 1: TV-L^1 Pre-processing This processing stage can be seen as optional, because it does not necessarily improve the accuracy of the estimated fundamental matrix. However, the pre-processing step is very useful to strip down the computational complexity for both the feature matching and fundamental matrix estimation stages. According to Section 3.7 it is proposed to use TV-L^1 pre-processing with parameter λ set to 0.5 in combination with the Harris corner detector.

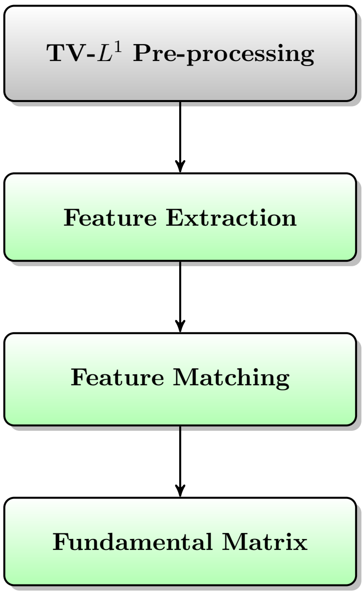


Figure 4.1: Block diagram of the proposed workflow.

Stage 2: Feature Extraction For the `DEN1` and `DAL1` image sets the MSER detector in combination with Steerable Filters was proposed. However, it turned out that MSERs are not applicable to the `DAL2` set (cf. Section 3.5 for further details). The Harris detector achieved good results for all image sets. Hence, this stage utilizes the Harris detector in combination with the SURF descriptor. The detection scale σ is set to 1.4 and the cornerness threshold value is 1024. For each detected interest point a descriptor vector is computed with the rotation variant version of the SURF descriptor and the descriptor footprint is set to a patch size of 127×127 pixels.

Stage 3: Feature Matching Different matching strategies have been already discussed in Section 3.3.2. Based on the evaluation results in Section 3.6, it is proposed to use nearest neighbor matching with a geometric constraint: In order to be independent from the actual image content the nearest neighbor search is restricted to features which are located in a 512×512 pixel neighborhood of the projected reference feature.

Stage 4: Robust Fundamental Matrix Estimation The set of putative feature matches is verified by a robust estimation of the underlying epipolar geometry based on the RANdom SAmple Consensus (RANSAC) algorithm [8]. A candidate match is defined as correct if the distance of the feature location to the estimated epipolar line is less than a pixel. In order to further improve the estimation quality, RANSAC is applied to the set of correct matches to re-estimate the fundamental matrix.

4.2 Performance Comparison with Ground Truth

In this section, the performance of the algorithm is demonstrated by comparing the estimated fundamental matrices with their ground truth counterparts. Therefore, the distances of manually verified correspondences to both the ground truth and the estimated epipolar lines are measured. Tables 4.1 to 4.3 list the measured values in terms of the mean distance and the standard deviation in pixels. Furthermore, each table shows the number of putative correspondences (i.e. resulting from stage 3) and the final number of inliers after the fundamental matrix was estimated.

Image	**Ground Truth**		**Estimated**		**Correspondences**	
Pair	**mean**	**std dev**	**mean**	**std dev**	**putative**	**inliers**
Fig. 4.2 (a) - (b)	0.24	0.14	0.27	0.18	307	191
Fig. 4.2 (c) - (d)	0.25	0.14	0.31	0.24	178	124
Fig. 4.2 (e) - (f)	0.24	0.14	0.24	0.18	173	119
Fig. 4.2 (g) - (h)	0.24	0.15	0.36	0.24	178	128

Table 4.1: Accuracy and correspondences of the estimated epipolar geometry for `DEN1`.

