Diss. ETH No. 16606

From Wide-baseline Point and Line Correspondences to 3D

A dissertation submitted to the
Swiss Federal Institute of Technology, ETH Zurich

for the degree of
Doctor of Sciences

presented by
HERBERT BAY
Dipl. Ing. Microtechnique EPFL
born June 13, 1974
citizen of Truttikon, ZH, Switzerland

accepted on the recommendation of
Prof. Dr. Luc Van Gool, examiner
Prof. Dr. Cordelia Schmid, co-examiner

2006

Abstract

In this thesis, we tackle the problem of point and line segment correspondences between two images. The images are supposed to be taken under wide-baseline conditions: a fact which represents an additional challenge for reliable feature matching. Image feature correspondences are used for many computer vision tasks. We demonstrate the performance of our approach on some practical applications. In particular, we study the problem of 3D reconstruction from only two views using matched points and line segments.

The most interesting type of image features in the search for correspondences are points. Not only their locations are geometrically constrained between two images of the same rigid scene, but also they are simple to detect and thus applicable to solve many problems. Hence, multiple methods for finding interest point correspondences exist. Our method outperforms all these schemes while being much simpler and therefore faster to compute.

Few approaches were proposed to solve the problem of line segment correspondences between two images. The absence of a geometrical constraint for line matches in two views made this problem considerably more challenging. However, we solved that problem by introducing, besides a novel appearance-based descriptor, constraints on the topological configuration of the line segments. To the author's knowledge, we are the first to address the wide-baseline case without any prerequisite.

Line correspondences are of particular interest for poorly-textured, man-made environments, where the number of detected interest points is typically low. The absence of interest points makes camera calibration and 3D modelling from two views a considerably more difficult task. We present a new method for complete reconstruction of such scenes based on line segment correspondences between two wide-baseline views. Moreover, using this method, point and line segment correspondences can be used in a unified manner.

Zusammenfassung

In dieser Dissertation wird das Problem der Bildung von Punkt- und Linien-Korrespondenzen zwischen zwei Bildern studiert. Dabei wird angenommen, dass die Bilder von weit auseinanderliegenden Standorten aufgenommen wurden. Dies ist eine zusätzliche Herausforderung für die zuverlässige Erkennung von Bildkorrespondenzen, die in vielerlei Anwendungen im Bereich der Bildverarbeitung unerlässlich sind. Wir schlagen eigene Lösungsansätze vor und demonstrieren deren Leistungsfähigkeit anhand praktischer Beispiele. Insbesondere studieren wir das Problem der Kamera-Kalibrierung und der 3D Rekonstruktion basierend auf Punkt- und Linien-Korrespondenzen zwischen lediglich zwei Ansichten.

Bei der Suche nach Bildkorrespondenzen sind Punkte besonders attraktiv, da auffallende Punkte sich im Allgemeinen relativ einfach detektieren lassen und der Suchbereich zwischen Bildern einer statischen Szene geometrisch eingeschränkt ist. Diese vorteilhaften Eigenschaften resultieren in einem breiten Anwendungsspektrum für Punktkorrespondenzen. Deshalb wurden auch bereits viele verschiedene Methoden für deren Erkennung und die zugehörige Korrespondenzsuche zwischen zwei Bildern vorgeschlagen. Unser Ansatz schlägt alle bisherigen Methoden in Rechengeschwindigkeit und Leistungsfähigkeit.

Nur wenige Lösungsansätze wurden bisher für das Linien-Korrespondenz-Problem präsentiert. Das Fehlen von geometrischen Zusammenhängen zwischen zwei Bildern erschwert die Aufgabe gegenüber dem Punkt-Korrespondenzen erheblich. Als Ersatz greifen wir auf das Erscheinungsbild und die topologische Anordnung der Linien zurück, die das Linien-Korrespondenz-Problem ausreichend begrenzen, um es lösen zu können. Dem Autor sind keine anderen Arbeiten bekannt, in denen das Linien-Korrespondenz-Problem für ähnliche Konfigurationen gelöst wird, ohne allzu starke Annahmen zu treffen.

Linienkorrespondenzen sind von besonderem Interesse in schlecht texturierten, architektonischen Umgebungen, wo die Anzahl detektierter Punkte typischer-

weise sehr gering ist. Ohne ausreichend viele solcher Punkte wird Kamera-Kalibrierung und damit auch 3D Rekonstruktion einer Szene beträchtlich erschwert. Wir präsentieren einen neuen Ansatz, der eine vollständige Rekonstruktion solcher untexturierten Szenen ermöglicht. Dieser beruht auf Linienkorrespondenzen zwischen zwei weit auseinanderliegenden Ansichten einer starren Szene. Ausserdem erlaubt unsere Methode, Punkt- und Linien-Korrespondenzen gleichermassen zu berücksichtigen.

Acknowledgements

During my time at the Computer Vision Lab, I had the chance to meet many wonderful and interesting people who inspired me or contributed to the accomplishment of my PhD thesis.

First of all, I would like to thank my supervisor Prof. Luc Van Gool for his valuable inputs and for having pushed me to go always a bit farther. I am also grateful for the numerous fruitful discussions we had, always admixed with a bit of dry, British humour in order to render the atmosphere more relaxed.

I most acknowledge the collaboration with Dr. Tinne Tuytelaars who showed me how well papers can be written and that there are people with a never-ending source of creativity and lots of good ideas.

Parts of this thesis would not have been possible without the inputs of Dr. Vittorio Ferrari. With his imperturbable optimism, he showed me that writing is actually a funny game about finding the best words to describe something properly.

I am grateful to Dr. Beat Fasel for many exciting discussions about Computer Vision, his major contribution to some of my work, and his encouraging words when things seemed to go in the wrong direction.

Special thanks go to Prof. Cordelia Schmid for having accepted to be my co-referee and for reading this thesis within a very short delay, to Dr. Bastian Leibe for his helpful support, to Dr. Krystian Mikolajczyk for answering my numerous e-mails, and to Prof. Jiří Matas for some constructive remarks on parts of my work.

Without the friendly atmosphere, created by my office mates Dr. Esther Koller-Meier, Peter Leškovský, Peter Čech, and Andreas Ess, my time at the institute would have been less enjoyable. From all of my colleagues at BIWI, I would like to specially thank again Andreas Ess who made an essential contribution

to my thesis with his programming skills and his endless assiduity. Manuel Oetiker for being the world best system administrator. Alexander Neubeck for his genius and the precious algorithmic advises for speeding up numerous procedures. Roland Kehl and Andreas Griesser for the Thursday evening beers. Stephan Tuchschmid for initiating me to Kite Surfing. Dr. Philippe Cattin for his motivation to keep on practising sports especially during stressful periods. Dr. Alexey Zalesny for organising the seminars and supporting me with the idea of the reading group. Gerald Bianchi for some support on Firewire cameras and speaking French with me from time to time. Pascal Müller for the nice conversation about the Roman empire. Stefan Saur for compiling SURF under MS Windows. Till Quack for the Symbian implementation of SURF. Ana Cristina Murillo for the spatio-temporal approach of SURF. Tijl Vereenooghe for the SURF website and for speaking Dutch with me. Jutta Spanzel, Barbara Widmer and Vreni Vogt for their administrative support. The team at VISICS in Leuven, especially Maarten Vergauwen, Nico Cornelis, Dr. Thomas Koninckx, Dr. Indra Geys-Koninchx, Toon Goedemé, Stefaan De Roeck and Egemen Özden.

Finally, this thesis would not have been possible without the genes of my mother Hedy Bay, and the moral support as well as the continuous motivation of my beloved wife Dr. Asma Jebali Bay.

Contents

List of Figures

List of Tables

1

Introduction

"Simplicity is the ultimate sophistication."
– Leonardo da Vinci (1452-1519)

Looking at a picture, one is able to tell apart different objects and estimate depth without having any other information about the imaged scene. This is the fruit of a livelong learning process during which the observer, sometimes unconsciously, encounters and studies different object classes. It is therefore easy for a human to distinguish different objects, even on abstract paintings. However, it would be almost impossible to accurately reverse engineer a scene or object only based on images of the latter.

It has taken some of the greatest minds to understand the basic processes of perspective projection, that form the link between 3D reality and the 2D projections we have to work from. This process of projection, combined with the illumination conditions, lets appear the same part of a scene very different in different images. Being able to discard irrelevant changes in appearance is one of computer vision's grand challenges up to this day.

1.1 From Vitruvius To Our Days

The technique of perspective projection was already known to the ancient Greeks and Romans. The Roman architect Vitruvius published specifications about perspective drawings already in about 25 BC. In his ten books on architecture [Vitruvius 1914] he wrote, *Perspective is the method of sketching a*

front with the sides withdrawing into the background, the lines all meeting in the centre of a circle.

In Pompeii, a Roman city which was buried and lost under many meters of ash from 79 AD to 1748, the well conserved wall paintings still witness of the Roman knowledge about perspective projection. In particular, a Roman wall painting in Villa Poppaea in Oplontis, near Pompeii, shows a quite precise representation of an architectural scene (see figure 1.1).

Figure 1.1: A Roman wall painting of architectural features in Villa Poppaea in Oplontis.

During the *Dark Ages*, the Roman knowledge about perspective projection has been lost: art was handmaiden of religion. The call for a "return to nature" of the Renaissance was the reincarnation of the quest for the rules of perspective.

In the 13th and 14th century, painters like Giotto di Bondone (1267-1337) experimented with the illusionary representation of a three-dimensional space on a two-dimensional surface (see figure 1.2). Nevertheless, the projections had no defined vanishing points, nor a genuine sense. Perspective still had to be studied geometrically.

The Italian architect Filippo Brunelleschi (1377-1446) was the first to work out the principles of perspective. He created a demonstration panel with specific viewing constraints for complete accuracy of reproduction. In 1435, Leono Battista Alberti (1404-1472) published a treatise on perspective "Della Pittura" [Alberti 1977]. *A painting* (the projection plane) *is the intersection of*

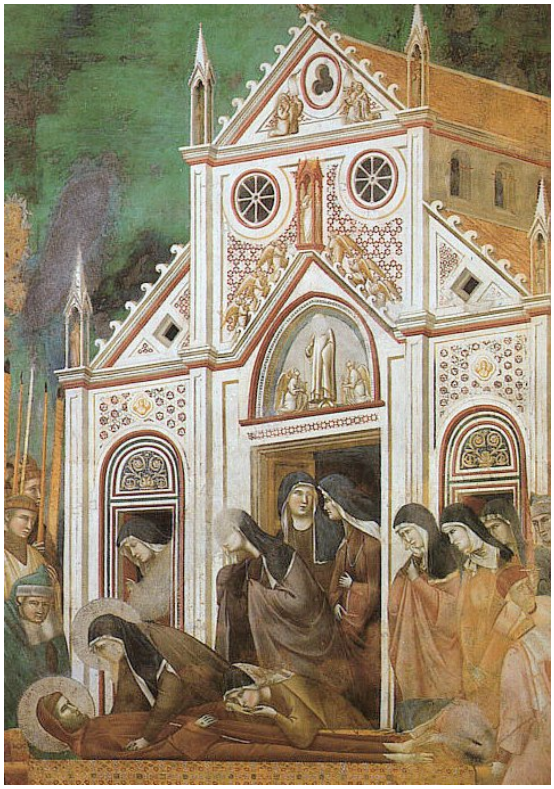

Figure 1.2: Giotto di Bondone, *St.Francis Mourned by St.Clare*, around 1300, Fresco, $270 \times 230\,\mathrm{cm}$. Upper Church, San Francesco, Assisi

a visual pyramid (view volume) *at a given distance, with a fixed centre* (centre of projection) *and a defined position of light, represented by art with lines and colours on a given surface* (the rendering).

Albrecht Dürer (1471-1528) described the concept of similar triangles geometrically and mechanically in widely read publications. The projection of a point from the scene on a canvas became possible using the proposed methods (see figure 1.3).

From that time on, many painters and scientists were using and developing the technique of perspective painting. Leonardo Da Vinci's masterpiece "The Last Supper" is an excellent example of that period. Also, in his notes, Leonardo Da Vinci (1452-1519) described something which would get a high impact on the field of computer vision: the *camera obscura* (dark chamber). Around 1560, Giambattista Della Porta (1535-1615) reinvented and amended the camera obscura with a lens and helped to spread its name. In the 16th century, the field of perspective projection with the pinhole-camera model was geometrically fully

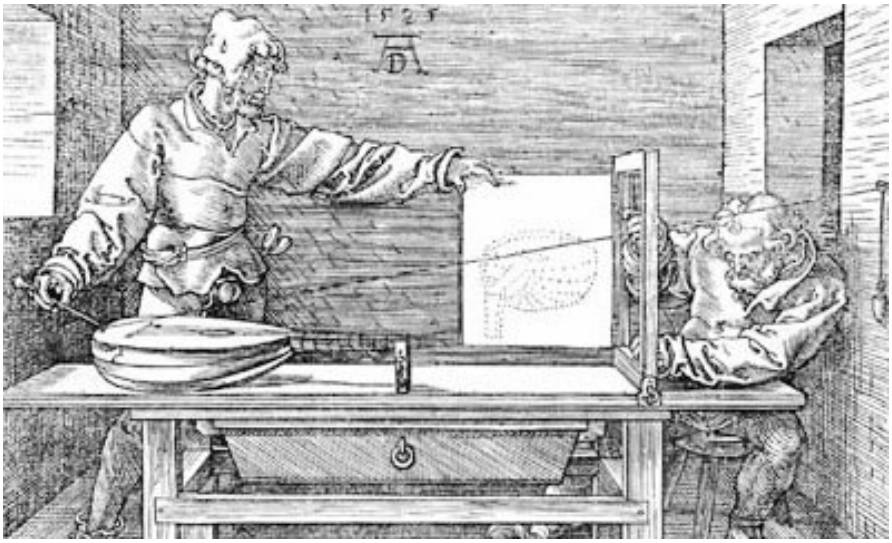

Figure 1.3: Albrecht Dürer, *Unterweysung der Messung*, Woodcut, Nürnberg, 1525.

explored. The camera obscura was used to project the scene to be painted on a paper on which an artist could then copy the image. The Dutch Masters, such as Johannes Vermeer (1632-1675), who were hired as painters in the 17th Century, were known for their magnificent attention to details. It has been widely speculated that they made use of such a camera, but the extent of its use by artists at this period remains a matter of considerable controversy.

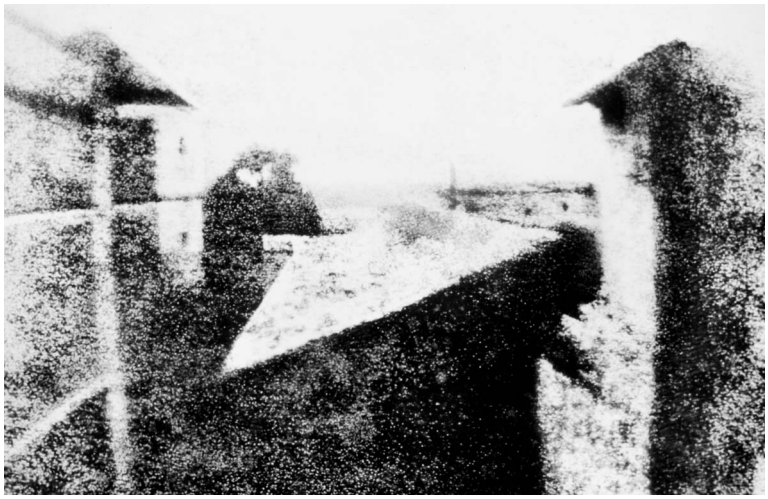

Figure 1.4: The first photograph by Nicéphore Niépce. It shows the view of a window in Le Gras.

The first photograph, shown in figure 1.4, was taken in 1826 by the French inventor Nicéphore Niépce. It was exposed during eight hours to bright sunlight. Therefore, Nicéphore called this procedure ”heliography” which means ”sun writing”. From then on, the development of photography continued in a fast pace until our days.

In 1838 Sir Charles Wheatstone invented stereoscopy. It creates the illusion of depth by looking simultaneously through each eye at a different image of the same scene. The images have to be taken from a slightly translated viewpoint. This technique is still used for creating nice effects in movies, computer games, product presentations, etc.

With the advent of computers and digital imaging, it became possible to measure 3D positions of points in the scene which are visible in two or more images taken from different view points. This technique is known as *photogrammetry*. The presence and identification of multiple such points (also called interest points) allow for the calibration of the cameras i.e. the positions from which the images were taken, the focal point of the camera etc. Camera self-calibration is often used in movie post-production, where computer generated objects are combined with the photographed scene (augmented reality).

1.2 Image Correspondences

With the cameras calibrated, it is possible to generate a dense 3D model of a filmed scene. This process takes, as it was the case for the first photographs, still a long time until every pixel has found its place in space. The important computation time is due to the high number of images and points to be considered.

Recently, new methods for automatic interest point identification and matching were suggested [Tuytelaars and Van Gool 2000, Lowe 2004, Matas *et al.* 2002, Mikolajczyk and Schmid 2004]. These methods are able to cope with a more important view change (baseline) between two successive images. With such a method, 3D modelling is possible from only a few images taken with a standard photo camera (see figure 1.5 for an example). Another advantage is the high resolution of still images compared to video frames. A wider baseline means at the same time more spatial information, but also a higher risk of occlusions, more mismatches, and less accurate interest point localisation. Therefore, the

main challenge for automatic camera self-calibration and 3D reconstruction is to find point correspondences between two or more images of the same scene or object under wide-baseline conditions. In particular, strategies for finding interest points and their correspondences should have the following properties.

- Accuracy of the interest point localisation
- Invariance to scale changes (zoom) within a certain range
- Invariance to in-plane rotations (rotation around the optical axis)
- Stability against changes in lighting and contrast
- Steadiness towards affine or projective transformations

Point matches between images or objects are very useful for many practical computer vision applications. One example is Object Recognition. Suppose

Figure 1.5: Dense 3D model of the city hall in Leuven. This model has been automatically created from 5 images, taken from substantially different view points, in less than a second.

one takes a view of a specific object, let's say a Teddy bear. In order to recognise this same toy in a second image, containing many other additional objects, the interest points of the bear are matched with the interest points of the second image. If there is a high number of correspondences between the two images, the Teddy bear is most probably present in the second image. Moreover, point matches can be verified geometrically between two views if the object is rigid. This helps to identify the object in a more reliable manner. With larger image databases, speed becomes certainly an issue. Therefore, a method for fast identification and matching of the interest points is required.

The number of automatically detected interest points in an image depends directly on the amount of texture present in the imaged scene. In a picture of a homogeneous surface e.g. a blank sheet of paper, there is nothing to *grab hold* of in that, there are no distinctive intensity pattern. Images of man-made environments, like architectural interiors, are often very poorly textured. Entire walls may be blank. Therefore, the number of extracted interest points is too small for many applications, including robust camera calibration, not to mention dense 3D modelling. In such situations, image correspondences based on line segments are more appropriate. Line segments are ubiquitous in man-made environments and can be automatically detected. The drawback however is the absence of a geometrical constraint for line matches in two views. This makes the search for correspondences more challenging. Moreover, in order to create a dense 3D model of a poorly textured scene, some prior knowledge about the planar nature of the scene has to be used.

This thesis presents novel methods and algorithms to solve the problem of finding point and line correspondences between two images. The images are supposed to be taken under wide-baseline conditions. Furthermore, we show how the structure-from-motion problem can be addressed from only two images of textured and untextured scenes.

1.3 Our Contribution

The main contributions of the thesis are:

- *A fast interest point detector* that detects blob-like features on different scales at nearly real-time speed on a standard PC. The detector is based on the Hessian matrix which is approximated using simple box filters.

The important speed gain is due to the use of integral images for the computation of these filters.

- *A powerful descriptor scheme* based on Haar wavelet responses. The wavelet responses are also computed using integral images for minimal computational costs. The descriptor is very simple and performant at the same time. The percentage of correct matches is typically higher than for competing techniques. In short, the method yields better matches at higher speeds. Also, the dimension of the descriptor is low when compared to what other schemes need to arrive at similar performances. As a result, the matching is performed in lower-dimensional spaces, which yields a higher speed. The matching is further accelerated by considering the contrast of the interest point i.e. dark interest point on a light background or vice versa. This results in an additional doubling of the matching speed.

- *A method for matching line segments* between two images taken under wide-baseline conditions. Exploring both appearance and topology leads to a robust matching strategy for line segments. Moreover, it is possible to use additional information provided by interest point correspondences which not only reduces the number of mismatched lines segments, but also increases the number of correct matches. This is due to a topological filter that takes the respective topological configuration of line segments and interest points into account. We also propose a method to increase the number of matches even further by exploring topological information to reduce the search space for candidate matches. Furthermore, we consider vanishing points as another additional constraint for a novel vanishing point filter which reduces the number of mismatches even further. Note that the epipolar geometry is neither known nor used.

- *A strategy for the construction of dense 3D models* of poorly-textured architectural scenes from two images. To tackle this challenging problem, a novel method for estimating the epipolar geometry from line segments had to be developed. The imaged scene is simultaneously clustered into coplanar regions while polyhedral junctions are detected. The junctions yield point correspondences and can be used, together with possibly other detected interest points, to estimate the epipolar geometry. The planar regions are used for the partially planar 3D model. Also, we pro-

pose a new method performing bundle adjustment for line segments in only two images.

1.4 Road Map

The whole thesis is split into three main parts. In the first part, we address the problem of point (also called region) correspondences between two images. For that task, we present a scale and rotation invariant interest point detector (chapter 2) and descriptor (chapter 3) scheme, coined SURF (Speeded Up Robust Features). Its performance is demonstrated with some standard computer vision applications such as real-life object recognition and mosaicing for medical applications (chapter 4).

The second part describes the strategy for finding line segment correspondences between two images. The line segments are first characterised with a descriptor based on the colour distribution in their neighbourhood. However, appearance-based line matching is not distinctive enough for wide-baseline conditions. Therefore, we introduce a line segment matcher based on appearance and the overall topological configuration (chapter 5).

The third part shows the use of point and line correspondences for camera self-calibration and 3D reconstruction. We limit ourselves to two views for both 3D from interest points (chapter 6) and 3D from line segments (chapter 7). We also show the advantage of a combined approach using interest points and line segments in a unified manner.

I

Point Correspondences

Outline

The task of finding point correspondences between two images of the same scene or object is part of many computer vision applications. Camera calibration, 3D reconstruction, image registration, and object recognition are just a few. The search for discrete image point correspondences – the goal of this part of the thesis – can be divided into three main steps. First, 'interest points' are selected at distinctive locations in the image, such as corners, blobs, and T-junctions. The most valuable property of an interest point *detector* is its repeatability. The repeatability expresses the reliability of a detector for finding the same physical interest points under different viewing conditions. Next, the neighbourhood of every interest point is represented by a feature vector. This *descriptor* has to be distinctive and at the same time robust to noise, detection displacements and geometric and photometric deformations. Finally, the descriptor vectors are *matched* between different images. The matching is often based on a distance between the vectors, e.g. the Mahalanobis or Euclidean distance. The dimension of the descriptor has a direct impact on the time this takes, and lower numbers of dimensions are desirable for fast interest point matching. However, lower dimensional feature vectors are in general less distinctive than their high-dimensional counterparts.

It has been our goal to develop both a detector and descriptor that, in comparison to the state-of-the-art, are fast to compute while not sacrificing performance. In order to succeed, one has to strike a balance between the above requirements like reducing the descriptor's size and complexity while keeping it sufficiently distinctive.

A wide variety of detectors and descriptors have already been proposed in the literature (e.g. [Lindeberg 1998, Lowe 2004, Mikolajczyk and Schmid 2002, Se *et al.* 2004, Tuytelaars and Van Gool 2000, Matas *et al.* 2002]). Also, detailed comparisons and evaluations on benchmarking datasets have been performed [Mikolajczyk and Schmid 2003, Mikolajczyk and Schmid 2005,

Mikolajczyk *et al.* 2005]. While constructing our fast detector and descriptor, we built on the insights gained from this previous work in order to get a feel for what are the aspects contributing to performance. In our experiments on these benchmarking datasets, the resulting detector and descriptor are not only faster, but the former is also more repeatable and the latter more distinctive.

When working with local features, a first issue that needs to be settled is the required level of invariance. Clearly, this depends on the expected geometric and photometric deformations, which in turn are determined by the possible changes in viewing conditions. Here, we focus on scale and in-plane rotation invariant detectors and descriptors. These seem to offer a good compromise between feature complexity and robustness to commonly occurring deformations. Skew, anisotropic scaling, and perspective effects are assumed to be second-order effects, that are covered to some degree by the overall robustness of the descriptor. Concerning the photometric deformations, we assume a simple linear model with a bias (offset) and contrast change (scale factor). Our detector and descriptor don't use colour.

In chapter 2, we describe the strategy applied for fast and robust interest point detection. The input image is analysed on different scales in order to guarantee invariance to scale changes. The detected interest points are provided with a rotation and scale invariant descriptor in chapter 3. Furthermore, a simple and efficient first-line indexing technique, based on the contrast of the interest point with its surrounding, is proposed. The resulting detector/descriptor scheme is tested for different specific computer vision applications in chapter 4. There we show the benefits in terms of speed and robustness of an upright version (not invariant to image rotation) of our descriptor for applications where the camera movement is restricted to rotate only about the vertical axis.

2

Interest Point Detection

Under interest points, we understand small image regions with high changes of the local gradient in two distinctive directions. Such points can be reliably extracted and provide a high amount of information. Furthermore, interest points are often robust to various transformations, e.g. rotation, scale and partially affine transformations. As also claimed by David Lowe [Lowe 2004], the additional complexity of full affine-invariant features often has a negative impact on their robustness and does not pay off, unless really wide viewpoint changes ($> 40°$) are to be expected. Therefore, We designed our interest point detector to be only invariant to changes in scale and in-plane rotation.

There are different methods to extract such interest points. In general, every method detects interest points of a specific nature like blobs, corners, etc. Our detector is based on the Hessian matrix, just as [Mikolajczyk and Schmid 2001, Lindeberg 1998], and detects therefore blob-like features. The difference in our approach is that we use a very basic Hessian-matrix approximation, just as the *Difference of Gaussians* blob detector by David Lowe [Lowe 2004] is a very basic Laplacian-based detector approximation. Moreover, in order to reduce the computation time drastically, our detector relies on integral images as defined by [Viola and Jones 2001].

2.1 Related Work

The most widely used detector probably is the Harris corner detector [Harris and Stephens 1988], proposed back in 1988. It is based on the eigenvalues of

the second moment matrix. However, Harris corners are not scale-invariant. [Lindeberg 1998] introduced the concept of automatic scale selection. This allows to detect interest points in an image, each with their own characteristic scale. He experimented with both the determinant of the Hessian matrix as well as the Laplacian (which corresponds to the trace of the Hessian matrix) to detect blob-like structures. [Mikolajczyk and Schmid 2001] refined this method, creating robust and scale-invariant feature detectors with high repeatability, which they coined Harris-Laplace and Hessian-Laplace. They used a (scale-adapted) Harris measure or the determinant of the Hessian matrix to select the location, and the Laplacian to select the scale. Focusing on speed, [Lowe 1999] proposed to approximate the Laplacian of Gaussians (LoG) by a Difference of Gaussians (DoG) filter.

Several other scale-invariant interest point detectors have been proposed. Examples are the salient region detector, proposed by [Kadir and Brady 2001], which maximises the entropy within the region, and the edge-based region detector proposed by [Jurie and Schmid 2004]. They seem less amenable to acceleration though. Also several affine-invariant feature detectors have been proposed that can cope with wider viewpoint changes. However, these fall outside the scope of this thesis.

From studying the existing detectors and from published comparisons [Mikolajczyk and Schmid 2004, Mikolajczyk and Schmid 2005], we can conclude that Hessian-based detectors are more stable and repeatable than their Harris-based counterparts. Moreover, using the determinant of the Hessian matrix rather than its trace (the Laplacian) seems advantageous, as it fires less on elongated, ill-localised structures. We also observed that approximations like the DoG can bring speed at a low cost in terms of lost accuracy.

2.2 Integral Images

In order to make the following chapters more self-contained, we briefly discuss the concept of integral images. They allow for fast computation of box type convolution filters. The entry of an integral image $I_\Sigma(\mathbf{x})$ at a location $\mathbf{x} = (x, y)^\top$ represents the sum of all pixels in the input image I within a rectangular region formed by the origin and $\mathbf{x}$.

$$I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,\, j). \tag{2.1}$$

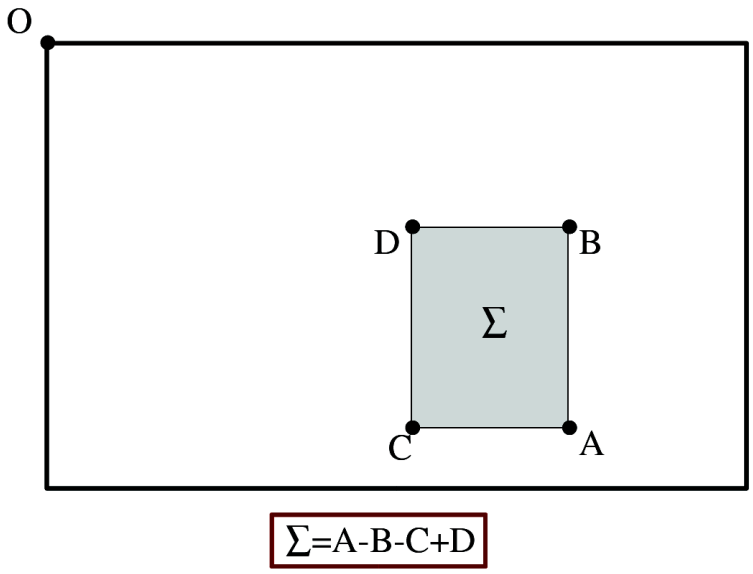

Figure 2.1: Using integral images, it takes only four operations to calculate the area of a rectangular region of any size.

Once the integral image has been computed, it takes four additions to calculate the sum of the intensities over any upright, rectangular area (see figure 2.1). Hence, the calculation time is independent of its size. This is important in our approach as we use big filter sizes.

2.3 Hessian Matrix Based Interest Points

We base our detector on the Hessian matrix because of its good performance in computation speed and accuracy. It detects blob-like structures at locations where its determinant is maximum. In contrast to the Hessian-Laplace detector by [Mikolajczyk and Schmid 2001], we rely on the determinant of the Hessian also for the scale selection, as in [Lindeberg 1998].

Given a point $\mathbf{x} = (x, y)$ in an image I, the Hessian matrix $\mathcal{H}(\mathbf{x}, \sigma)$ in $\mathbf{x}$ at scale σ is defined as follows

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}, \tag{2.2}$$

where $L_{xx}(\mathbf{x}, \sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image I in point $\mathbf{x}$, and similarly for $L_{xy}(\mathbf{x}, \sigma)$ and $L_{yy}(\mathbf{x}, \sigma)$.

Gaussians are optimal for scale-space analysis [Koenderink 1984, Lindeberg 1990], but in practice they have to be discretised and cropped (figure 2.2 left half). This leads to a loss in repeatability under image rotations around odd multiples of $\frac{\pi}{4}$. This weakness seems to hold for Hessian-based detectors in general. Figure 2.3 shows the repeatability rate of two detectors based on the Hessian matrix for pure image rotation. The repeatability attains a maximum around multiples of $\frac{\pi}{2}$. This is due to the square shape of the filter. Nevertheless, the detectors still perform well, and the slight decrease in performance does not outweigh the advantage of fast convolutions brought by the discretisation and cropping. As real filters are non-ideal in any case, and given David Lowe's success with his LoG approximations, we push the approximation for the Hessian matrix even further with box filters (in the right half of figure 2.2). These approximate second order Gaussian derivatives and can be evaluated at a very low computational cost using integral images. The calculation time therefore is independent of the filter size. As shown in the results section, the performance is comparable or better than with the discretised and cropped Gaussians.

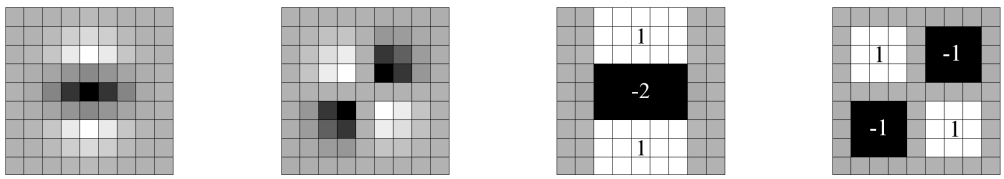

Figure 2.2: Left to right: the (discretised and cropped) Gaussian second order partial derivative in y- (L_{yy}) and xy-direction (L_{xy}), respectively; our approximation for the second order Gaussian partial derivative in y- (D_{yy}) and xy-direction (D_{xy}). The grey regions are equal to zero.

The 9×9 box filters in figure 2.2 are approximations of a Gaussian with $\sigma = 1.2$ and represent our lowest scale (i.e. highest spatial resolution). We will

denote them by D_{xx}, D_{yy}, and D_{xy}. The weights applied to the rectangular regions are kept simple for computational efficiency. The relative weights of the filter responses are further balanced in the expression for the Hessian's determinant. This is needed for the energy conservation between the Gaussian kernels and the approximated Gaussian kernels,

$$\frac{|L_{xy}(1.2)|_F|D_{yy}(9)|_F}{|L_{yy}(1.2)|_F|D_{xy}(9)|_F} = 0.912... \simeq 0.9, \tag{2.3}$$

where $|x|_F$ is the Frobenius norm. This yields

$$\boxed{\det(\mathcal{H}_{\text{approx}}) = D_{xx}D_{yy} - (0.9D_{xy})^2.} \tag{2.4}$$

Furthermore, the filter responses are normalised with respect to their size. This guarantees a constant Frobenius norm for any filter size. This is important for the scale space analysis as discussed in the next section.

The approximated determinant of the Hessian represents the blob response in the image at location $\mathbf{x}$. These responses are stored in a blob response map over different scales, and local maxima are detected as explained in section 2.5.

2.4 Scale Space Representation

Interest points need to be found at different scales, not least because the search of correspondences often requires their comparison in images where they are seen at different scales. Scale spaces are usually implemented as an image pyramid. The images are repeatedly smoothed with a Gaussian and subsequently sub-sampled in order to achieve a higher level of the Pyramid. David Lowe [Lowe 2004] subtracts these pyramid layers in order to get the DoG (Difference of Gaussians) images where edges and blobs can be found. Note that this sampling saves computation time, but it can lead to aliasing problems.

For the creation of this pyramid, the Gaussian kernel has been shown to be the optimal filter [Koenderink 1984]. In practice, however, the Gaussian needs to be discretised and cropped, and even with Gaussian filters, aliasing still occurs as soon as the resulting images are sub-sampled. Also, the fact that no new structures may appear while going to lower resolutions may have been proven in the 1D case, but is known to not apply to the relevant 2D case [Lindeberg

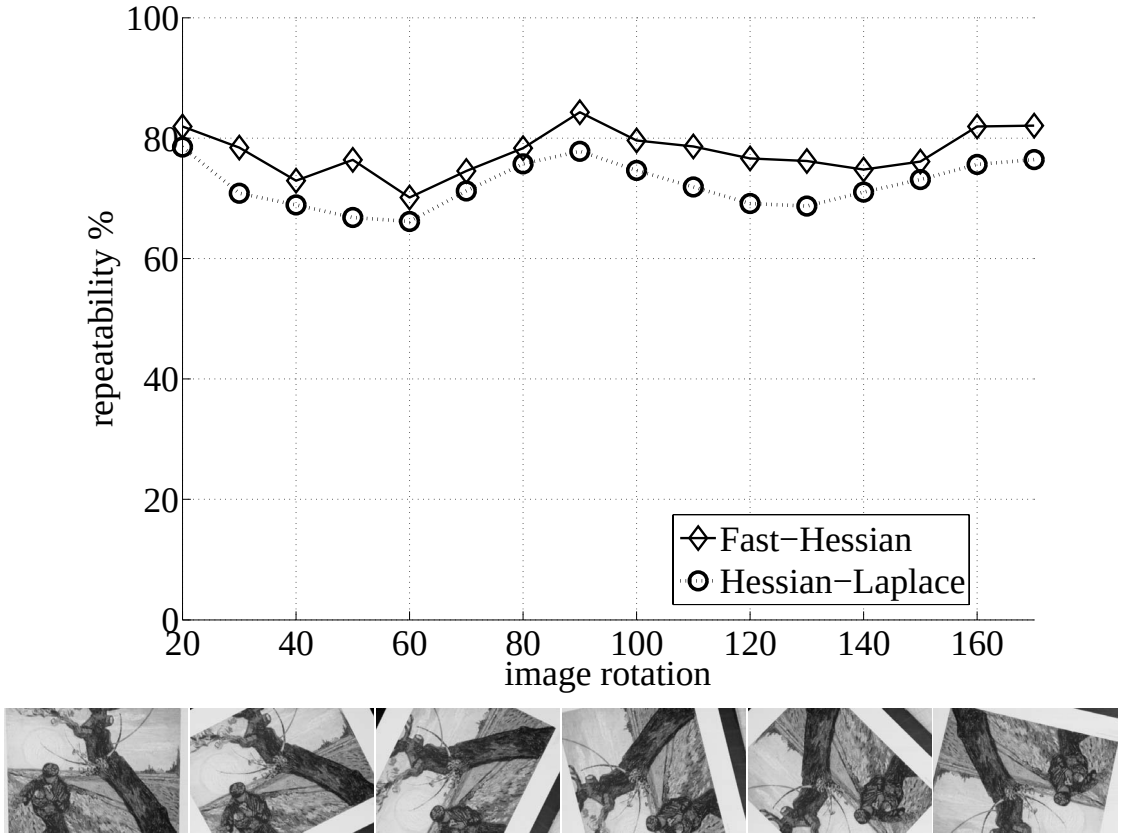

Figure 2.3: Top: Repeatability score for image rotation of 180 degrees. Hessian-based detectors have in general a lower repeatability score for angles around uneven multiples of $\frac{\pi}{4}$. Bottom: Sample images from the Van Gogh sequence that was used. Fast-Hessian is the more accurate version of our detector (FH-15), as explained in section 2.4.

1991]. Hence, the importance of the Gaussian seems to have been somewhat overrated in this regard, and here we test a simpler alternative.

Due to the use of box filters and integral images, as presented in the previous section, we do not have to iteratively apply the same filter to the output of a previously filtered layer, but instead can apply such filters of any size at exactly the same speed directly on the original image and even in parallel (although the latter is not exploited here). Therefore, the scale space is analysed by up-scaling the filter size rather than iteratively reducing the image size. The output of the 9×9 filter, introduced in the previous section, is considered as the initial scale layer, to which we will refer as scale $s = 1.2$ (corresponding to Gaussian derivatives with $\sigma = 1.2$). The following layers are obtained by filtering the

image with gradually bigger masks, taking into account the discrete nature of integral images and the specific structure of our filters.

The scale space is divided into octaves. An octave represents a series of filter response maps obtained by convolving the same input image with a filter of increasing size. In total, an octave includes a scaling factor of 2 (more than doubling the filter size). Each octave is subdivided into a constant number of scale levels. Due to the discrete nature of integral images, the maximum number of sub-divisions of the octaves depends on the initial length l_0 of the positive or negative lobes of the partial second order derivative in the direction of derivation (x or y). For the 9×9 filter, this length l_0 is 3. For two successive levels, we can increase this size by a minimum of 2 pixels (one pixel on every side) in order to keep the size uneven, hence with a total increase of the mask size with 6 pixels (see figure 2.4).

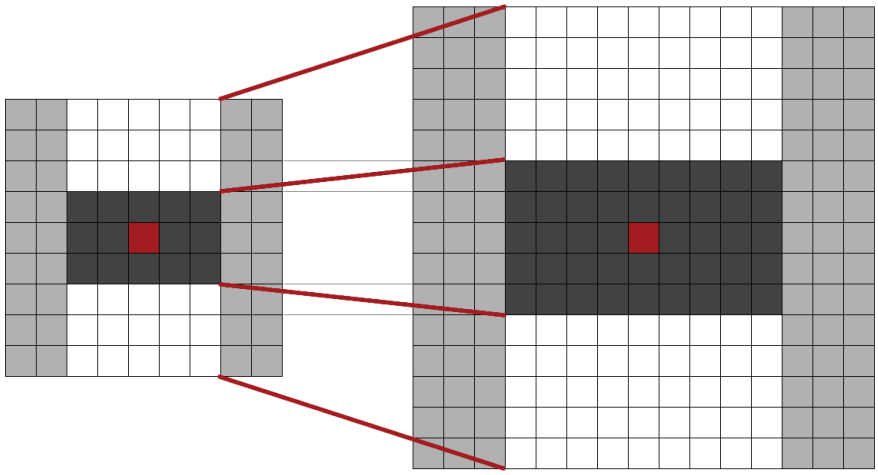

Figure 2.4: Filters D_{yy} for two successive scale levels (9×9 and 15×15). The length of the dark lobe can only be increased by an even number of pixels in order to guarantee the presence of a central pixel.

As mentioned above, the construction of the scale space starts with the 9×9 filters of the previous section, and calculates the blob response of the image for the first octave. Then, filters with sizes 15, 21, and 27 are applied, by which even more than a scale change of 2 has been achieved. But this is needed, as a non-maximum suppression is applied in a 3D space neighbourhood of the pixels, i.e. it is applied both spatialy and over the neighbouring scales (see figure 2.6). Hence, the first and last Hessian response maps in the stack cannot contain such maxima themselves, as they are used for reasons of comparison

only. Furthermore, the positions of the maxima are interpolated by fitting a paraboloid over space and scale. For the scale, the result of this interpolation (explained in section 2.5) can be shown to be no closer than halfway between the middle image and the lowest or highest level of the stack of 3 subsequently scaled images over which such interpolations are carried out (see appendix B). As an example, 3 such subsequent images could have been filtered with filters of size 9, 15, and 21. Halfway between 9 and 15 would correspond to size 12. As the 9×9 filter is the smallest we can reasonably use, 12 is the filter size for the finest scale at which a maximum can be found through interpolation. As mentioned above, the 9×9 filter corresponds to a Gaussian derivative with $\sigma_9 = 1.2$. Thus, the standard deviation and the current scale s for the filter of size 12 is $\sigma_{12} = 1.6 = s$. Another such stack of 3 subsequent images for the first octave is formed by the 15×15, 21×21, and 27×27 filtered images. These represent the crudest scales in the construction of the octave. The crudest possible scale that can actually be found through interpolation then corresponds to halfway between the 21×21 and 27×27 filtered images, i.e. at 24×24 ($s = \sigma_{24} = 3.2$). Hence, between the finest and crudest scale, an effective scale change with a factor of 2 has been achieved (i.e. from $s = 1.6$ to $s = 3.2$), hereby perfectly spanning one octave.