Image	**Ground Truth**		**Estimated**		**Correspondences**	
Pair	**mean**	**std dev**	**mean**	**std dev**	**putative**	**inliers**
Fig. 4.3 (a) - (b)	0.25	0.14	0.36	0.27	72	63
Fig. 4.3 (c) - (d)	0.22	0.14	0.38	0.36	74	58
Fig. 4.3 (e) - (f)	0.24	0.13	0.35	0.28	59	53
Fig. 4.3 (g) - (h)	0.26	0.14	0.41	0.39	63	56

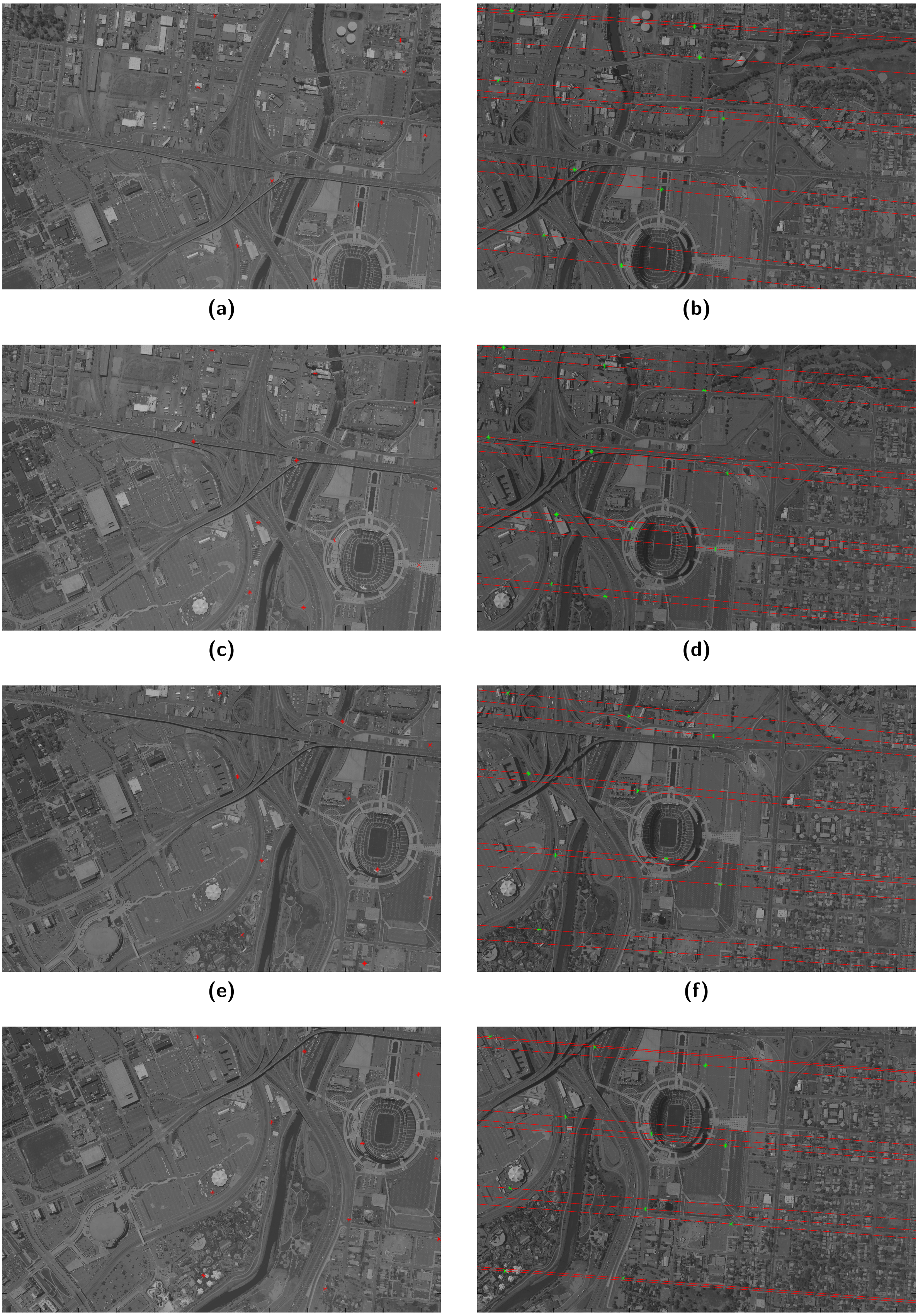
Table 4.2: Accuracy and correspondences of the estimated epipolar geometry for `DAL1`.