Similar considerations hold for the other octaves. For each new octave, the filter size increase is doubled (going from 6 to 12 to 24 to 48). Simultaneously, the sampling intervals for the extraction of the interest points can be doubled as well for every new octave. This reduces the computation time and the loss in accuracy is comparable to the image sub-sampling of the traditional approaches. The filter sizes for the second octave are 15, 27, 39, 51. A third octave is computed with the filter sizes 27, 51, 75, 99 and, if the original image size is still larger than the corresponding filter sizes, the scale space analysis is performed for a fourth octave, using the filter sizes 51, 99, 147, and 195. Note that more octaves may be analysed, but the number of detected interest points per octave decreases very quickly (see figure 2.5).

The large scale changes, especially between the first filters within these octaves (from 9 to 15 is a change of 1.7), renders the sampling of scales quite crude. Therefore, we have also implemented a scale space with a finer sampling of the scales. This first doubles the size of the image, using linear interpolation, and then starts the first octave by filtering with a filter of size 15. Additional filter sizes are 21, 27, 33, and 39. Then a second octave starts, again using filters which now increase their sizes by 12 pixels, after which a third and fourth

octave follow. Now the scale change between the first two filters is only 1.4 $(21/15)$. The lowest scale for the accurate version that can be detected through quadratic interpolation is $s = (1.2\frac{15}{9})/2 = 1$.

In scale space analysis, the question of scale normalisation arises. Scale normalisation, also called γ-normalisation, is necessary in order to prevent the reduction of Gaussian filter response at higher scales. Tony Lindeberg introduced the L_p-norms [Lindeberg and Bretzner 2003] that have to be constant over scales for a perfectly scale invariant filter. The L_p-norms are defined as follows.

$$\| g_{\xi m}(\cdot;\, t) \|_p = \left(\int_{x \in \Re^D} | g_{\xi m}(\cdot;\, t) |^p \, dx \right)^{1/p}, \tag{2.5}$$

where $g_{\xi m}$ is a the γ-normalised Gaussian operator, m the order of the derivative (in our case $m = 2$). D denotes the dimension of the signal (in our case 2). t is the scale and p is related to γ. Specifically, $\gamma = 1$ corresponds to $p = 1$ and thus to perfect scale invariance of the filter response. As the Frobenius norm remains constant for our filters at any size, it follows that they are already scale normalised, and no further weighting of the filter response is required.

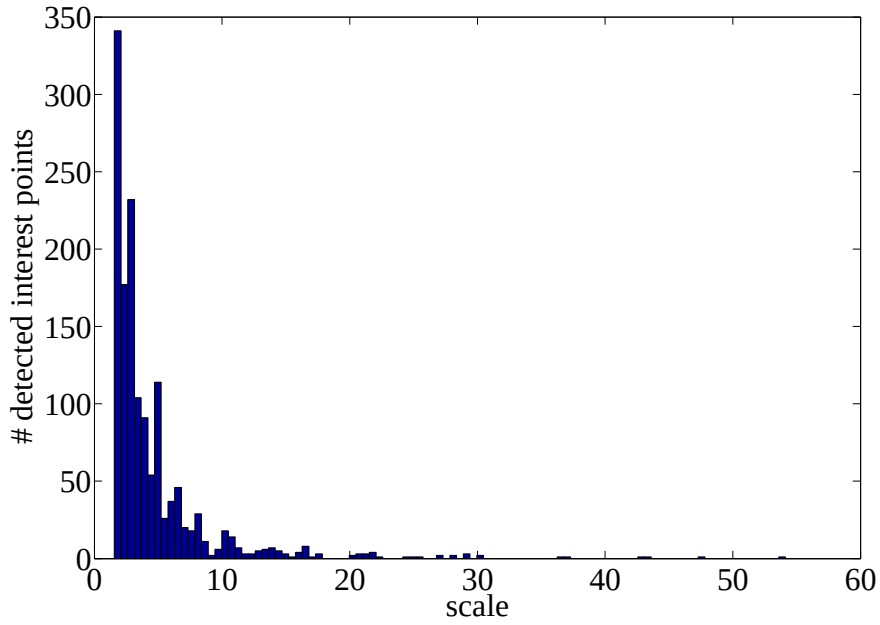

Figure 2.5: Histogram of the detected scales.

2.5 Interest Point Localisation

In order to localise interest points in the image and over scales, a non-maximum suppression in a $3 \times 3 \times 3$ neighbourhood (see figure 2.6) is applied. For that task, we use a fast non-maximum suppression algorithm explained in appendix A. The maxima of the determinant of the Hessian matrix are then interpolated in scale and image space with the method proposed by Matthew Brown *et al.* [Brown and Lowe 2002].

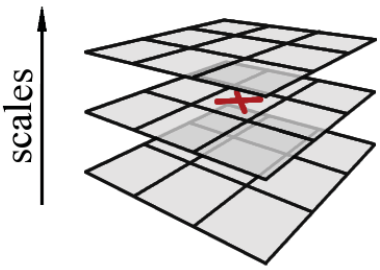

Figure 2.6: $3 \times 3 \times 3$ neighbourhood in image and scale space considered for the non-maximum suppression. If the intensity of the central pixel (marked with a cross) is higher than the intensities of its surrounding pixels, it is considered as a local maximum.

In order to find the interpolated location of the interest point, we take the blob responses of the same $3 \times 3 \times 3$ neighbourhood (denoted B) in each dimension around the detected maximum as described above. We then locate the maxima to sub-pixel/sub-scale accuracy by fitting a 3D quadratic to the scale-space blob-response map.

$$B(\mathbf{x}) = B + \left(\frac{\partial B}{\partial \mathbf{x}}\right)^{\top} \mathbf{x} + \frac{1}{2}\mathbf{x}^{\top}\frac{\partial^2 B}{\partial \mathbf{x}^2}\mathbf{x}, \tag{2.6}$$

where $\mathbf{x} = (x, y, s)^{\top}$ is the scale-space coordinate and $B(\mathbf{x})$ is the blob response (determinant of the approximated Hessian matrix) at the location $\mathbf{x}$. The quadratic coefficients are computed by approximating the derivatives using finite differences of the neighbouring samples. The sub-pixel/sub-scale interest point location $\hat{\mathbf{x}}$ is the extremum of this 3D quadratic.

$$\boxed{\hat{\mathbf{x}} = -\left(\frac{\partial^2 B}{\partial \mathbf{x}^2}\right)^{-1} \frac{\partial B}{\partial \mathbf{x}}.} \tag{2.7}$$

Scale space interpolation is especially important in our case, as the difference in scale between the first layers of every octave is relatively large. Figure 2.7 (left) shows an example of the detected interest points using our 'Fast-Hessian' detector.

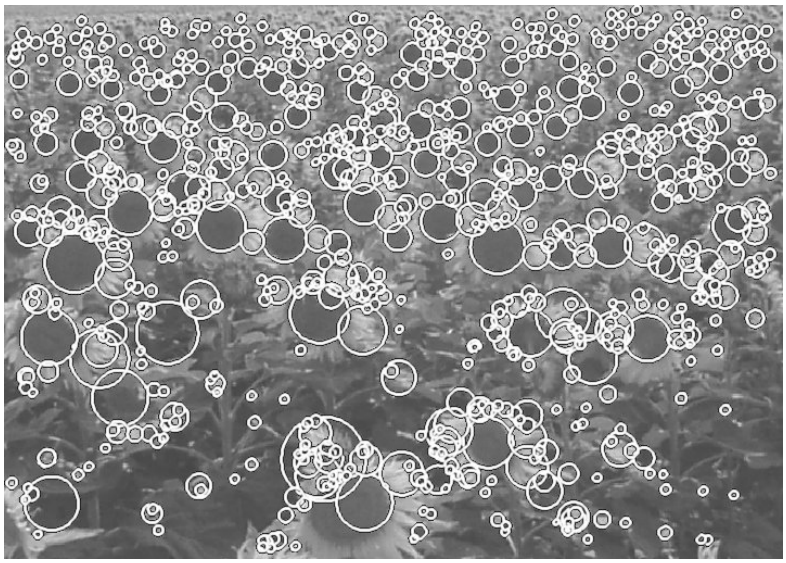

Figure 2.7: Detected interest points for a Sunflower field. This kind of scenes shows clearly the nature of the features from Hessian-based detectors

2.6 Results

We tested our detector using the image sequences (see appendix C) and testing software provided by Krystian Mikolajczyk[1]. The evaluation criteria is the *repeatability score*. The timings were measured on a standart PC Pentium IV, running at 3 GHz.

Repeatability Score The repeatability score is the average number of corresponding interest points in images under different geometric and photometric

[1] http://www.robots.ox.ac.uk/~vgg/research/affine/

transformations, in relative terms. It represents the percentage of geometrically correct interest point correspondences over all interest points in the common visible part in two images. Geometrically correct means that an interest point is found at the same physical position. In general, the repeatability score is calculated over multiple images using a reference image. With every image, the transformation becomes more important with respect to the reference image. The sequences tested here use six images, where the first image is used as the reference. The imaged scenes can be related with a *homography*, which is known beforehand and describes the transformation from the reference image to a given image in the sequence. A homography is a projective mapping from one plane to another. For more details about homographies, the reader is referred to [Hartley and Zisserman 2004].

The problem consists in judging whether the interest points correspond or not. Due to the different scales of the interest points, it would not make sense to compare the distance between given interest points and its geometrically transformed counterpart from the reference image [Schmid *et al.* 2000]. The correspondence measure has to be the same for all scales. Therefore, we use the same criteria as in [Mikolajczyk *et al.* 2005], which is based on the overlap of, in our case, a scale dependent circular region centred in the interest point. It is obvious that for homographies corresponding to projective transformations, such regions may become elliptic (see figure 2.8). Nevertheless, the overlap criteria is still representative, as the respective scale changes accordingly from one image to the other of the same physical interest point. More precisely, we observed that if a circular region is squeezed to an ellipse by the transformation from one image to the other, the corresponding interest point in the second image is typically detected at a lower scale and the overlap is therefore still sufficient (see figure 2.8). Similar properties apply for the case when a circular region is stretched to an ellipse.

A candidate interest point in one image is considered a correspondence to a given interest point in the reference image, if its transformed circular region generates an area overlap of at least 60% with the circular region of the interest point in the reference view. For more information we refer the reader to [Mikolajczyk *et al.* 2005].

Experimental Results The test sequences are images of real textured and structured scenes. There are different types of geometric and photometric transformations, like affine transformation, zoom and rotation, image blur,

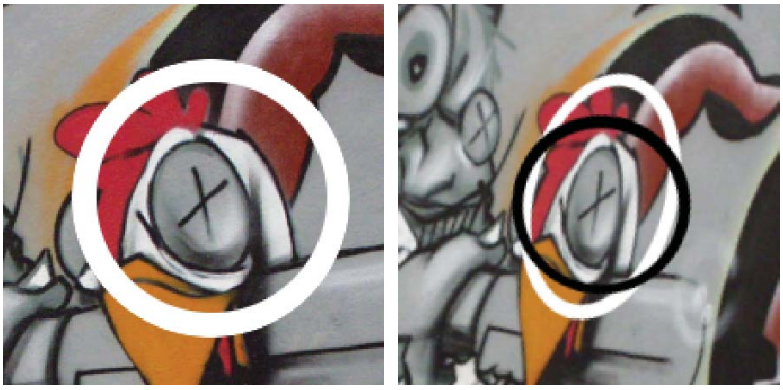

Figure 2.8: Left: Circular region for the detected interest point in the reference view. Right: Transformed circular region from the reference view (white) and circular region for the detected counterpart (black).

detector	threshold	nb of points	comp. time (ms)
FH-15	900	1493	210
FH-9	600	1418	120
Hessian-Laplace	1000	1979	650
Harris-Laplace	2500	1664	1800
DoG	default	1520	400

Table 2.1: Thresholds, number of detected points and calculation time for the detectors in our comparison. (First image of Graffiti scene, 800×640)

lighting changes and JPEG compression. The test images are shown in appendix C.

We tested two versions of our *Fast-Hessian* detector depending on the initial Gaussian derivative filter size. *FH-9* stands for our Fast Hessian detector with the initial filter size 9×9, and *FH-15* is the 15×15 filter on the double input image size version. Throughout this chapter, including the object recognition experiment (see section 4.1), we always use the same thresholds.

The detector is compared to the Difference of Gaussians (DoG) detector by [Lowe 2004], and the Harris- and Hessian-Laplace detectors proposed by [Mikolajczyk and Schmid 2004]. The number of interest points found is on average very similar for all detectors. This holds for all images, including those from the databases used in the object recognition experiment, (see table 2.1 for an example). The thresholds were adapted according to the number of interest

points found with the DoG detector kindly provided by David Lowe. Note that the original implementation of SIFT uses the double image size in order to detect many DoG features. Furthermore, it may detect more than one interest point for a same interest point location. This is the case if a detected orientation is not clearly dominating. Hence, multiple interest points are constructed each with a different orientation corresponding to the peaks in the orientation histogram which are higher than a certain threshold (see [Lowe 2004]). In order to have equal conditions for each interest point detector, we disabled these additional options which would also result in slower detection times for the speed comparisons. However, the *FH-9* detector is about 3 times as fast as DoG and 6 times faster than Hessian-Laplace. The *FH-15* detector, is almost twice as fast as DoG and almost three times as fast as Hessian-Laplace. At the same time, the repeatability scores for our detectors are comparable or even better than for the competitors.

The repeatability scores for the Graffiti sequence (figure 2.9 top) are comparable for all detectors. The repeatability score of the *FH-15* detector for the Wall sequence (figure 2.9 bottom) outperforms the competitors. Note that the sequences Graffiti and Wall contain out-of-plane rotation, resulting in affine deformations, while the detectors in the comparison are only invariant to image rotation and scale. Hence, these deformations have to be tackled by the overall robustness of the features. In the Boat sequence (figure 2.10 top), the *FH-15* detector shows again a better performance than the others. The *FH-9* and FH-15 detectors are outperforming the others in the Bikes sequence (figure 2.11 top). For all other sequences (figure 2.10–2.12), our detectors perform comparably well than the competitors.

2.7 Conclusion and Outlook

This chapter presented a fast multi-scale interest point detector. The important speed gain is due to the use of integral images, which drastically reduce the number of operations for simple box convolutions. We based our detector on the Hessian matrix as it performed well in former comparative studies. The results showed that the performance of our Hessian approximation is comparable and sometimes even better than the state-of-the-art interest point detectors. The high repeatability is advantageous for camera self-calibration, where an accurate interest point detection has a direct impact on the accuracy of the

camera self-calibration and therefore on the quality of the resulting 3D model (see chapter 6).

The most important improvement, however, is the speed of the detector. An almost real-time computation without loss in performance represents an important advantage for many on-line computer vision applications.

The most interesting enhancement to the interest point detector would be to upgrade it in order to be affine invariant. A comparison of the repeatability score for the Graffiti sequence with the results achieved by Krystian Mikolajczyk in his comparative study on affine invariant regions [Mikolajczyk *et al.* 2005] shows clearly the possible improvement in performance under wide-baseline viewpoint changes.

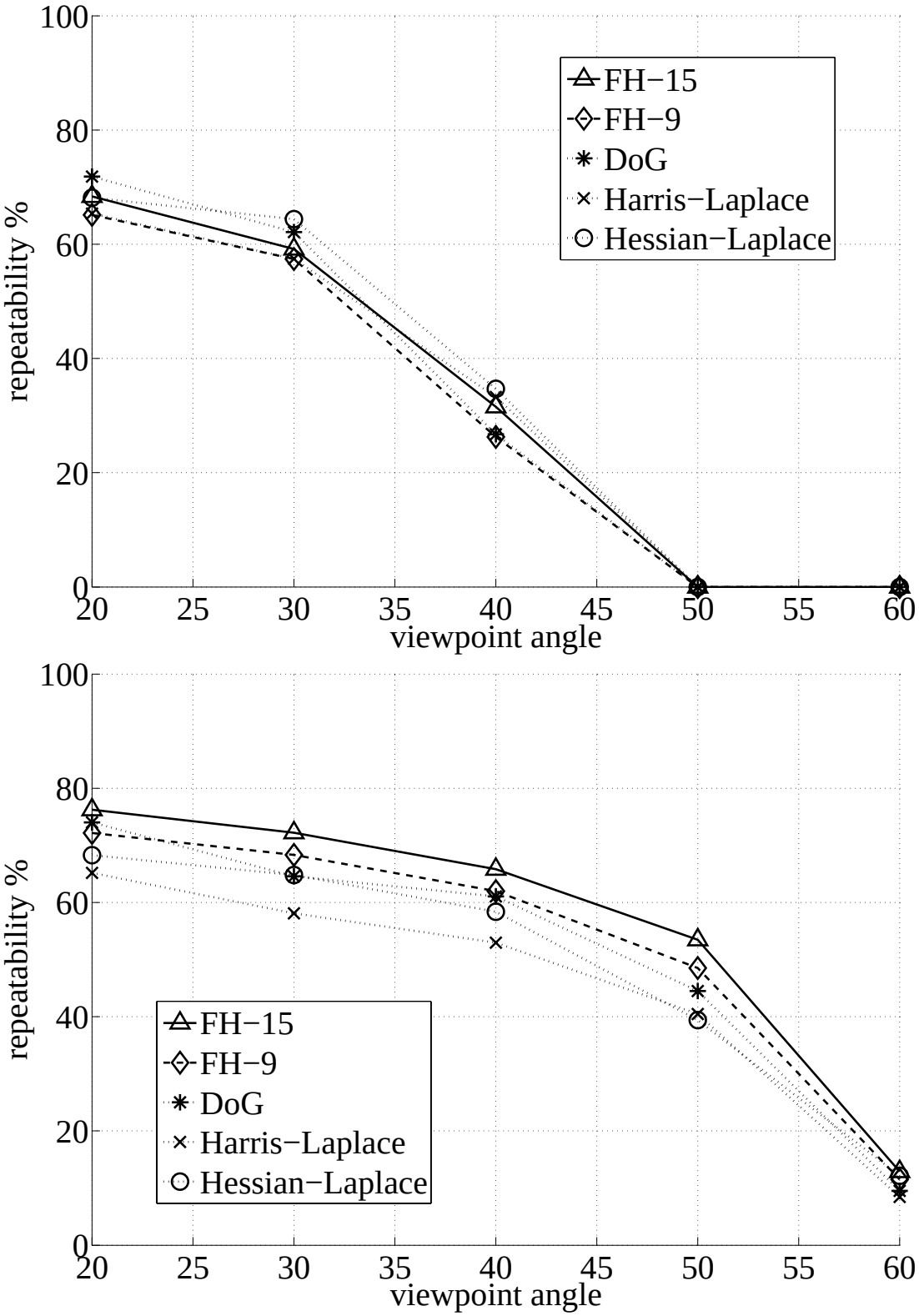

Figure 2.9: Repeatability score for the Graffiti (top) and Wall (bottom) sequence (Viewpoint Change).

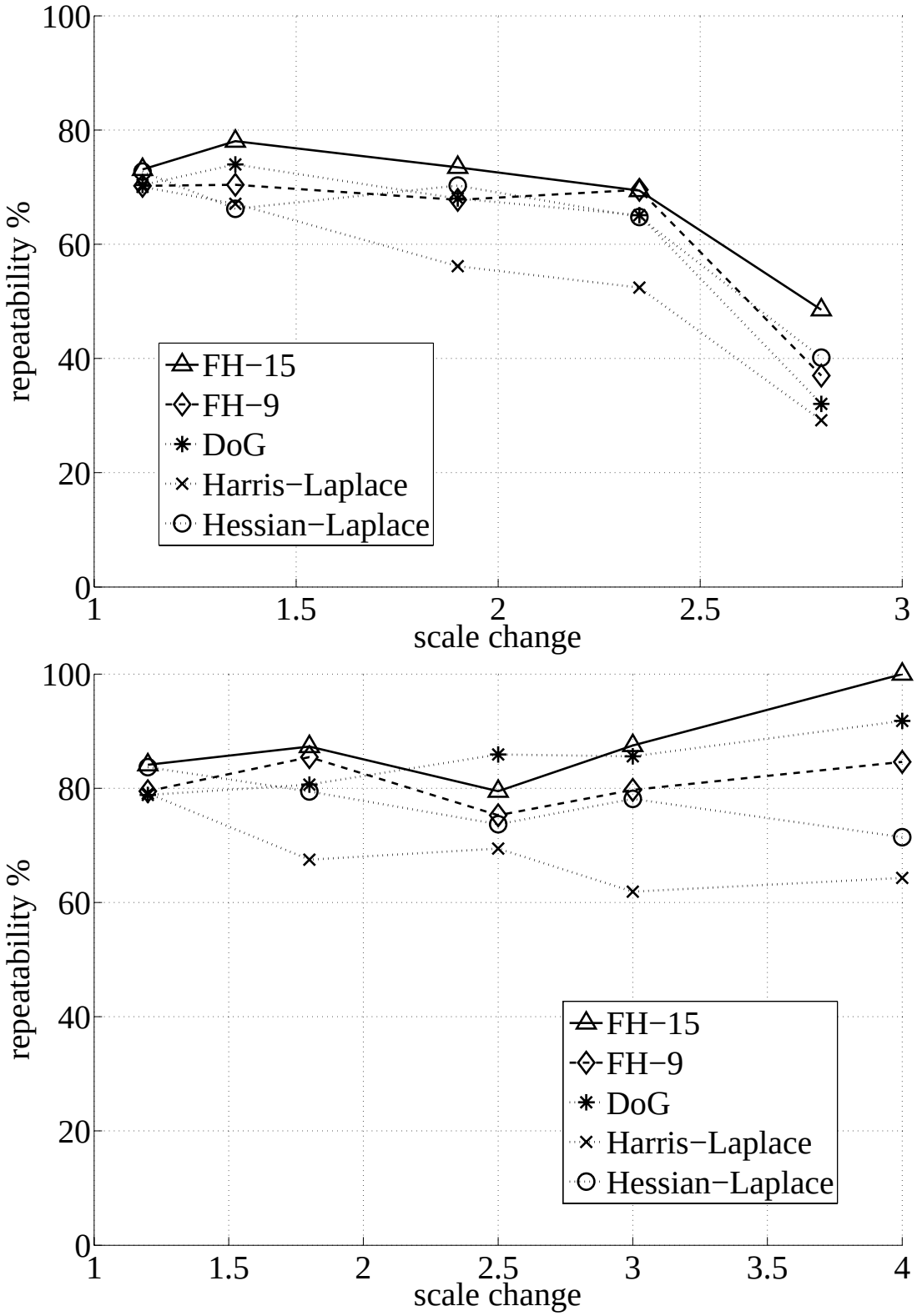

Figure 2.10: Repeatability score for the Boat (top) and Bark (bottom) sequence (Zoom and Rotation).

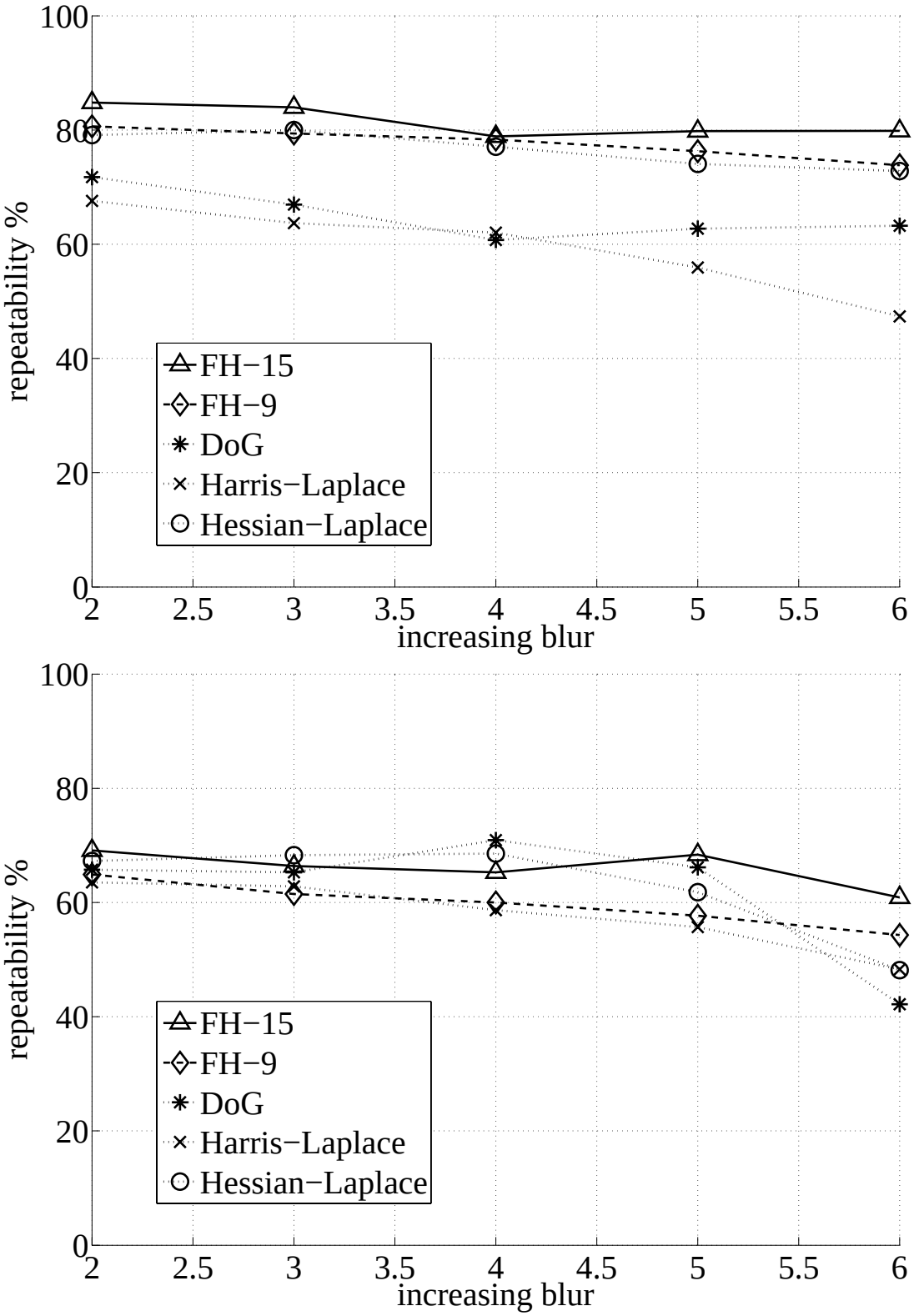

Figure 2.11: Repeatability score for the Bikes (top) and Trees (bottom) sequence (Image Blur).

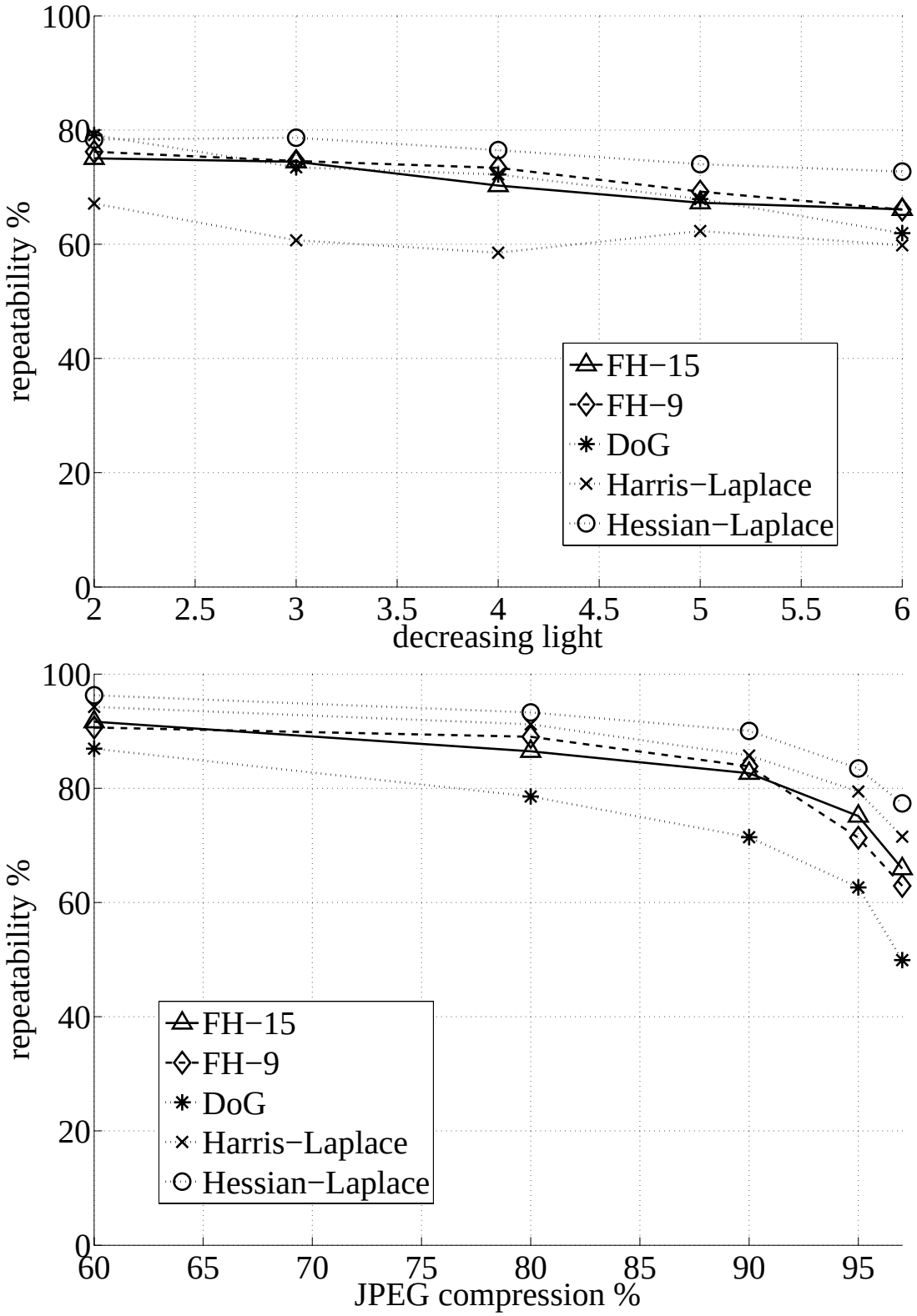

Figure 2.12: Top: repeatability score for the Leuven (Lighting Change), and bottom: for the Ubc (JPEG Compression) sequence.

3

Interest Point Description and Matching

In the previous chapter, we proposed a fast interest point detector, approximating the Hessian matrix with very basic box-like filters. In order to get image correspondences, these interest points have to be robustly characterised by a descriptor in a first step. Then, the distances of the point pairs in descriptor space are computed. In general, this is done for all possible interest point combinations. More efficient matching techniques like the best-bin-first method proposed by David Lowe [Lowe 2004] use rapid indexing in order to compute the descriptor distances only for the most probable matching candidates. However, such techniques reduce the quality of the resulting matches (more mismatches).

Our descriptor describes the distribution of the intensity content within the interest point neighbourhood, a bit similar to the gradient information extracted by SIFT and its variants. We work on the distribution of first order Haar wavelet responses in x and y direction rather than the gradient, exploit integral images for speed, and only use 64 dimensions. This fact reduces the time for feature computation and matching, and has proven to simultaneously increase the robustness. Furthermore, we present a new indexing step based on the sign of the Laplacian, which increases not only the robustness of the descriptor, but also the matching speed (factor of two in the best case). We refer to our detector-descriptor scheme as SURF – Speeded Up Robust Features.

The good performance of SIFT compared to other descriptors [Mikolajczyk and Schmid 2005] is remarkable. Its mix of crude localisation information

and the distribution of gradient related features seems to yield good distinctive power while fending off the effects of localisation errors in terms of scale or space. Using relative strengths and orientations of gradients reduces the sensitivity to photometric changes.

The proposed SURF descriptor is based on features with a complexity stripped down even further. The first step consists of fixing a reproducible orientation based on information from a circular region around the interest point. Then, we construct a square region aligned to the selected orientation, extract the SURF descriptor from it, and finally search for correspondences in a second image. These three steps are explained in this chapter, but first we describe the state-of-the-art of local feature descriptors.

3.1 Related Work

A large variety of feature descriptors has been proposed, like Gaussian derivatives [Florack *et al.* 1994], moment invariants [Mindru *et al.* 2004], complex features [Baumberg 2000, Schaffalitzky and Zisserman 2002], steerable filters [Freeman and Adelson 1991], phase-based local features [Carneiro and Jepson 2003], and descriptors representing the distribution of smaller-scale features within the interest point neighbourhood. The latter, introduced by Lowe [Lowe 2004], have been shown to outperform the others [Mikolajczyk and Schmid 2003]. This can be explained by the fact that they capture a substantial amount of information about the spatial intensity patterns, while at the same time being robust to small deformations or localisation errors. The descriptor in [Lowe 2004], called SIFT for short, computes a histogram of local oriented gradients around the interest point and stores the bins in a 128-dimensional vector (8 orientation bins for each of 4×4 location bins).

Various refinements on this basic scheme have been proposed. Ke and Sukthankar [Ke and Sukthankar 2004] applied PCA on the gradient image around the detected interest point. This PCA-SIFT yields a 36-dimensional descriptor which is fast for matching, but proved to be less distinctive than SIFT in a second comparative study by [Mikolajczyk and Schmid 2005] and slower feature computation reduces the effect of fast matching. In the same paper [Mikolajczyk and Schmid 2005], the authors proposed a variant of SIFT, called GLOH, which proved to be even more distinctive with the same number of dimensions.

However, GLOH is computationally more expensive as it uses again PCA for data compression.

The SIFT descriptor still seems the most appealing descriptor for practical uses, and hence also the most widely used nowadays. It is distinctive *and* relatively fast, which is crucial for on-line applications. Recently, [Se *et al.* 2004] implemented SIFT on a Field Programmable Gate Array (FPGA) and improved its speed by an order of magnitude. However, the high dimensionality of the descriptor is a drawback of SIFT at the matching step. For on-line applications relying only on a regular PC, each one of the three steps (detection, description, matching) had to be fast. David Lowe proposed a best-bin-first alternative [Lowe 2004] in order to speed up the matching step, but this results in lower accuracy of the matches.

3.2 Orientation Assignment

In order to be invariant to image rotation, we identify a reproducible orientation for the interest points. For that purpose, we first calculate the Haar wavelet responses in x and y direction within a circular neighbourhood of radius $6s$ around the interest point, with s the scale at which the interest point was detected (see previous chapter). Also, the sampling step is scale dependent and chosen to be s. In keeping with the rest, also the wavelet responses are computed at that current scale s. Accordingly, at high scales the size of the wavelets is big. Therefore, we use again integral images for fast filtering. The used filters are shown in figure 3.1. Only six operations are needed to compute the response in x or y direction at any scale. The side length of the wavelets is $4s$.

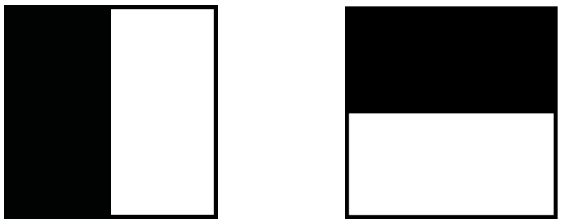

Figure 3.1: Haar wavelet filters to compute the responses in x (left) and y direction (right). The dark parts have the weight -1 and the light part $+1$.

Once the wavelet responses are calculated and weighted with a Gaussian ($\sigma = 2s$) centred at the interest point (see figure 3.2 left), the responses are rep-

resented as points in a space with the horizontal response strength along the abscissa and the vertical response strength along the ordinate. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of size $\frac{\pi}{3}$, see figure 3.2. The horizontal responses within the window are summed, and also the vertical responses. The two, summed responses then yield a local orientation vector. The longest such vector lends its orientation to the interest point. The size of the sliding window is a parameter which had to be carefully chosen. Small sizes fire on single dominating gradients, large sizes tend to yield maxima in vector length that are not outspoken. Both result in misorientation of the interest region.

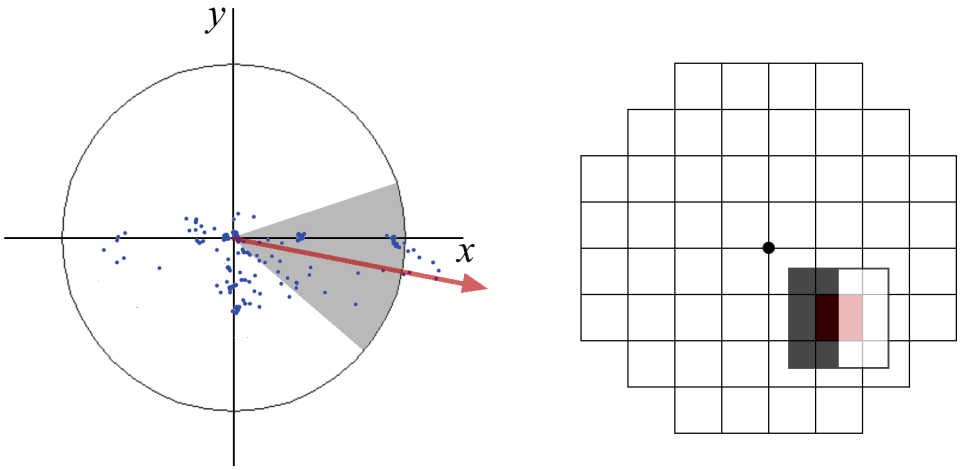

Figure 3.2: Orientation assignment: A sliding orientation window of size $\frac{\pi}{3}$ (left) detects the dominant orientation of the Gaussian weighted Haar wavelet responses at every sample point within a circular neighbourhood around the interest point (right).

Note that the interest point location is a floating point number, but the integral image is not interpolated. Interpolation curiously didn't increase the repeatability of the orientation assignment. The effect of the geometrical transformations between the images seems to outweigh the error due to the absence of interpolation.

3.3 Sum Of Haar Wavelet Responses Descriptor

For the extraction of the descriptor, the first step consists of constructing a square region centred around the interest point and oriented along the orienta-

tion selected in the previous section. For the upright version, this transformation is not necessary. The size of this window is $20s$. Examples of such square regions are illustrated in figure 3.3.

Figure 3.3: Detail of the Graffiti scene showing the size of the descriptor window at different scales.

The region is split up regularly into smaller 4×4 square sub-regions. This keeps important spatial information in. For each sub-region, we compute a few simple features at 5×5 regularly spaced sample points. For reasons of simplicity, we call d_x the Haar wavelet response in horizontal direction and d_y the Haar wavelet response in vertical direction (filter size $2s$), see figure 3.1 again. "Horizontal" and "vertical" here is defined in relation to the selected interest point orientation (see figure 3.4). To increase the robustness towards geometric deformations and localisation errors, the responses d_x and d_y are first weighted with a Gaussian ($\sigma = 3.3s$) centred at the interest point.

Then, the wavelet responses d_x and d_y are summed up over each sub-region and form a first set of entries to the feature vector. In order to bring in information about the polarity of the intensity changes, we also extract the sum of the absolute values of the responses, $|d_x|$ and $|d_y|$. Hence, each sub-region has a four-dimensional descriptor vector $\mathbf{v}$ for its underlying intensity structure

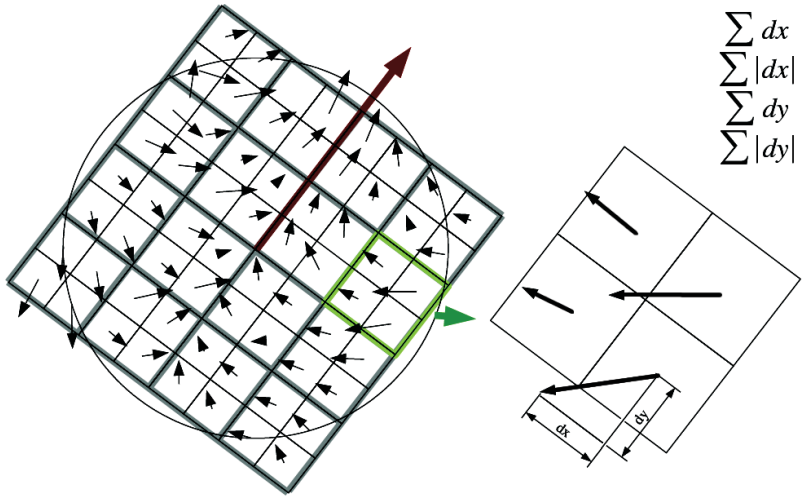

Figure 3.4: To build the descriptor, a quadratic grid with 4×4 square sub-regions is laid over the interest point (left). For each sample, the wavelet responses are computed. For this figure 2×2 vectors per sub-region for reasons of illustration. For each sub-region (right), the sums of dx, $|dx|$, dy, and $|dy|$ relative to the orientation of the grid, are computed.

$\mathbf{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. This results in a descriptor vector for all 4×4 sub-regions of length 64. The wavelet responses are invariant to a bias in illumination (offset). Invariance to contrast (a scale factor) is achieved by turning the descriptor into a unit vector.

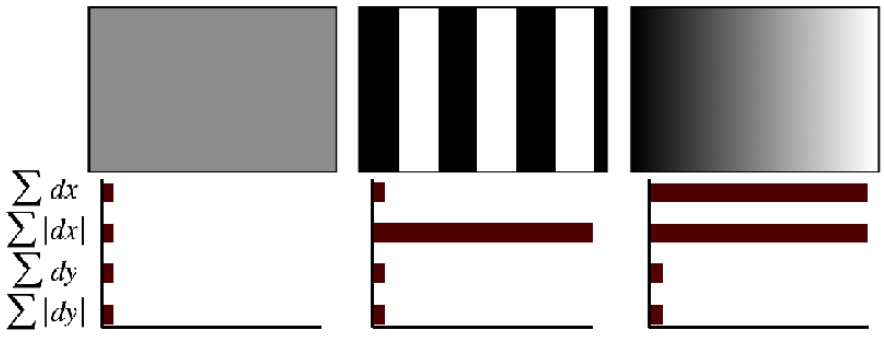

Figure 3.5: The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of a homogeneous region, all values are relatively low. Middle: In presence of frequencies in x direction, the value of $\sum |d_x|$ is high, but all others remain low. If the intensity is gradually increasing in x direction, both values $\sum d_x$ and $\sum |d_x|$ are high.

Figure 3.5 shows the properties of the descriptor for three distinctively different image intensity patterns within a sub-region. One can imagine combinations of such local intensity patterns, resulting in a distinctive descriptor.