Image	**Ground Truth**		**Estimated**		**Correspondences**	
Pair	**mean**	**std dev**	**mean**	**std dev**	**putative**	**inliers**
Fig. 4.4 (a) - (b)	0.23	0.13	0.45	0.42	67	48
Fig. 4.4 (c) - (d)	0.22	0.13	0.28	0.23	75	53
Fig. 4.4 (e) - (f)	0.22	0.14	0.36	0.28	80	51
Fig. 4.4 (g) - (h)	0.21	0.13	0.35	0.26	73	50

Table 4.3: Accuracy and correspondences of the estimated epipolar geometry for `DAL2`.

The results clearly demonstrate the robustness of the proposed workflow to different temporal changes. The overall performance of the workflow regarding both accuracy and the total number of correspondences is reasonably well. For all image pairs, the accuracy of the estimated fundamental matrices does not significantly differ from the ground truth matrices. For the `DEN1` data set, the algorithm finds 120 to 190 correspondences for different image pairs and the accuracy of the estimated fundamental matrix is convincing – the mean distance of the previously selected correspondences to the epipolar lines range from 0.27 to 0.36 pixels. Hence, the difference to the ground truth fundamental matrices is about 0.1 pixels. For `DAL1` and even for the seasonal changes in the `DAL2` set, the mean distances of the features to the corresponding epipolar lines vary from 0.35 to 0.45 pixels and differ from the ground truth by about 0.2 pixels. Note that due to the TV-L^1 pre-processing stage the ratio of putative correspondences to inliers is quite good for all image sets.

Figures 4.2 to 4.4 display the estimated fundamental matrices for images of the same scene. On the left-hand side a subset of inliers is plotted as red circles. The corresponding epipolar lines are drawn in the images on the right-hand side along with the corresponding feature locations (green circles).



(a)

(b)

(c)

(d)