In order to arrive at these SURF descriptors, we experimented with fewer and more wavelet features, using d_x^2 and d_y^2, higher-order wavelets, PCA, median values, average values, etc. From a thorough evaluation, the proposed sets turned out to perform best. We then varied the number of sample points and sub-regions. The 4×4 sub-region division solution provided the best results. Considering finer subdivisions appeared to be less robust and would increase matching times too much. On the other hand, the short descriptor with 3×3 sub-regions (SURF-36) performs worse, but allows for very fast matching and is still quite acceptable in comparison to other descriptors in the literature. Figure 3.6 shows only a few of these comparison results (SURF-128 will be explained shortly).

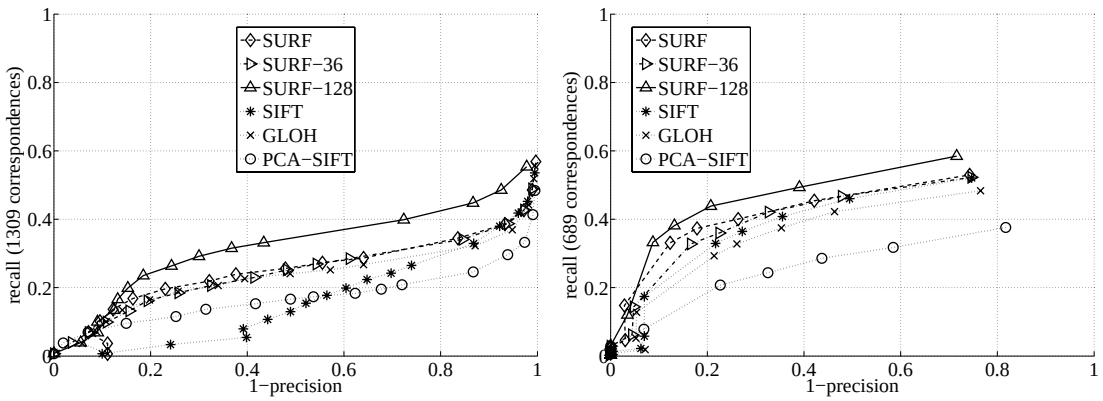

Figure 3.6: The *recall* vs. *(1-precision)* graph (discussed in section 3.5) for different binning methods and two different matching strategies tested on the 'Graffiti' sequence (image 1 and 3) with a view change of 30 degrees, compared to the current descriptors. The interest points are computed with our 'Fast Hessian' detector. Note that the interest points are not affine invariant. The results are therefore not comparable to the ones in [Mikolajczyk and Schmid 2005]. SURF-128 corresponds to the extended descriptor (explained below). Left: Similarity-threshold-based matching strategy. Right: Nearest-neighbour-ratio matching strategy (See section 3.5).

We also tested an alternative version of the SURF descriptor that adds a couple of similar features (SURF-128). It again uses the same sums as before, but now splits these values up further. The sums of d_x and $|d_x|$ are computed separately for $d_y < 0$ and $d_y \geq 0$. Similarly, the sums of d_y and $|d_y|$ are split up according to the sign of d_x, thereby doubling the number of features. The descriptor is

more distinctive and not much slower to compute, but slower to match due to its higher dimensionality.

In Figure 3.6, the parameter choices are compared for the standard 'Graffiti' scene, which is the most challenging of all the scenes in the evaluation set of Mikolajczyk [Mikolajczyk and Schmid 2005], as it contains out-of-plane rotation, in-plane rotation as well as brightness changes. The extended descriptor for 4×4 sub-regions (SURF-128) comes out to perform best. Also, SURF performs well and is faster to handle. Both outperform the existing state-of-the-art. On a Pentium IV at 3 GHz, the SURF-64 descriptor is computed in 0.2 ms per interst point.

3.4 Fast Indexing For Matching

For fast indexing during the matching stage, the sign of the Laplacian (i.e. the trace of the Hessian matrix) for the underlying interest point is included. Typically, the interest points are found at blob type structures. The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation. This feature is available at no extra computational cost as it was already computed during the detection phase. In the matching stage, we only compare features if they have the same type of contrast, see figure 3.7. Hence, this minimal information allows for faster matching, without reducing the descriptor's performance.

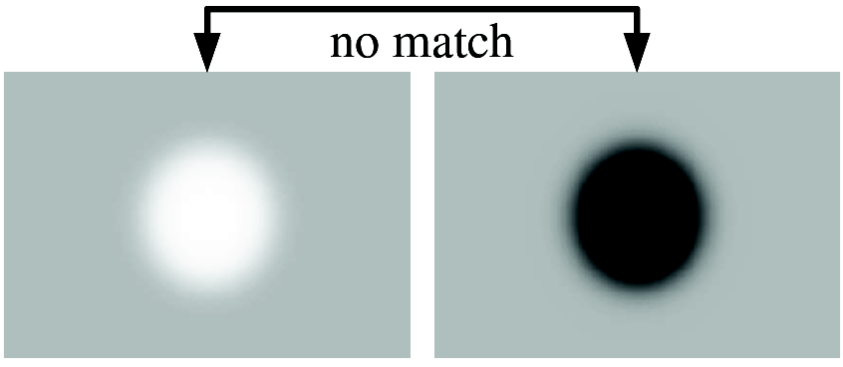

Figure 3.7: If the contrast between two interest points is different (dark on light background vs. light on dark background), the candidate is not considered a valuable match.

3.5 Results

For evaluating the descriptor, we used the *recall-precision* evaluation criterion i.e. the number of correct and false correspondences between two images, as in [Ke and Sukthankar 2004] and [Mikolajczyk and Schmid 2005]. The results are presented with *recall* versus *1-precision* plots. The values for recall depend on the number of correspondences as follows:

$$\text{recall} = \frac{\text{correct matches}}{\text{correspondences}} \tag{3.1}$$

The number of false matches relative to the total number of matches is represented by $1 - \text{precision}$.

$$1 - \text{precision} = \frac{\text{false matches}}{\text{correct matches} + \text{false matches}} \tag{3.2}$$

The matching is performed using the similarity-threshold-based matching strategy. That means that two interest points are matched if their Euclidean distance in descriptor space is smaller than a threshold. As a consequence, an interest point can have more than one correct match. An alternative matching strategy would be based on the nearest neighbour of a given interest point. For the nearest-neighbour-ratio matching, two interest points are matched if the ratio between their distance and the distance to the second nearest neighbour is smaller than a threshold. This approach generates only one correct match for a given interest point. The precision is typically higher for the nearest-neighbour-based approaches than for the threshold-based matching strategy. However, the latter represents well the distribution of the different methods in descriptor space. Therefore, we chose the threshold-based approach.

The ground truth (number of correspondences) is evaluated similar as in chapter 2, but this time using 50% of overlap error for the respective circular regions.

We evaluated the descriptor on the same image sequences as used for evaluating the detector (see chapter 2 and appendix C). Each time, we used the first and the fourth image of the sequence, except for the Graffiti (image 1 and 3) and the Wall scene (image 1 and 5), corresponding to a viewpoint change of 30

and 50 degrees, respectively. In figures 3.6 and 3.8-3.10, we compared our descriptors (SURF 64 and SURF 128) to GLOH, SIFT and PCA-SIFT, based on interest points detected with FH-15, the more accurate version of our interest point detector. Both SURF 64 and SURF 128 outperform the other descriptors in a systematic and significant way, with often more than 10% improvement in recall for the same level of precision. For low values of 1-precision, the difference between the normal and extended version of our descriptor is insignificant. For higher 1-precision levels, on the other hand, the extra dimensions of the extended version seem to bring a slight additional improvement. At the same time, they are fast to compute (248 ms for detection and description of 200 interest points in the first Graffiti).

3.6 Conclusion and Outlook

This chapter presented a descriptor based on sums of Haar wavelet components. In order to be invariant to rotation, we first presented a method for reproducible orientation assignment. For the descriptor, the wavelet responses are then calculated according the identified orientation.

Our descriptor outperformed the state-of-the-art methods clearly. It seems that the description of the nature of the underlying image-intensity pattern is more distinctive than histogram based approaches. The simplicity and again the use of integral images made our descriptor a competitive one in terms of speed. Moreover, the Laplacian-based indexing strategy makes the matching step faster without losing in terms of performance. We observed, as already stated in [Mikolajczyk and Schmid 2005], that SIFT has a higher increase in performance than other description schemes for matching strategies based on the nearest neighbour distance. However, both versions of SURF still outperforms the others for all test cases.

The descriptor's performance was shown in combination with our interest point detector. Nevertheless, our descriptor can be applied for any type of detector, except affine-invariant ones which is one of the possible improvements. Our descriptor is easily adaptable for affine invariant interest points (regions), as the ones in [Tuytelaars and Van Gool 2000, Matas *et al.* 2002, Mikolajczyk and Schmid 2002]. However, the speed would be negatively affected by this change. In case of affine invariance, information about the frequency could contribute to the performance of the descriptor. Until now, we can tell about

the nature of the intensity profile around the interest point i.e. homogeneous, gradually increasing, alternative, etc. Additional information about the value of the frequency could improve the results even farther.

In chapter 4.1, we will present a rotation variant version. This version is very appealing for object recognition applications and robot navigation, where the images are all taken horizontally. The speed gain is important and the rotation variance is favourable for the recognition rates.

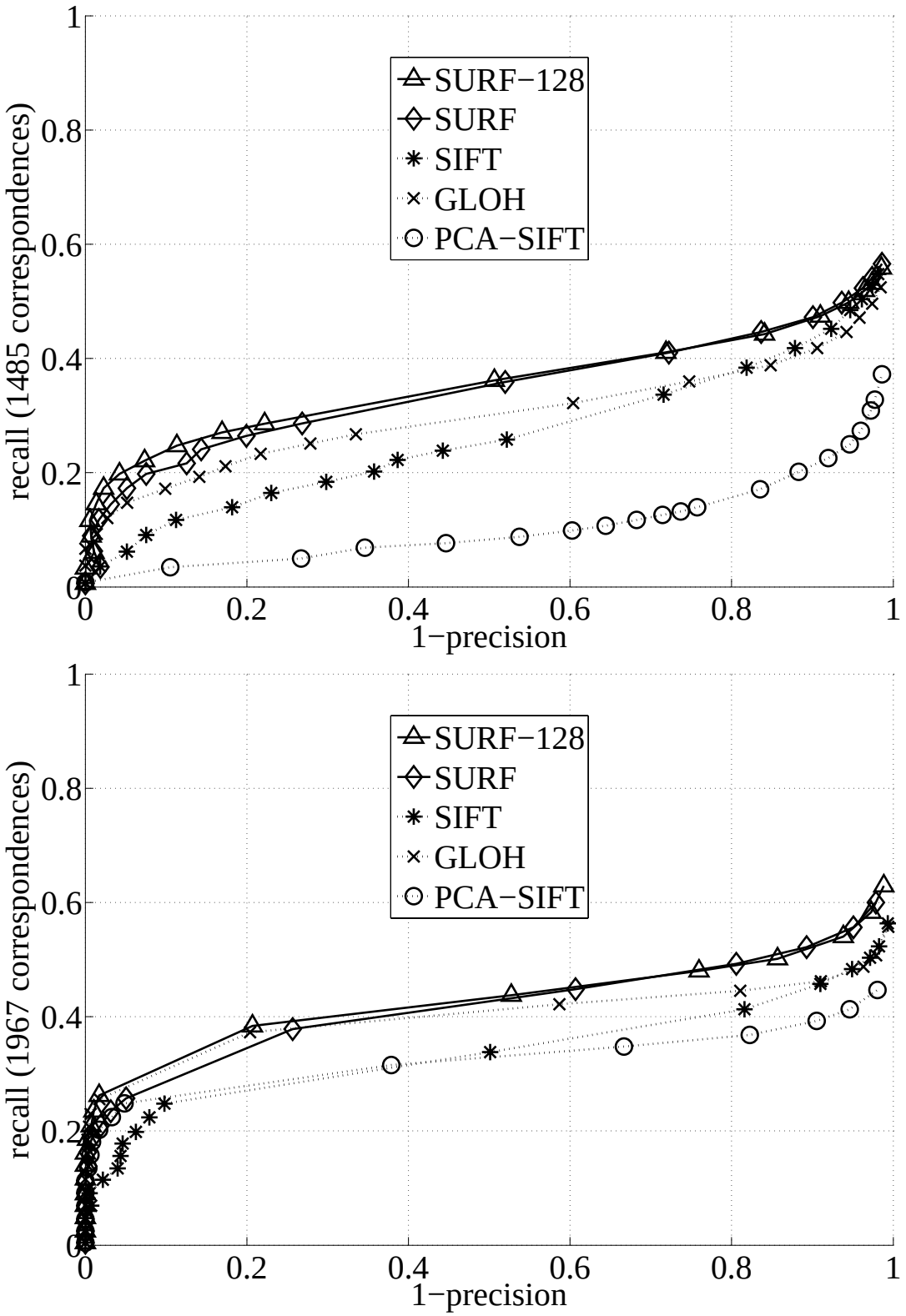

Figure 3.8: Recall, 1-Precision graphs for, top: Viewpoint change of (Wall) 50 degrees, and bottom: scale factor 2 (Boat).

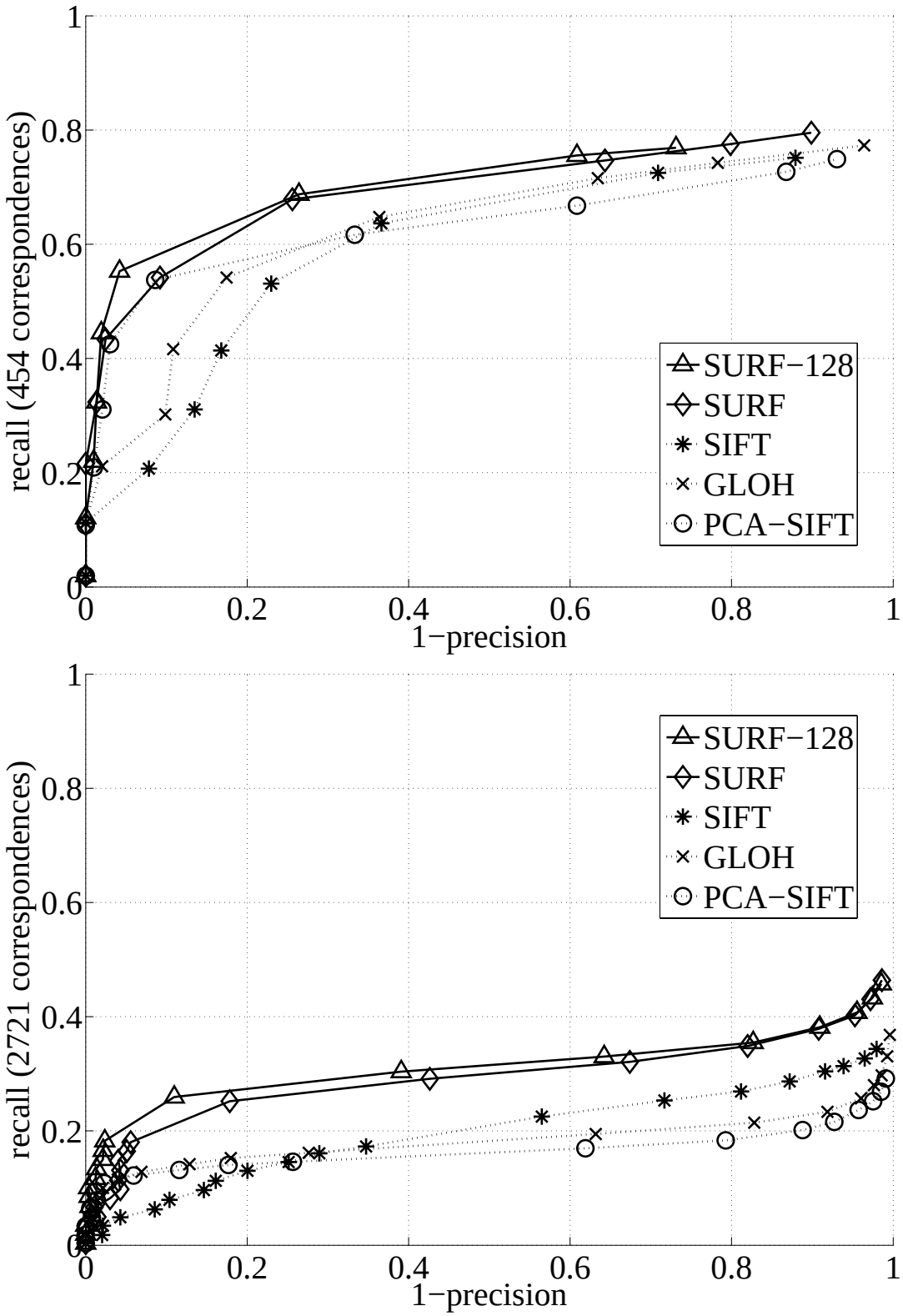

Figure 3.9: Recall, 1-Precision graphs for image blur, top: Bikes sequence, and bottom: Trees sequence.

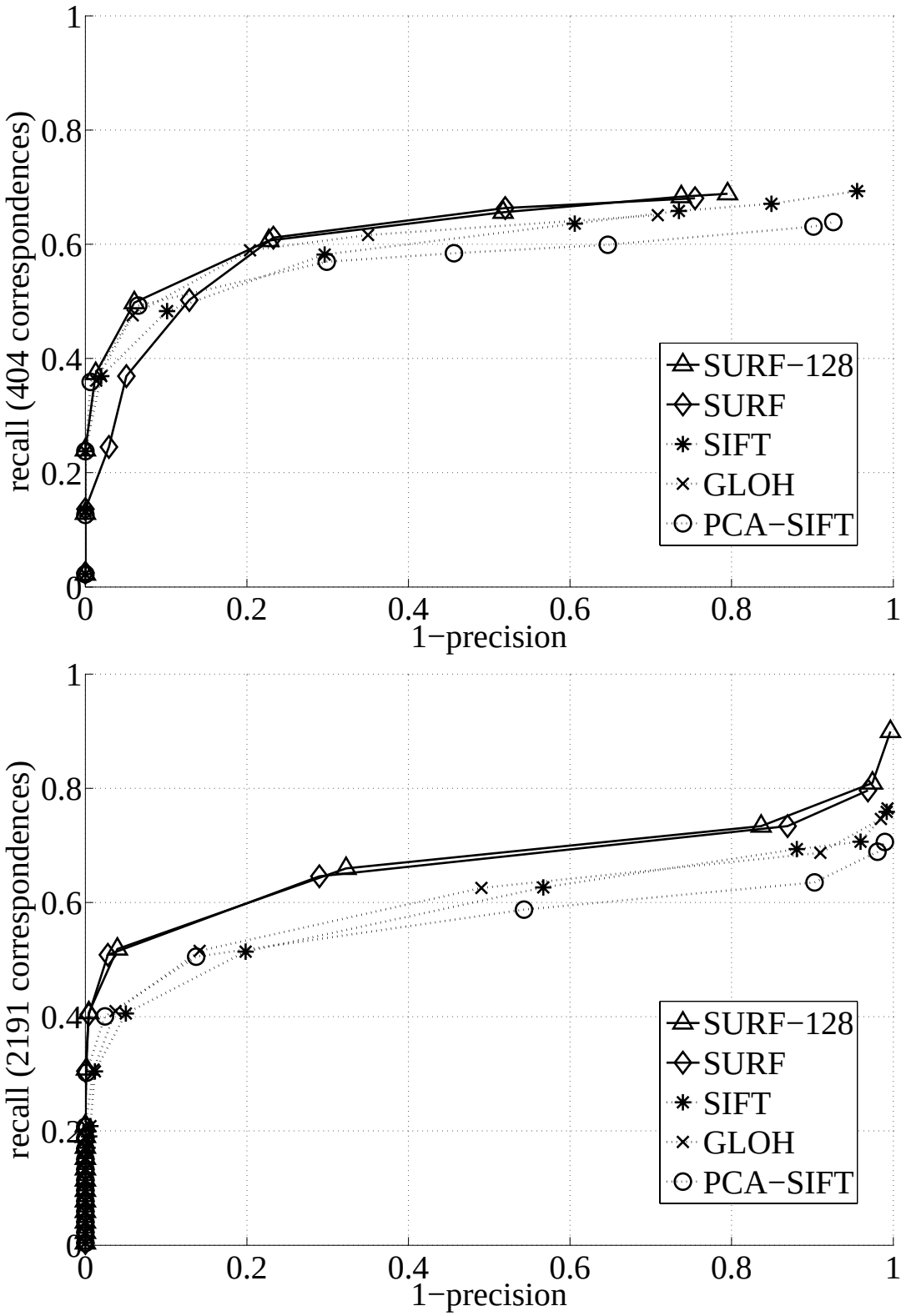

Figure 3.10: Recall, 1-Precision graphs for, top: brightness change (Leuven), and bottom: JPEG compression (Ubc).

4

Applications

In the previous two chapters, we presented an interest point detector and descriptor. The combination of these two approaches leads to a powerful detector/descriptor scheme, called SURF. In this chapter, we show two practical applications (object recognition and image mosaicing) which clearly show the advantage of our method. We introduce a scale-invariant only version of our descriptor, which we refer to as 'upright SURF' (U-SURF). For such kind of applications, the camera typically only rotates about the vertical axis. The benefit of avoiding the overkill of rotation invariance in such cases is not only increased speed, but also increased discriminative power.

For the object recognition application, we designed a prototype of an interactive museum guide. It runs on a tablet PC that features a touchscreen, a webcam and a Bluetooth receiver. This guide recognises objects on display in museums based on images of the latter which are taken directly by the visitor. The objects are recognised based on a *recognition score* which traditionally represents the number of interest point correspondences between the object to be recognised and the objects in the database. Here we propose a new method based on the geometrical mean of the matches in descriptor space. This method results in a significant increase of the recognition rate compared to the traditional approach. As an additional feature of the the interactive museum guide, the computer can determine the visitor's location by receiving signals emitted from Bluetooth senders in the museum, so called BTnodes. This information is used to reduce the search space for the extraction of relevant objects. Hence, the recognition accuracy is increased and the search time reduced. Moreover, this information can be used to indicate the user's current location in the museum. The prototype has been demonstrated to visitors of the Swiss National Museum in Zurich.

As a second application, we show the performance of SURF for automatically fusing retina images to a visually appealing mosaic. Current methods are based on the presence of discernable vascularity in order to detect bifurcations in the vascular tree. In contrast to the state-of-the-art, our approach works even for retina images of highly pathological cases, where no discernable vascularity is present.

4.1 Object Recognition

Many museums present their exhibits in a rather passive and non-engaging way. The visitor has to scan a booklet in order to find some general information about the object. However, searching for information about object after object is quite tedious and, due to the limited content of such booklets, the information found does not always cover the visitor's specific interests. One possibility of making exhibitions more attractive to the visitor is to improve their interaction with the guide.

Recently, several approaches and methods have been proposed that allow visitors to interact with an automatic guide in a museum. [Kusunoki *et al.* 2002] proposed a system for children that uses a sensing board which can rapidly recognise types and locations of multiple objects. It creates an immersive environment by giving audio-visual feedback to the kids. Other approaches are robots that guide users through museums [Burgard *et al.* 1998, Thrun *et al.* 2000]. However, such robots are difficult to adapt to new environments, and they are not appropriate for individual use. An interesting approach using hand-held devices, like mobile phones, was proposed by [Föckler *et al.* 2005], but their recognition technique is limited to constant illumination.

In order to show the performance of our detector/descriptor scheme (SURF), we present an interactive museum guide that is invariant to changes in lighting, viewpoint, scale (zoom), and image rotation (optionally). Our method was implemented on a tablet PC using a conventional USB webcam for image acquisition, see figure 4.1. This hand-held device allows the visitor to simply take a picture of an object of interest from any position and is provided, almost immediately, with a detailed description of it. Moreover, it provides further links and references allowing the visitor to browse comfortably on the Internet for an even broader description of the object.

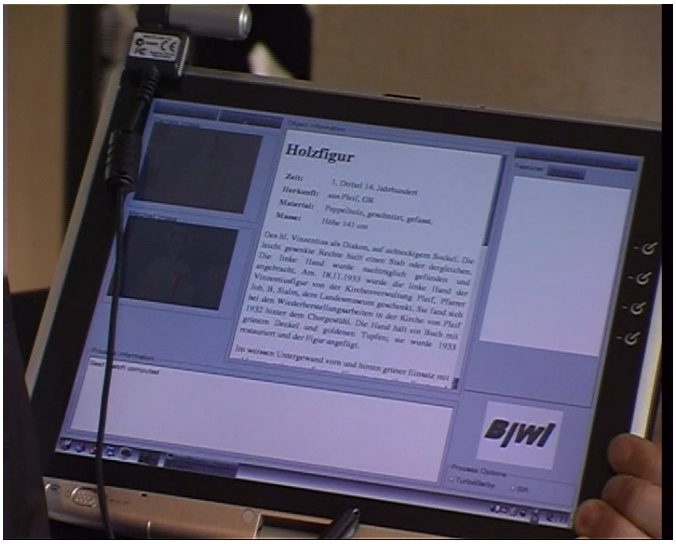

Figure 4.1: Tablet PC with the USB webcam fixed on the screen. The interface of the object-recognition software is operated via a touchscreen.

Bluetooth senders in every exposition room provide information about the current position of the visitor. Therefore, only objects in the corresponding exposition room have to be considered for the recognition task. This fact reduces the recognition speed while increasing the recognition rate. Furthermore, this device can be extended to display location dependent information on a map, such as the closest emergency exits, the toilets or the direction to the next coffee shop relative to the visitor's position. Our museum guide neither imposes a predefined visiting order, nor the inconvenient task of scanning a vast database.

The museum guide has been shown to the public in the framework of the 150 years anniversary celebration of the ETH Zurich, Switzerland. It was demonstrated in the Swiss National Museum Zurich. About 250 visitors took part in the demonstration in 20 guided tours of 10-15 persons each. The object descriptions were read by a synthetic computer voice, which enhanced the comfort offered by the guide.

In this section, we first present an upright version of the SURF detection/description scheme (U-SURF). Then, the image-based object recognition strategy is explained. Furthermore, we present a new object selection method based on the geometric mean in descriptor space. After some specifications about the automatic room detection through Bluetooth beacons, we explain the functionality of the interactive museum guide. Then, we compare SURF to other current in-

terest point descriptors for the specific application of object recognition. Therefore, we apply these different descriptors on the same interest points, detected with our Fast-Hessian detector FH-9 (see chapter 2). Note that the performance of the interest point detector has a minor influence on the results in case of object recognition. This is due to the fact that for this kind of applications, the precise location of the interest point is less important than the number and quality of correct matches between the images.

4.1.1 U-SURF

Rotation-invariant object recognition is not always necessary. Therefore, a scale-invariant-only version of the SURF descriptor is introduced and denoted 'upright SURF' (U-SURF). Indeed, in the scenario of a hand-held interactive museum guide, where the museum visitor holds the device in both hands, it is save to assume that images are mostly taken with an upright camera. Therefore, U-SURF can be used as an alternative descriptor with the benefit of both increased speed and discrimination power. U-SURF is faster than SURF as it does not perform the orientation related computations.

Here we compare the results for SURF, referred to as SURF-64, and some alternative version (SURF-36, SURF-128) as well as for the upright counterparts (U-SURF-64, U-SURF-36, U-SURF-128) that are not invariant to image rotation. As already mentioned in the previous chapter, the difference between SURF and its variants lies in the dimension of the descriptor. SURF-36 extracts the descriptor vector for only 3×3 subregions. SURF-128 is an extended and more distinctive version of SURF. Moreover, it is not much slower to compute than SURF, but slower to match due to its higher dimensionality (but still faster to match than SIFT). This is achieved by the indexing based on the sign of the Laplacian for the individual interest points. This minimal information distinguishes bright blobs on a dark background from the inverse situation. 'Bright' interest points are only matched against other 'bright' interest points and similarly for the 'dark' ones. This permits to almost double the matching speed and it comes at no computational costs, as it has been already computed during the interest point detection step.

4.1.2 Method

Our interactive museum guide contains two different modules. The first is an image-based object recognition module, and the second consists of an automatic exposition room detector using Bluetooth. The combination of both techniques results in a robust and fast object recognition for large image databases. However, this performance shift is not evaluated here.

Object Recognition In order to retrieve the correct object, a database of images has to be established containing images of each object taken from different viewpoints. This fact assures a certain viewpoint independence of the guide and allows it to estimate the approximate direction from which the visitor took the picture. This information can be used as an extension for a more detailed, viewpoint dependent description. A sample image of each of the 20 chosen objects is shown in figure 4.2. These objects of art are made of different materials, have different shapes and encompass wooden statues, paintings, metal and stone items as well as objects enclosed in glass cabinets which produce interfering reflections. The images were taken from substantially different viewpoints under arbitrary scale, rotation and varying lighting conditions. An example of a model image set can be seen in figure 4.3.

For each model image, a set of interest points is computed and described by a scale and (optionally) image-rotation invariant descriptor. The individual descriptor vectors for each model image are then stored as file a on the TabletPC device.

In order to recognise the correct object from the database, we proceed as follows. The input image, taken by the user, is compared to all model images in the database by matching their respective interest point descriptors. In order to create these correspondences M, we used the nearest neighbour ratio matching strategy [Baumberg 2000, Lowe 2004, Mikolajczyk and Schmid 2003]. This states that a matching pair is detected, if its Euclidean distance in descriptor space is closer than 0.8 times the distance to the second nearest neighbour.

The selected object is the one figuring on the model image with the highest recognition score S_R. This score is traditionally the number of total matches in M. However, the presence of mismatches often lead to false detections. This can be avoided with a new alternative for the estimation of the recognition score. Hereby, we calculate the geometrical mean distance d_m to the individual

Figure 4.2: Sample images of the 20 chosen art objects from the Landesmuseum.

nearest neighbours for each image pair. This value is typically smaller for corresponding image pairs than for non-corresponding ones, and it does not depend on the number of extracted features in the individual images. Hence, we maximise the following recognition score.

$$S_R = \text{argmax}(\frac{1}{d_m}) \tag{4.1}$$

Therefore, we chose the object for which the mean distance of its matches is smallest.

Notice that traditional object recognition methods rely on model images, each representing a single object in isolation. In practice however, the necessary segmentation is not always affordable or even possible. For our object recognition application, we use model images where the objects are not separated from the background. Thus, the background also provides features for the matching task. In any given test image, only one object or object group that belongs together is assumed. Hence, object recognition is achieved by image matching. We do not take advantage of fast matching strategies e.g. the best-bin-first

Figure 4.3: Sample of model images and an input image (lower right image) of an object in the museum. Note the important differences in appearance between the model images and the input image. Also, the scale and viewpoint of the input image differs from those of all the model images.

method proposed by David Lowe [Lowe 2004]. Such types of strategies would influence the results of the comparison.

Automatic Room Detection In every exposition room are positioned one or more Bluetooth senders, also called BTnodes [BTnode], see figure 4.4.

A BTnode is a versatile, autonomous wireless communication and computing platform based on a Bluetooth radio, a second low-power radio and a micro controller. Every BTnode covers a specific area of the museum and provides it with a localisation signal broadcasted at constant intervals. The signal received by the interactive museum guide is used for two purposes. Firstly, the position of the visitor can be evaluated and displayed on a map. Further location-dependent information may then be retrieved. Secondly, as several images of an object are needed in order to robustly recognise it, the number of images in the database increases rapidly depending on the number of objects featured in the museum. This risks to slow the object recognition process down. Moreover, as more similar objects may enter the database, the accuracy of the recognition decreases. Classical object recognition methods would be computationally too expensive to get any result in time. To increase the matching speed, the search

is reduced to objects in the area close to the visitor. This area is defined with the signal of a BTnode.

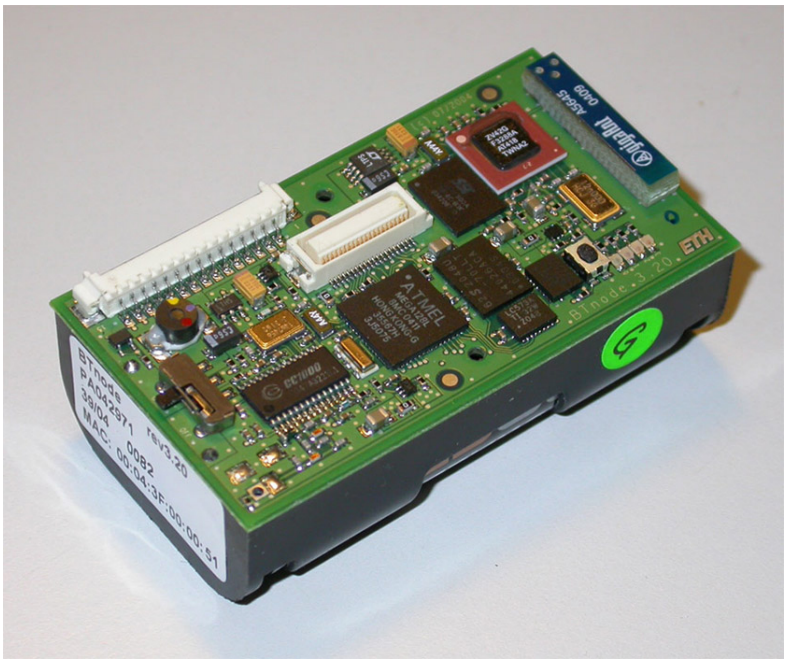

Figure 4.4: Image of a BTnode. These devices were placed in different exposition rooms of the museum. Each node broadcasts its identification number at regular time intervals.

4.1.3 Adding New Objects

Adding new objects to the database is easily accomplished. First, a few model images of the object have to be taken from different viewpoints with any kind of camera. The size of the images must be reduced in order to get a reasonable detection time without loosing important details of the object. We chose 320 $\times$ 240 pixels. Second, interest points of the model images have to be detected and characterised by our SURF descriptor. Finally, the model image names have to be indexed in a table in order to attribute the documentation to the figured object. Additionally, the identification number of the BTnode, covering the area where the object is located, has to be mentioned in the table.

4.1.4 Interactive Museum Guide

As soon as the computer receives the signal of a BTnode, it recognises the room in which it is located and selects the part of the database representing the objects in that same room. For the demonstration in the Swiss National Museum, we used only two of such BTnodes (see Figure 4.5), one in the entrance hall and the other in the first exposition room of the museum. Once the visitor passes the threshold to the entrance hall, the computer receives the signal of the first BTnode and *says* "Welcome to the Landesmuseum[1]". As soon as the visitor enters the exposition room, the computer *says* "Exposition room" and launches automatically the object recognition application. The interface of the latter is shown in Figure 4.6.

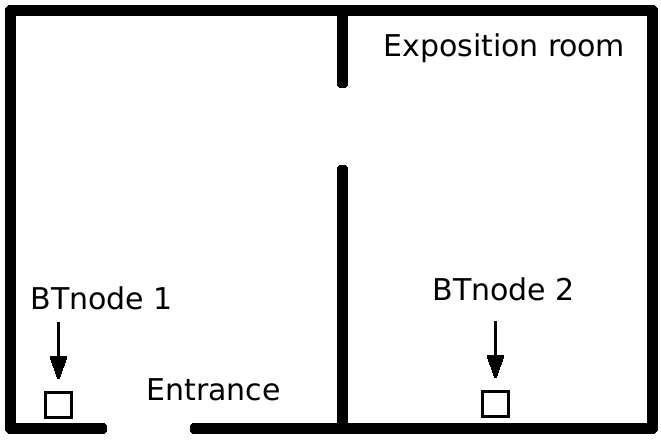

Figure 4.5: Schematic representation of the visited rooms and the BTnode distribution in the museum.

When the user takes a picture of an exhibit, the computer displays, almost immediately, the requested information in a browser window. Furthermore, the visitor can browse to some more specific information on the Intranet / Internet for related objects that are currently exposed in the museum (e.g. made by the same artist). Also, the visitor has the option to have the description in the browser to be *read* by the computer via a text-to-speech synthesis engine.

[1] *Landesmuseum* means "National Museum"

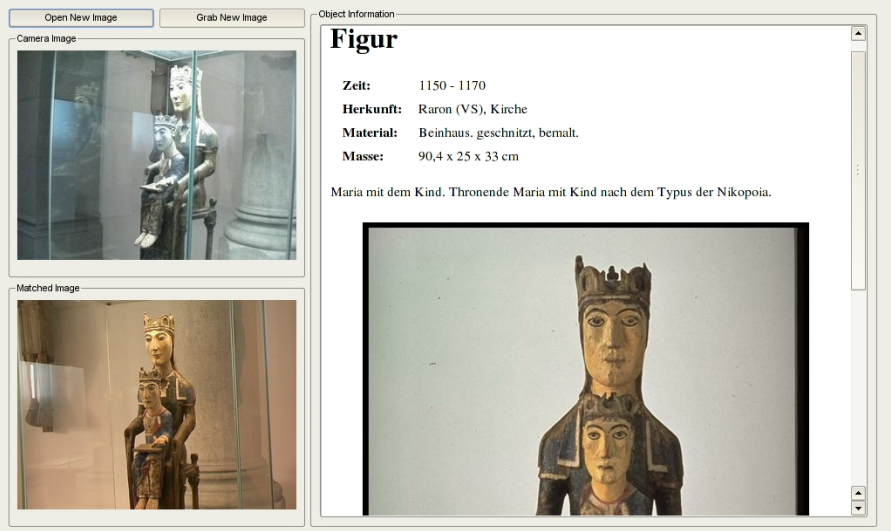

Figure 4.6: Interface of the object recognition application. On the upper left, the input image taken by the visitor is shown. On the lower left, the matched reference image is displayed. On the right hand side, the browser window can be seen. There, the description of the object, associated to the matched model image, is shown.

4.1.5 Results

For the comparison, we used GLOH [Mikolajczyk and Schmid 2005], SIFT [Lowe 2004] and PCA-SIFT [Ke and Sukthankar 2004] in combination with our Fast-Hessian interest points described in chapter 2. These interest points are the same for all descriptors. This allows for a more accurate comparison of the descriptor performances.

As already mentioned, for each of the 20 objects of art in our database (see figure 4.2), images of size 320×240 were taken from different viewing angles. This allows for a certain degree of view-point independence. The database includes a total of 205 model images. These are grouped in two model sets (M1 and M2) with 105 and 100 images, respectively. The reasons for the choice of two different model sets are the use of two different cameras and the presence of different lighting conditions. Moreover, less model images for a given object represents a more challenging situation for object recognition. For similar reasons, we built 3 different test sets (T1-T3) with a total of 116 images (42, 34, 40). Each set contains one or more images of all objects.

The test image sets were evaluated on each of the model sets. The obtained recognition results are shown in table 4.1 and 4.2. Listed are the results for

Method	Time	Recognition Rate						Total
	(s)	T1/M1	T2/M1	T3/M1	T1/M2	T2/M2	T3/M2	
SURF-36	10+16	81	79	85	71	94	78	81.0
SURF-64	10+26	88	79	90	69	100	78	83.6
SURF-128	10+50	81	91	90	71	97	75	83.5
U-SURF-36	7+16	74	79	90	74	91	75	80.2
U-SURF-64	7+26	86	85	88	74	94	78	83.8
U-SURF-128	7+50	83	94	95	76	94	80	86.5
GLOH	NA+89	71	94	93	67	91	70	80.2
SIFT	127+89	79	88	90	76	91	75	82.7
PCA-SIFT	NA+25	74	82	93	71	97	80	82.3

Table 4.1: Image matching results for different SURF versions and SIFT. Listed are both the total detection and matching time for all 3 test sets combined with the model sets.

the standard recognition score based on the maximum number of matches (table 4.1) and the geometric mean (table 4.2) as described in equation (4.1). It can be seen that most versions of SURF outperform SIFT for most test sets while being substantially faster for both computation and matching. These timings express the "real" computation and matching time without the offset for reading and writing to files etc. For the computation of the descriptor, SURF is about 13 times faster than SIFT. U-SURF is another 30% faster than SURF. The different binning strategies show all the same timing for this test, but they are slightly different (not measurable). The difference for the matching is also significant, however, the values are not representative. Fast matching techniques like the best-bin-first approach proposed by David Lowe SURF would reduce the difference for descriptors of equal size e.g. SURF-128, SIFT and GLOH. The reported computation times were achieved on a Linux TabletPC equipped with an Intel Pentium M processor running at 1.7 GHz. The recognition rates for the new recognition score, based on the geometric mean, increase up to 10%.

Figures 4.7 and 4.8 show cases where SURF and SIFT fail to recognise the same foreground objects. The matches are represented with white lines connecting the respective locations. On the bottom of figure 4.7, two image pairs are displayed where the foreground object is not correctly recognised by the SURF algorithm. Note that a correct match was found for valid objects that are visible in the background. However, in the context of mismatched foreground objects, SIFT did not find any matches with objects in the background that were also contained in the model database.

Method	Time	Recognition Rate						Total
	(s)	T1/M1	T2/M1	T3/M1	T1/M2	T2/M2	T3/M2	
SURF-36	10+16	86	88	90	76	97	73	84.5
SURF-64	10+26	83	91	88	83	97	83	87.1
SURF-128	10+50	88	85	93	79	100	85	88.0
U-SURF-36	7+16	86	100	98	81	100	85	91.1
U-SURF-64	7+26	86	94	93	81	100	85	89.4
U-SURF-128	7+50	86	94	95	86	100	90	91.5
GLOH	NA+89	76	94	98	76	94	83	86.3
SIFT	127+89	83	91	100	76	94	80	86.9
PCA-SIFT	NA+25	71	88	95	74	97	83	84.1

Table 4.2: Image matching results for different SURF versions and SIFT with the new matching strategy. Listed are both the total detection and matching time for all test sets combined with the model sets.

Figures 4.9 and 4.10 show typical cases where either SIFT or SURF fail to recognise the correct foreground object. Note that the goblet shown on the top row of figure 4.9 was twice not correctly recognised by SIFT. Not a single match was found on the object itself, but many on the enclosing showcase. However, many model objects contained in the database are enclosed in showcases and can thus lead to false matches when it comes to the recognition of the foreground object of interest.

Figure 4.10 (left) shows a case where only SURF produces a false recognition. Notice that many false matches were found between the object of interest and a background image that is not part of the model database. Hence, test objects can be falsely recognised due to model images that contain similar arbitrary background objects that are not part of the objects of interest.

Finally, figure 4.10 (right) shows a successfully recognised object. In that specific case, the background information was helpful for the recognition of the object.