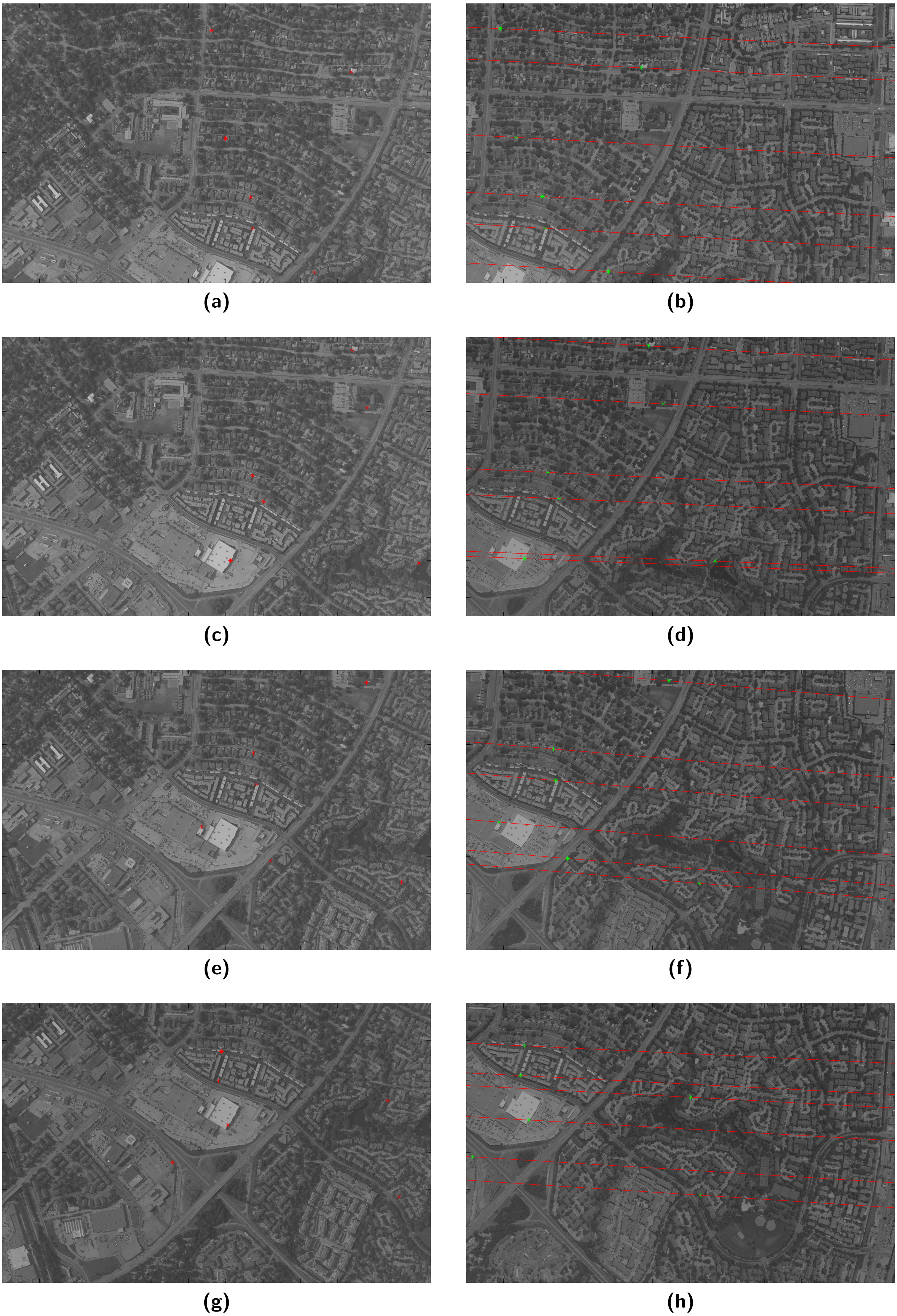
(e)

(f)

(g)

(h)

Figure 4.2: Estimated epipolar geometry for DEN1.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 4.3: Estimated epipolar geometry for `DAL1`.