4.1.6 Discussion

With the computational efficiency of SURF, the object recognition can be performed instantaneously (U-SURF: 0.3 s) for the 20 objects on which we tested the different schemes. The images were taken with a low-quality webcam. However, this fact affected the results only up to a limited extent. Note that in contrast to the approach described in [Föckler *et al.* 2005], all the tested

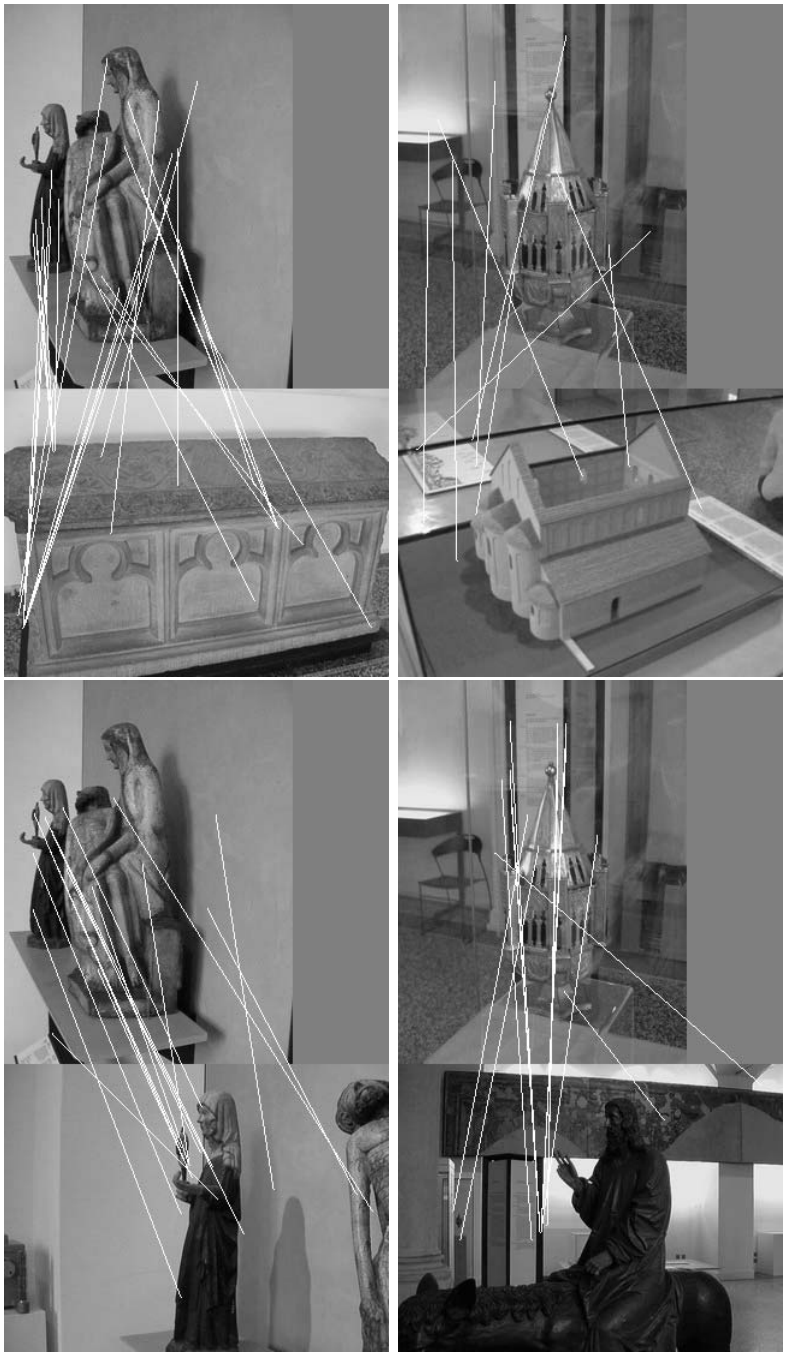

Figure 4.7: Common image matching mistakes. Both SIFT (top row) and SURF (bottom row) fail to recognise the same test objects. In each of the four two-image combinations, test images are shown on the top and matched model images on the bottom.

schemes do not use colour information for the object recognition. This is one of the reasons for the above-mentioned recognition robustness under various

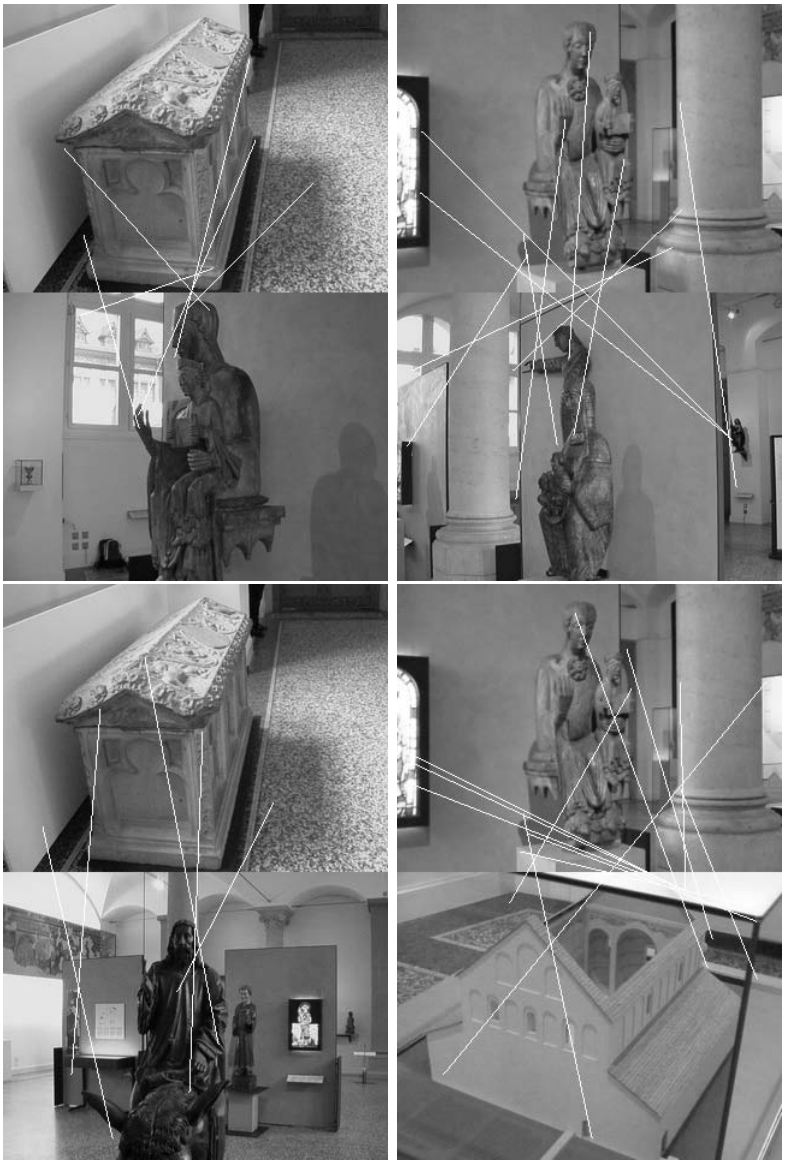

Figure 4.8: Common image matching mistakes. Both SIFT (top row)and SURF (bottom row) fail to recognise the same test object. In each of the four two-image combinations, test images are shown on the top and matched model images on the bottom.

lighting conditions. We experimentally verified that illumination variations, caused by artificial and natural lighting, lead to low recognition results when colour was used as additional information.

The fact that the model images include background information can be helpful for the object recognition. Especially in cases, where the objects of interest are too similar or do not provide enough robust and discriminant features, back-

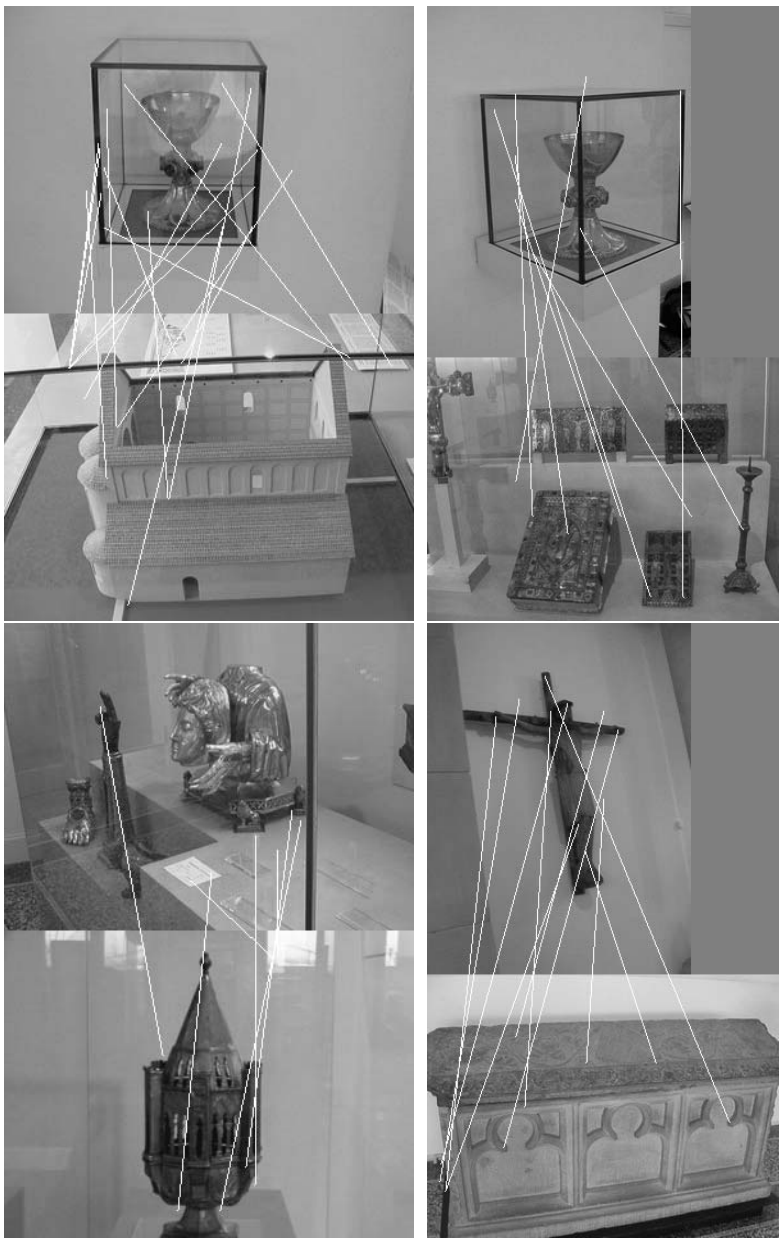

Figure 4.9: Individual image matching mistakes produced by SIFT. In each of the four image combinations, test images are shown in the top row and the matched model image in the bottom row.

ground information may lead to recognise the object successfully. However, if a dominating background object is present in the test image, the recognition method fires on the object in the background rather than on the one in the foreground and leads to a false recognition, see figure 4.7.

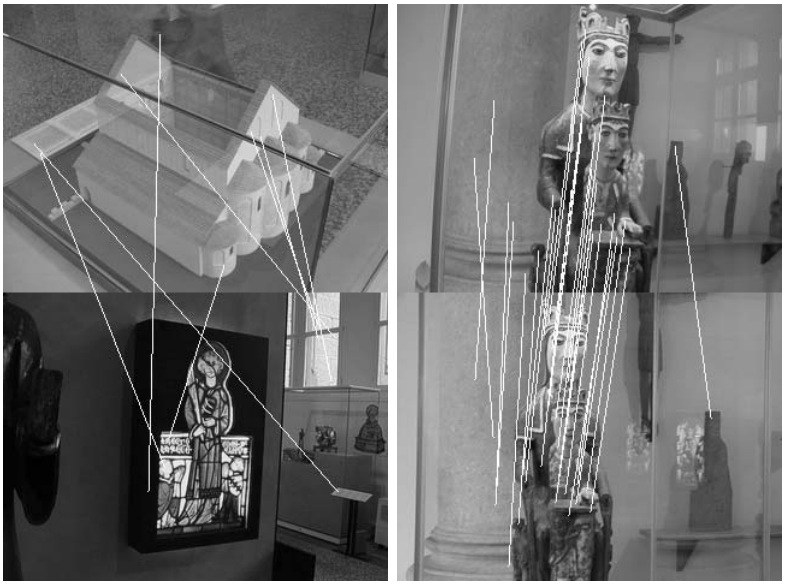

Figure 4.10: Individual image matching mistake produced by SURF (left) and a successfully recognised object (right). The test image is shown in top and the matched model image on the bottom.

4.2 Mosaicing

Another practical computer vision application using interest point correspondences is image mosaicing, also called image stitching. Image mosaics are often created as panoramas of a scene, generated from individual images [Brown *et al.* 2005]. Other applications use mosaic images for the representation of the information contained in video sequences [Bartoli *et al.* 2004].

Here, we focus on image mosaicing of retinal images for medical applications, like pathology diagnosis, laser surgery, etc. This is necessary, as only a small portion of the retina can be imaged at once with an ophthalmoloscope. A 7-image (2256×2032) protocol is often used to map the retina. To get a better overview of the patient's retina for localisation of irregularities or diseases, mosaic images would be practical. The construction of mosaic images of the retina is a challenging task, as the image overlap is low in the standard 7-image capturing protocol. Furthermore, the eye is quasi spherical and requires more sophisticated image transformations than just a planar projection (homographies).

[Can *et al.* 2002] proposed a benchmarking algorithm for the automatic registration of image pairs on the human retina. The authors matched the images using bifurcations of the vascular tree on the retinal surface. These bifurcations are simple to detect, but lack distinctiveness, and their localisation is often quite inaccurate. Furthermore, the number of bifurcations in an image is typically low and images of pathological retinas often lack a segmentable vascular tree. This results in inaccurate image registration or, in the worst case, even to a complete failure to build the mosaic.

In order to test SURF on practical computer vision applications, we applied it for building such retina mosaics. In contrast to the state-of-the-art, the proposed approach is independent of the retina's vascular tree and thus reliably works even in highly pathological cases. The eye movements are limited to rotations around the horizontal and vertical axis, but do not turn around the optical axis of the camera. These properties can be exploited in order to ease the image matching and registration step. Furthermore, the upright version of SURF (U-SURF) can be used for this specific application for additional robustness. The presented results are obtained using the upright version of our descriptor U-SURF-128 as presented in the last section.

4.2.1 Method

The method for obtaining a mosaic image from multiple images of the retina can be divided into four main steps. Firstly, interest points are detected and matched for all overlapping image pairs. Secondly, possible mismatches are detected and removed by a simple homography filter (explained below). This is possible because the small retinal patches imaged by the zoomed-in ophtalmoscope are planar to a good approximation. Thirdly, the image that is overlapping with the most neighbouring images is selected as the base image and thus defines the reference coordinate frame. In order to compensate for the spherical nature of the retina, a quadratic image transformation is estimated for all overlapping image pairs. Then, a global optimisation is performed on the initial transformations, minimising the back projection error of the mutual interest point correspondences. Finally, all images are warped into the coordinated frame of the reference image and a multi-band blending [Burt and Adelson 1983] over six octaves is used to seamlessly fuse the individual images.

4.2.2 Image Correspondences

Given a set of N input images for the mosaic. At first, the overlapping images have to be identified. For this purpose, the interest point correspondences for all possible image pairs of the N images are established. The matching between an image pair is carried out in the same way as for the object recognition application. For all interest points in a first image I_1, we compare the similarity to every interest point in the second image I_2 by computing the Euclidean distance of their respective descriptor vector. A match is selected, if its distance in descriptor space is smaller than 0.8 times the distance of the second nearest neighbour. Note that, using this matching strategy, a given interest point $\mathbf{x}_2$ in I_2 can have multiple matches in I_1, but not vice versa. We take care of this by keeping only the one with the smallest distance in descriptor space to its correspondence in I_1.

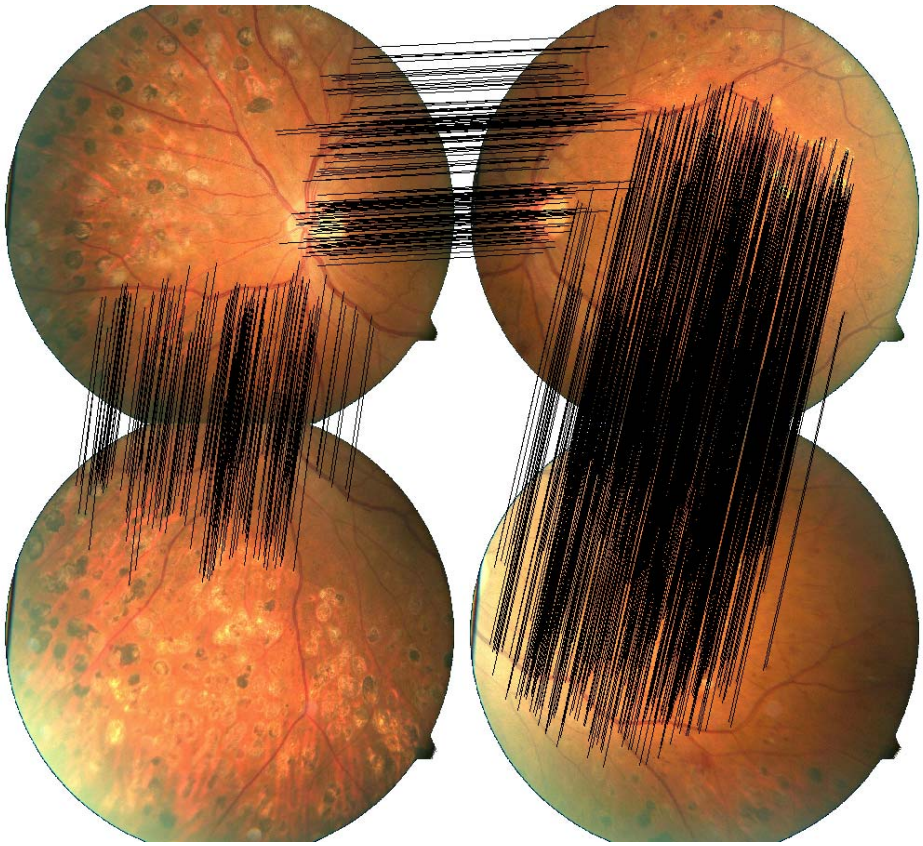

Figure 4.11: Example of pairwise image correspondences between four input images. The lines represent the matched interest points for overlapping image regions. Notice that there is no overlap between the two lower images.

As the image surface support is supposed to be almost planar, the image correspondences can be further improved. First, a *homography* is robustly estimated using the traditional RANSAC scheme [Fischler and Bolles 1981, Hartley and Zisserman 2004]. Given a point $\mathbf{X}$ on a planar surface and its projections, $\mathbf{x}_1$ and $\mathbf{x}_2$, in two images taken from different view points, respectively. A homography $\mathbf{H}$ describes the transformation which connects $\mathbf{x}_1$ and $\mathbf{x}_2$ for any point $\mathbf{X}$ on that surface.

$$\boxed{\mathbf{x}_2 = \mathbf{H}\mathbf{x}_1} \tag{4.2}$$

Therefore, the mismatches are identified and removed. If the transfer error $d(\mathbf{x}_2, \mathbf{H}\mathbf{x}_1)^2$ of a given interest point correspondence ($\mathbf{x}_1$ and $\mathbf{x}_2$) is within a certain tolerance region, the match is considered a correct one. Mismatches are discarded and not considered for the consequent steps. Finally, more interest point correspondences can be identified. For a given interest point $\mathbf{x}_1$ in image I_1, only candidates in a certain region around the transformed location (according equation 4.2) are considered as possible matches. Figure 4.11 shows an example of typical interest point correspondences between four images. The large numbers of matches between the overlapping parts of the images assure an accurate image registration for the mosaicing process.

To construct the final mosaic a coordinate frame must be defined. For simplicity this reference coordinate frame is generally chosen as the image coordinates of one of the stitched images, also referred to as the *anchor image*. The anchor image is selected as the image with the highest total number of overlaps with different images.

4.2.3 Quadratic Transformation

Even if overlapping regions of image pairs can be related using homographies, a planar transformation model for the mosaic would result in a false representation of distances. This artefact is more pronounced for image informations farther away from the centre of the mosaic (anchor image). The curved nature of the retina can be taken into account by using a quadratic transformation model.

It transforms a point $\mathbf{x} = (x,\, y)^\top$ of an input image to a point $\mathbf{x}' = (x',\, y')^\top$ in the mosaic image.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} & \theta_{15} & \theta_{16} \\ \theta_{21} & \theta_{22} & \theta_{23} & \theta_{24} & \theta_{25} & \theta_{26} \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \\ x \\ y \\ 1 \end{pmatrix} \tag{4.3}$$

In order to initialise the global optimisation process, the quadratic transformation for each overlapping image pair is estimated solving the equation system shown in equation (4.3). For image pairs having no mutual overlap, a different, indirect approach is followed. The quadratic transformation of such image pairs is derived as linear combination of two or more known transformations. As an example, the mapping of image (a) onto image (d) in figure 4.12 can be calculated via the 5 already known transformations (a-b, a-c, b-c, b-d, c-d). There are different possible combinations to estimate the transformation (a-d). The question arises, which combination delivers the most reliable transformation for that task. This is done by weighting the connections with the number of matches between the involved image pairs. In order to find the ideal path from image (a) to image (d), we aim at minimising the number of connections while maximising the number of total matches for the corresponding image pairs. This is a well known graph problem of finding the maximum spanning tree [Garey and Johnson 1979]. These initial transformation estimates are then subject to a global optimisation process, in which the sum-squared reprojection error is minimised using a downhill simplex method [Nelder and Mead 1965].

4.2.4 Image Stitching

At this stage, the geometrical relationship between the different images is known. In order to produce the final mosaic, all images are warped into the spatial domain of the anchor image. What remains is to produce a visually appealing, fused mosaic with smooth image boundaries. This is achieved by multi-band blending [Burt and Adelson 1983]. The idea behind this approach is to blend the high frequencies over a small spatial range and the low frequencies over a large spatial range. This is performed for multiple frequency bands.