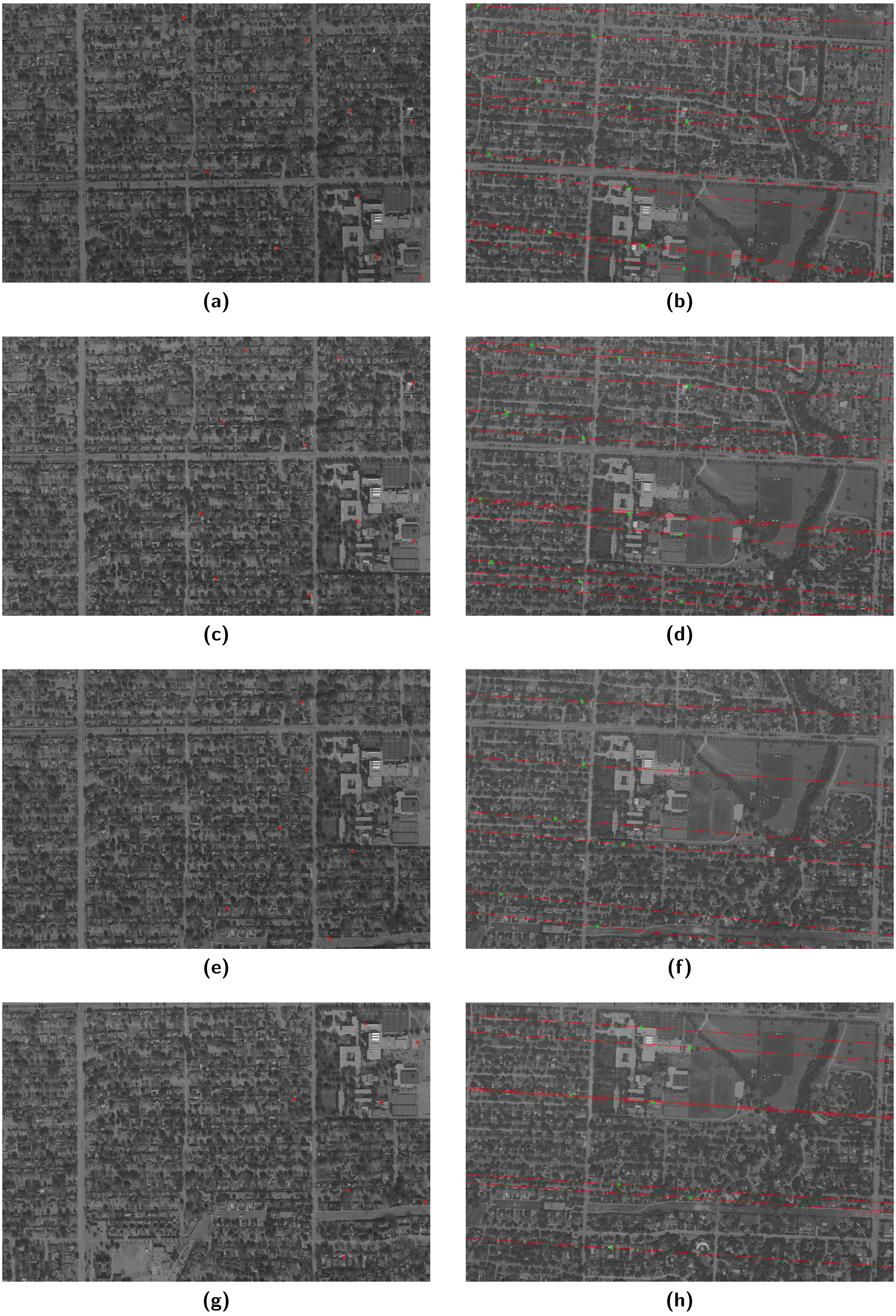


Figure 4.4: Estimated epipolar geometry for DAL2.

Chapter 5

Conclusion

Contents

5.1 Summary

This thesis presents a performance evaluation of local detectors and descriptors in the presence of temporal variations in non-planar aerial imagery. The goal was to develop a fully-automated aerial image matching workflow that can cope with substantial temporal variations which are captured in the provided real-world images.

Local features proved to be a powerful way to find correspondences in two or more images and the evaluation results in Chapter 3 show that there is a significant performance decrease in the presence of temporal variations. Parameter settings which yield a higher number of precise correspondences usually show lower repeatability rates (cf. Section 3.5.2). In this context, it was shown in Section 3.5.3 that a pre-processing step based on total variation both improves the repeatability rate and preserves a reasonable number of correspondences for the Harris detector.

For the temporal variations in the `DEN1` and `DAL1` data sets, Maximally Stable Extremal Regions (MSERs) performed best. However, it turned out that this feature detector is heavily effected by the seasonal variations captured in the `DAL2` set where MSERs proved to be inapplicable in the presence of massive appearance changes. In contrast to MSERs, the Harris corner detector achieves reasonable results for all image sets. Hence, this detector is used in the prototype implementation of a temporal insensitive aerial image matching workflow in Chapter 4.

The descriptor evaluation results in Section 3.6 showed that with increasing temporal variations image matching becomes more and more both a detection *and* a description problem. The main reason for this is an increased distance for corresponding descriptor vectors due to appearance changes that poses a problem particularly for repetitive image structures. Thus, the feature matching strategy becomes a key issue for the overall performance. It has been shown in Section 3.6.1 that restricting the area for nearest neighbor search boosts the performance significantly.

Steerable Filter descriptors perform best when they are combined with Maximally Stable Extremal Regions. In all other cases, the use of the Speeded-up Robust Features (SURF) descriptor is proposed for several reasons. First, it has been shown that SURF achieves highly competitive performances. Second, due to the use of integral images, the descriptor computation can be executed faster than for other state-of-the-art methods. Third, due to the relatively low descriptor dimensionality the computational complexity of feature matching is reduced.

The temporal insensitive aerial image matching workflow which is developed in Chapter 4 is actually a system of existing methods. In order to demonstrate the robustness of the algorithm, the estimated fundamental matrices are compared with their ground truth counterparts. Therefore, the distances of manually verified correspondences to both the ground truth and the estimated epipolar lines are measured. The overall performance of the workflow in terms of both accuracy and total number of correspondences is reasonably well (i.e. correspondence localization errors less than 0.5 pixels) and the accuracy of the estimated fundamental matrices does not significantly differ from the ground truth matrices.

5.2 Future Work

This thesis showed the capability of a feature-based temporal insensitive aerial image matching algorithm. Although the proposed workflow achieves reasonable results on the provided image sets, there is still plenty of room for improvements. Further investigations may focus the following issues.

Exploit 12 Bit Panchromatic Image Depth The panchromatic aerial imagery is originally available with a 12 bit depth. Using this information may improve the quality of both the pre-processing stage and the feature extraction stage. However, even if this is fairly easy to implement, there are increased requirements for the hardware.

Limited Repeatability and Complementary Features Even though a lot of progress was made in the field of feature extraction in the recent years, the repeatability rate of the feature detectors is still limited. A variety of detectors has been proposed and each of these has its individual strengths and weaknesses. Hence it is quite common to use several detectors in parallel. This may help to improve the repeatability and simultaneously increase the number of detected correspondences.

Enhanced Matching Strategy Another improvement might be achieved with an enhanced matching strategy. This may be either a simple cross-checking strategy where the reference feature is required to be found as nearest neighbor match of its nearest neighbor or a more advanced iterative feature matching stage where initially estimated fundamental matrices are used to iteratively refine the search for feature correspondences. For example, Yang et al. [65] used such a technique for the registration of challenging image pairs.

Increased Number of Temporal Variations and Image Scenes The performance of local features for aerial image matching depends also on the captured image scenes. Thus, it would be of interest to increase the number of test sets with different image scenes (e.g. rural areas with forests).

Bibliography

[1] ASPRS: The Imaging and Geospatial Information Society. American Society for Photogrammetry and Remote Sensing, 2010. `http://www.asprs.org/society/about.html` [Online; accessed 3-January-2010]. (Cited on page 5.)

[2] Jean-Francois Aujol, Guy Gilboa, Tony Chan, and Stanley Osher. Structure-texture image decomposition – modeling, algorithms, and parameter selection. *International Journal of Computer Vision*, 67(1):111–136, 2006. (Cited on page 25.)

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features. *Journal of Computer Vision and Image Understanding*, 110(3):346–359, 2008. (Cited on pages 14, 18, 29, 35 and 52.)

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded-up robust features. In *Proceedings of the 9th European Conference on Computer Vision*, pages 404–417, 2006. (Cited on pages 18 and 52.)

[5] P. R. Beaudet. Rotationally invariant image operators. In *Proceedings of the International Joint Conference on Pattern Recognition*, pages 579–583, 1978. (Cited on pages 9, 28 and 37.)

[6] Matthew Brown and David Lowe. Recognising panoramas. In *Proceedings of the 9th International Conference on Computer Vision*, volume 2, page 1218, 2003. (Cited on page 1.)

[7] Terrence Chen, Wotao Yin, Xiang Zhou, Dorin Comaniciu, and Thomas Huang. Total variation models for variable lighting face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1519–1524, 2006. (Cited on page 25.)

[8] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. (Cited on pages 35 and 66.)

[9] Wolfgang Förstner. A framework for low level feature extraction. In *Proceedings of the 3rd European Conference on Computer Vision*, pages 383–394, 1994. (Cited on page 12.)

[10] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proceedings of the Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, 1987. (Cited on page 12.)

[11] Friedrich Fraundorfer and Horst Bischof. Evaluation of local detectors on non-planar scenes. In *Proceedings of the Austrian Association for Pattern Recognition Workshop*, pages 125–132, 2004. (Cited on pages 2, 8, 24 and 42.)

[12] Friedrich Fraundorfer and Horst Bischof. A novel performance evaluation method of local detectors on non-planar scenes. In *Workshop Proceedings Empirical Evaluation Methods in Computer Vision, IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005. (Cited on pages 2, 8, 24 and 42.)

[13] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, Sep 1991. (Cited on pages 21 and 52.)

[14] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988. (Cited on pages 11, 28 and 37.)

[15] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. (Cited on page 35.)

[16] Institute for Computer Graphics and Vision, Graz University of Technology. Gpu4vision, 2008. `http://gpu4vision.icg.tugraz.at/` [Online; accessed 2-May-2010]. (Cited on page ix.)

[17] Andrew Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1):433 – 449, May 1999. (Cited on page 23.)

[18] Yan Ke and Rahul Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 506–513, 2004. (Cited on pages 23 and 34.)

[19] Jan Koenderink and Andrea van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375, 1987. (Cited on page 17.)

[20] Kalin Kolev, Maria Klodt, Thomas Brox, and Daniel Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 84(1):80–96, 2009. (Cited on page 25.)

[21] Peter D. Kovesi. MATLAB and Octave functions for computer vision and image processing. School of Computer Science & Software Engineering, The University of Western Australia, 2000. `http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/` [Online; accessed 2-May-2010]. (Cited on page ix.)

[22] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A sparse texture representation using affine-invariant regions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 319–324, 2003. (Cited on pages 23 and 24.)

[23] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998. (Cited on page 14.)

[24] David Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th International Conference on Computer Vision*, pages 1150–1157, 1999. (Cited on pages 2, 14, 17 and 52.)

[25] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. (Cited on pages 2, 17, 29 and 52.)

[26] Jiri Matas, Ondrej Chum, Martin Urban, and Tomas Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the 13th British Machine Vision Conference*, pages 384–393, 2002. (Cited on pages 14, 15, 28 and 37.)

[27] Microsoft Inc. Bing Maps, 2008. `http://www.bing.com/maps/` [Online; accessed 25-April-2010]. (Cited on page 8.)

[28] Krystian Mikolajczyk. Affine covariant features, 2010. `http://www.robots.ox.ac.uk/~vgg/research/affine/` [Online; accessed 19-April-2010]. (Cited on pages ix and 37.)

[29] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision*, pages 128–142, 2002. (Cited on page 16.)

[30] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 257–263, 2003. (Cited on pages 2, 8, 24, 29, 32 and 59.)

[31] Krystian Mikolajczyk and Cordelia Schmid. Comparison of affine-invariant local detectors and descriptors. In *Proceedings of the 12th European Signal Processing Conference*, 2004. (Cited on pages 8 and 32.)

[32] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004. (Cited on pages 2, 14 and 32.)

[33] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. (Cited on pages 2, 8, 22, 23, 24, 29, 32, 34 and 59.)

[34] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005. (Cited on pages 2, 8, 16, 24, 28 and 32.)

[35] Hans Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artifical Intelligence*, pages 584–590, 1977. (Cited on pages 9 and 11.)

[36] Mila Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20(1-2):99–120, 2004. (Cited on page 25.)

[37] David Nistér and Henrik Stewénius. Linear time maximally stable extremal regions. In *Proceedings of the 10th European Conference on Computer Vision*, pages 183–196, 2008. (Cited on page 15.)

[38] Stepan Obdrzalek and Jiri Matas. Object recognition using local affine frames on distinguished regions. In *Proceedings of the 13th British Machine Vision Conference*, pages 113–122, 2002. (Cited on page 15.)

[39] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio. Pedestrian detection using wavelet templates. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 193–199, 1997. (Cited on page 19.)

[40] Constantine Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. *Proceedings of the 6th International Conference on Computer Vision*, pages 555–562, 1998. (Cited on page 19.)

[41] Thomas Pock. *Fast Total Variation for Computer Vision.* Phd thesis, Graz University of Technology, 2008. (Cited on page 25.)

[42] Thomas Pock, Markus Unger, Daniel Cremers, and Horst Bischof. Fast and exact solution of total variation models on the GPU. In *CVPR Workshop on Visual Computer Vision on the GPU*, 2008. (Cited on page 25.)

[43] Thomas Pock, Martin Urschler, Christopher Zach, Reinhard Beichel, and Horst Bischof. A duality based algorithm for TV-L^1-optical-flow image registration. In *Proceedings of the 10th International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 511–518, 2007. (Cited on page 25.)

[44] Richard J. Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, 2005. (Cited on page 1.)

[45] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Proceedings of the 10th International Conference on Computer Vision*, pages 1508–1515, 2005. (Cited on page 16.)

[46] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision*, pages 430–443, 2006. (Cited on page 16.)

[47] Peter M. Roth and Martin Winter. Survey of appearance-based methods for object recognition. Technical report, Institute for Computer Graphics and Vision, Graz University of Technology, 2008. (Cited on page 29.)

[48] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992. (Cited on page 25.)

[49] Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *Proceedings of the 7th European Conference on Computer Vision*, pages 414–431, 2002. (Cited on page 24.)

[50] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997. (Cited on page 17.)

[51] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Comparing and evaluating interest points. In *Proceedings of the 5th International Conference on Computer Vision*, pages 230–235, 1998. (Cited on page 32.)

[52] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000. (Cited on pages 2, 8, 24 and 28.)

[53] Stephen M. Smith and Michael J. Brady. SUSAN – a new approach to low level image processing. *International Journal of Computer Vision*, 23(34):45–78, 1997. (Cited on page 16.)

[54] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. In *Proceedings of the ACM Special Interest Group on Graphics and Interactive Techniques*, pages 835–846, 2006. (Cited on page 1.)

[55] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991. (Cited on page 12.)

[56] Bill Triggs, Philip Mclauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory & Practice*, pages 298–375. Springer, 2000. (Cited on page 8.)

[57] Tinne Tuytelaars and Luc Van Gool. Content-based image retrieval based on local affinely invariant regions. In *International Conference on Visual Information Systems*, pages 493–500, 1999. (Cited on page 16.)

[58] Tinne Tuytelaars and Luc Van Gool. Wide baseline stereo matching based on local affinely invariant regions. In *Proceedings of the 11th British Machine Vision Conference*, pages 412–425, 2000. (Cited on page 16.)

[59] Tinne Tuytelaars and Krystian Mikolajczyk. *Local Invariant Feature Detectors: A Survey.* Now Publishers Inc., Hanover, MA, USA, 2008. (Cited on pages 2, 8, 24 and 28.)

[60] Andrea Vedaldi and Brian Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008. `http://www.vlfeat.org/` [Online; accessed 13-April-2010]. (Cited on pages ix and 52.)

[61] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001. (Cited on pages 14 and 18.)

[62] Willow Garage. The OpenCV computer vision library, 2008. `http://opencv.willowgarage.com/` [Online; accessed 2-May-2010]. (Cited on page ix.)

[63] Simon Winder and Matthew Brown. Learning local image descriptors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. (Cited on pages 23, 29 and 52.)

[64] Simon Winder, Gang Hua, and Matthew Brown. Picking the best daisy. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami, June 2009. (Cited on pages 23, 29 and 52.)

[65] Gehua Yang, Charles Stewart, Michal Sofka, and Chia-Ling Tsai. Registration of challenging image pairs: Initialization, estimation, and decision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1973–1989, 2007. (Cited on pages 2 and 72.)