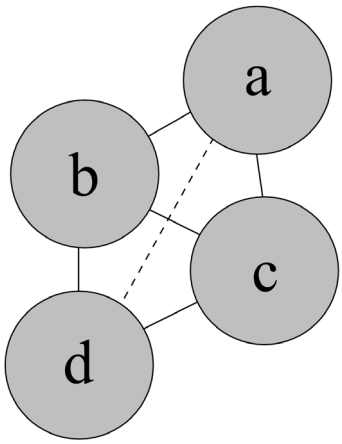

Figure 4.12: In order to connect the images (a) and (d) with a quadric transformation, multiple linear combinations through existing image pair connections are possible. The dashed line represents the absence of overlap or correspondences.

4.3 Conclusion and Outlook

In this chapter we have described the functionality of an interactive museum guide and a mosaicing application for medical purposes. Both applications are based on SURF and demonstrate its performance.

The recognition rate of the museum guide is clearly higher when using SURF than with the other state-of-the-art methods we tried. Moreover, the overall speed for the recognition of a given object is significantly reduced. Therefore, our guide allows to robustly recognise museum exhibits under difficult environmental conditions almost immediately. Furthermore, it is running on standard low-cost hardware. We also presented a possibility to improve the accuracy and speed of object recognition by combining image-based object recognition with automatic room detection through Bluetooth emitters.

We also demonstrated that SURF can be efficiently used to mosaic highly self-similar retina images even for cases with no discernable vascularisation. The algorithm is currently integrated into the interventional planning system. Future work could aim at approaches allowing to utilise the applied feature matching method for intra-operative navigation support. Moreover, a Mercator representation of the retina i.e. cylindrical projection, could be useful in a later step when the geometrical form of the eye-ball is known.

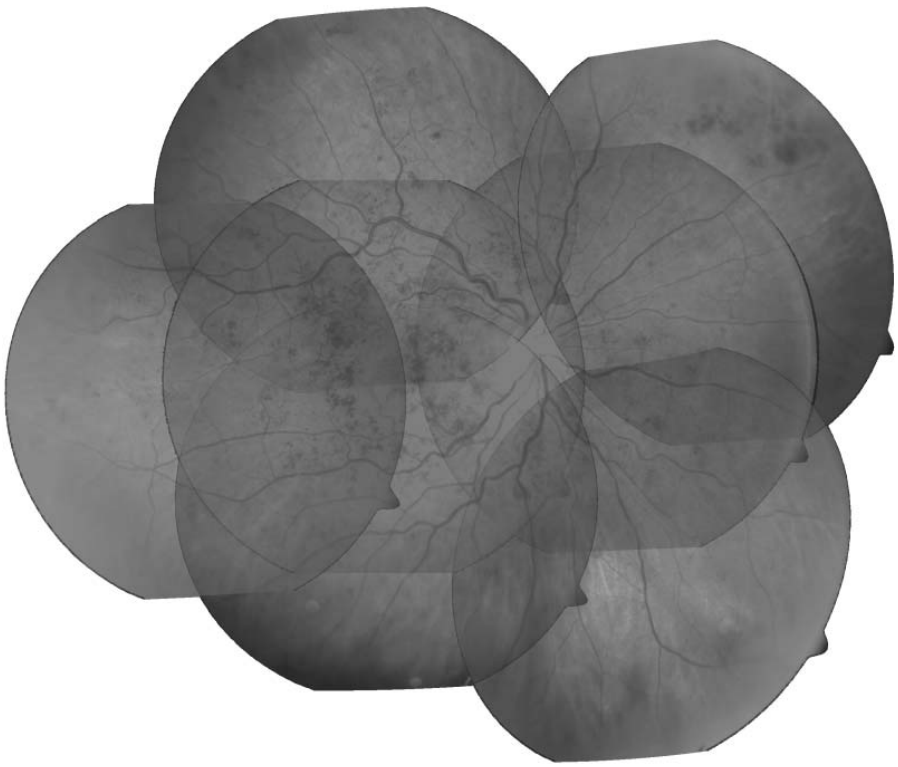

Figure 4.13: After the global optimisation, the images are geometrically aligned, but the image borders are still visible.

Figure 4.14: Mosaic of a healthy human retina.

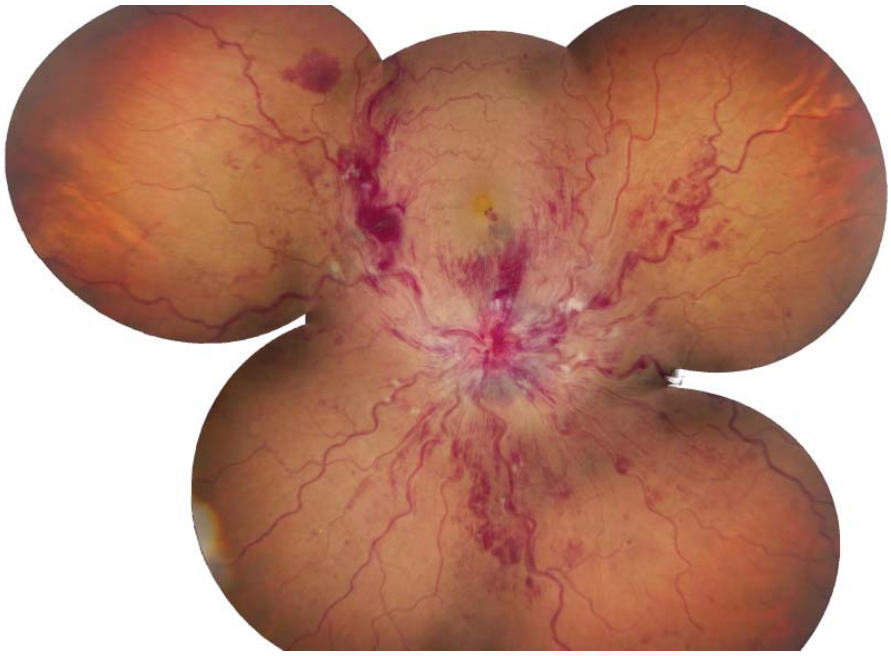

Figure 4.15: Mosaic of a sick human retina.

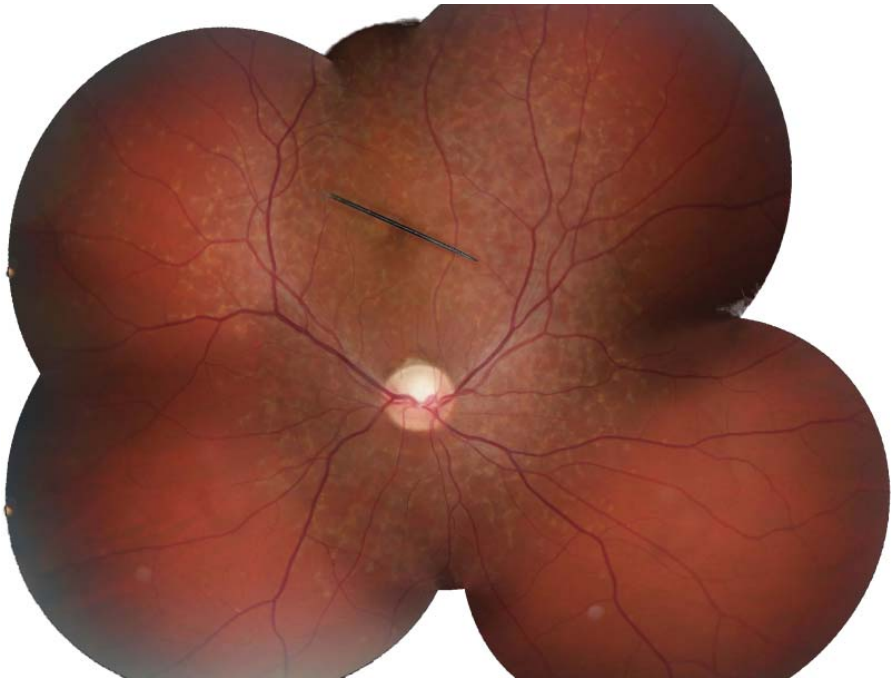

Figure 4.16: Mosaic of a healthy human retina, injured by a foreign object .

II

Line Segment Correspondences

Outline

As described in the previous part, point matches between two images are most useful for many computer vision applications. However, not all types of scenes provide a sufficient number of such interest points. This number depends strongly on the amount of distinctive texture present in a given scene. Images of poorly textured scenes, like indoor environments, produce only a few or no matches at all. In this part of the thesis, we address the problem of finding correspondences between images of such poorly-textured scenes, namely man-made environments with homogeneous surfaces, like blank walls in architectural interiors.

Despite lacking texture, images of such scenes often contain line segments, which can be used as additional features. Line segments convey an important amount of geometrical and topological information about the constitution of the scene.

In the following chapter, we present the detection and appearance based description of line segments as well as a method for merging partially detected ones. Furthermore, we propose a novel matcher for straight line segments, based on their descriptors and their topological layout. It is capable of reliably finding correspondences among line segments between two views taken from substantially different viewpoints. It works in a completely uncalibrated setting, and hence does not assume epipolar geometry to be known beforehand.

Finally, a new method for guided matching of line segments is proposed. We generate an initial set of line segment correspondences, and then iteratively increase their number by adding matches consistent with the topological structure of the current ones. If vanishing points are detected, this additional information is used to further increase the number of matches and filter mismatches. Moreover, we show the benefits of matching line segments *and* interest points in a combined fashion.

5

Line Segment Matching

Although the geometrical properties of lines and line segments over multiple views have already been thoroughly studied back in the 1980's, there are just a few practical computer vision applications based on line segment correspondences compared to the large number of existing algorithms that are based on interest point correspondences. The reasons for this are manifold. First, line segments are more difficult to detect. This is due to causes like radial distortion, which results in bended lines; to insufficient gradient and to aliasing effects. All these factors lead to partially detected line segments. Secondly, the end-points of the line segments are not reliably detected. This makes the use of epipolar geometry for line segments in two views not as appealing as for interest points. We will study this problem in the third part of this thesis. Thirdly, uncertain end point locations result in the fact that for different images, different parts of the same physical line segment are detected. As the appearance of the neighbourhood along the line segment may change drastically, the detected parts may 'look' completely different. As a consequence, this renders purely appearance-based matching of line segments more challenging.

Current line segment matching methods describe the line segments using a mixture of different information like brightness, direction (angle between the line and the x-axis of the image), coordinates of end points, mid points, etc. These heterogenous characteristics are somehow mixed together and used for the matching, e.g. via the Mahalanobis distance [Pellejero *et al.* 2003]. Furthermore, such methods are limited to small-baseline conditions and are sensitive to image rotation. Until now, only a few methods for automatic line segment matching for wide-baseline stereo exist. [Schmid and Zisserman 1997] perform guided line matching using a plane sweep algorithm. However, their

approach requires the prior knowledge of the epipolar geometry. [Lourakis *et al.* 1998] use the '2 lines + 2 points' projective invariant for images of planar surfaces, and therefore their method is limited to such scenes. Several other approaches restricted to small-baseline cases have been proposed (e.g. [Pellejero *et al.* 2003]). Often, geometrical constraints like homographies and projective invariants are used to grow the number of matches between two images. These constraints apply only for a specific type of scenes (planar), but do not hold in the general case. Furthermore, in order to estimate a homography, at least four lines have to be detected for every single plane. However, this minimal number is not always present not to mention a sufficient number of additional line segments to geometrically verify the choice of the homography. Even if the epipolar geometry is available, it cannot be fully applied to line segments that have the same or a similar orientation as the epipolar lines. Therefore, other properties have to be used for robust line segment matching.

We propose a new matching strategy that is able to find correspondences between line segment in two images, which are taken under wide-baseline conditions. Therefore, we first introduce a new *descriptor* for line segments, based on the colour distribution on each side along the line segment. This distribution is expressed by means of histograms. Also, the line segment is given a robust direction, which is independent to image rotation. A direction is needed for e.g. merging collinear line segments and identifying mismatches based on the overall topology.

Once all line segments have been characterised by a descriptor, partially-detected, collinear line segments are merged in order to reduce the number of line segment fragments and generate more substantial features. A quadratic metric is used to calculate the dissimilarity (distance) between the line segment descriptors (colour histograms) to provide initial candidate matches to a given line segment.

Some examples will show that a classical 1-to-1 matching strategy is not robust enough for poorly-textured scenes, imaged under wide-baseline conditions. Hence, we introduce a robust matching scheme for line segments based on both their appearance and topological configuration. Instead of considering only the best appearance-based candidates, we work with multiple candidate matches for a given line segment. A *topological filter*, automatically converges to a robust solution. It filters mismatches that typically do not fit in the overall topological configuration. Furthermore, it is possible to increases the number of correspondences by reducing the search space for valid matches, using the

same topological constraint as for the filter. If interest point matches are available, they will be automatically integrated into the topological configuration, and exploited in combination. As will be shown, the combined information of line segments and interest points yields better results than with line segments alone.

5.1 Line Segment Extraction

There are two main approaches for extracting line segments both based on a prior edge detection step. The first approach applies the Hough transform to the detected edge pixels (edgels) in order to first detect the infinite support lines and then the end points of the line segments [Palmer *et al.* 1997]. The disadvantage of the Hough transform is that it fires on many false lines in highly-textured images due to accidental linear arrangements of edgels. Furthermore, the location of the detected lines is often not accurate enough for further processing.

The second approach groups the detected edgels into significant straight versus curvilinear structures. The quality of a straight line fit to a list of points can be estimated by calculating the ratio of the length of the line segment divided by the maximum deviation of any point from the line (Curvature estimation). This procedure can be applied on different scales, in order to achieve scale invariance [Lowe 1987]. A widely used approach of a similar nature was proposed by [Burns *et al.* 1986]. The algorithm groups pixels with similar gradient orientation into edge support regions and fits line segments to these regions.

The approach that is used here follows the second line segment detection scheme on one scale. First, Canny edges [Canny 1986] are extracted and a list of linked edgels is created. A snake [Kass *et al.* 1987, Blake and Isard 1998] is applied to every item in the list in order to split the edgel chains at locations of high curvature. Straight line segments are then fitted to the remaining edge segments using orthogonal regression. For this step, only corrections of a certain deviation to the originally detected edge are tolerated. Finally, the remaining curvilinear edges and small line segments are rejected. Note that this method may provide different line segments depending on the chosen parameters for the Canny edge detection. Here, we used the same parameters for all experiments in this thesis (see table 5.1). However, by investing some time in finding the ideal parameters for every single image, the results of the proposed matching strategy may be better. The choice of our parameters is based on the weak

σ	low	high
1.2	2	8

Table 5.1: Selected parameters for the Canny edge detector. σ is the standard deviation for the Gaussian smoothing operator. *low* and *high* denote the lower and higher values for the hysteresis thresholding, respectively. The Canny edge detector has more parameters that have to be specified, but these three have the highest influence on the outcome.

contrast present in the majority of our example images and has been evaluated experimentally on the latter.

Just to give prior notice: the first priority in the list of further improvements is the development of a robust line segment detector. The current approaches are still not robust enough for reliable line segment detection under varying conditions.

5.2 Histogram Based Descriptor

Because of the unknown motion range, epipolar geometry, and the unstable end point locations, line segments cannot be described using the classic correlation windows or corrected correlation patches [Schmid and Zisserman 1997]. However, line segments are locations where the gradient is high. Typically, a line segment is the intersection of two differently coloured or oriented areas. Also, different orientations tend to yield different colours in the image, due to differences in illumination. These colours undergo only slight changes under changing viewpoint, even if the camera motion is important. This is the case for most predominantly diffuse surfaces, and also for surfaces that are not viewed from a specular direction. The colour distribution on either side of a line segment is therefore an appropriate measure to compare the similarity of line segments. In order to guarantee a certain invariance towards geometrical transformations, The neighbouring colours are extracted along the line segments and not in perpendicular direction. Therefore, we consider the colour profiles along each side of a line segment for the computation of the descriptor.

The colour profiles are computed along two stripes parallel to the line segment, one on either side separately. The separation between those stripes has to be as small as possible, but not smaller than the standard deviation of the Gaussian

filter used for the Canny edge detection. The standard deviation of the Gaussian smoothing operator defines the scale in scale space analysis. Therefore, if the distance separating the stripes is smaller than the scale at which the line segments are detected, we risk to describe the edge segment itself, which is not particularly distinctive. As the standard deviation for the Gaussian smoothing is constant in our implementation ($\sigma = 1.2$), we chose a distance of 4 pixels (see figure 5.1 on the right hand side) in all our experiments. The colour intensity values for the profiles are computed using fast B-Spline interpolation as proposed by [Unser *et al.* 1991].

5.2.1 Direction Assignment

For later steps it is important to assign a robust direction to the line segment. The direction of the line segment has to be invariant to image rotation and view point changes. Let Ψ^R and Ψ^L be the colour profiles along the right and along the left side of a directed line segment. The direction of the segment is determined by the mean intensity along the profiles: the brighter profile is set to be on the left (figure 5.1). This simple criteria allows to determine the direction consistently over multiple views for the vast majority of cases.

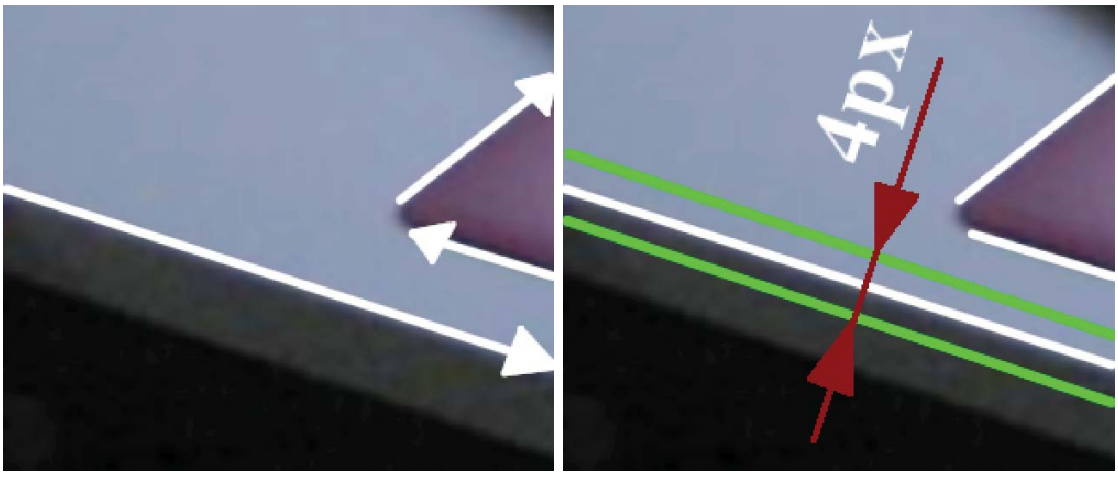

Figure 5.1: Left: The lines are directed so that the darker neighbourhood is on the right hand side. Right: The colour profiles (green lines) are separated by 4 pixels.

5.2.2 Colour Histogram

A direct comparison of the colour intensity profiles, as in [Tell and Carlsson 2000], is not possible because of the inaccurate localisation of the end points along the line segment, and the projective distortion of the profiles. Therefore, we use two colour histograms of Ψ^R and Ψ^L as descriptor of a given line segment, as they are more robust to these factors (although not invariant either). Histograms represent the colour distribution rather than the spatial distribution of a colour profile. We define a colour histogram $\mathbf{h}$ of a profile $\Psi^{R|L}$ as the vector $[h_1, \ldots, h_M]$ in which each bin h_m contains the number of pixels of $\Psi^{R|L}$ having a certain colour m, normalised by the length N of the line segment in pixels.

$$h_m = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} 1 & \text{if } \Psi^{R|L}(i) = m \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

5.2.3 Colour Space Quantisation

Colour histograms considering all possible colours for a 32-bit colour representation would be very big and increase the computational time for the matching considerably. Furthermore, they are overly sensitive to illumination changes as well as to noise. Thus, we reduce the number of colours by applying a crude colour space quantisation, using a predefined palette of colours. Let C be a colour space and $P = \{c_1, c_2, \ldots, c_i, \ldots, c_n \mid c_i \in C, n \ll \| C \|\}$ the quantisation space. The quantiser Q is the function that maps every colour in C to an element, also called bin, in P.

The choice of the quantisation space is crucial for the quality of the candidate matches. The goal is to achieve a certain invariance towards changes in illumination intensity, while being still distinctive enough for the characterisation of the colour profiles. We tried the normalised RGB colour space, as it is insensitive to changes in illumination intensity. However, is considers gray levels, black, and white equally. This is dangerous as grayish-looking colours are widely present in architectural interiors. Other colour spaces, like CIELab and XYZ, are not invariant to illumination intensity. Therefore, we chose the approach proposed by Smith and Chang [Smith and Chang 1995] who partitioned the HSV colour space into 166 bins, placing more importance on the

Hue channel (H) than on Saturation (S) and Value (V). The Hue channel is the most useful as it is invariant to changes in illumination intensity. However, in order to distinguish between greyish looking colours, which are frequent in man-made environments, the Saturation and Value channel should also be used to some extent. Therefore, we divide the channels, as suggested by Smith and Chang, into 18 bins for Hue, 3 for Saturation, and 3 for Value. Additionally, 4 grey levels including black and white are defined (see figure 5.3).

In order to create the histograms, the colour profiles in HSV colour space are not discretely quantised and attributed to a single bin, as it is the case for the example in figure 5.2 top right. For a better invariance towards illumination changes and also for more accurate description, a given colour in the profile is distributed to its eight neighbouring bins in the quantised colour space. The

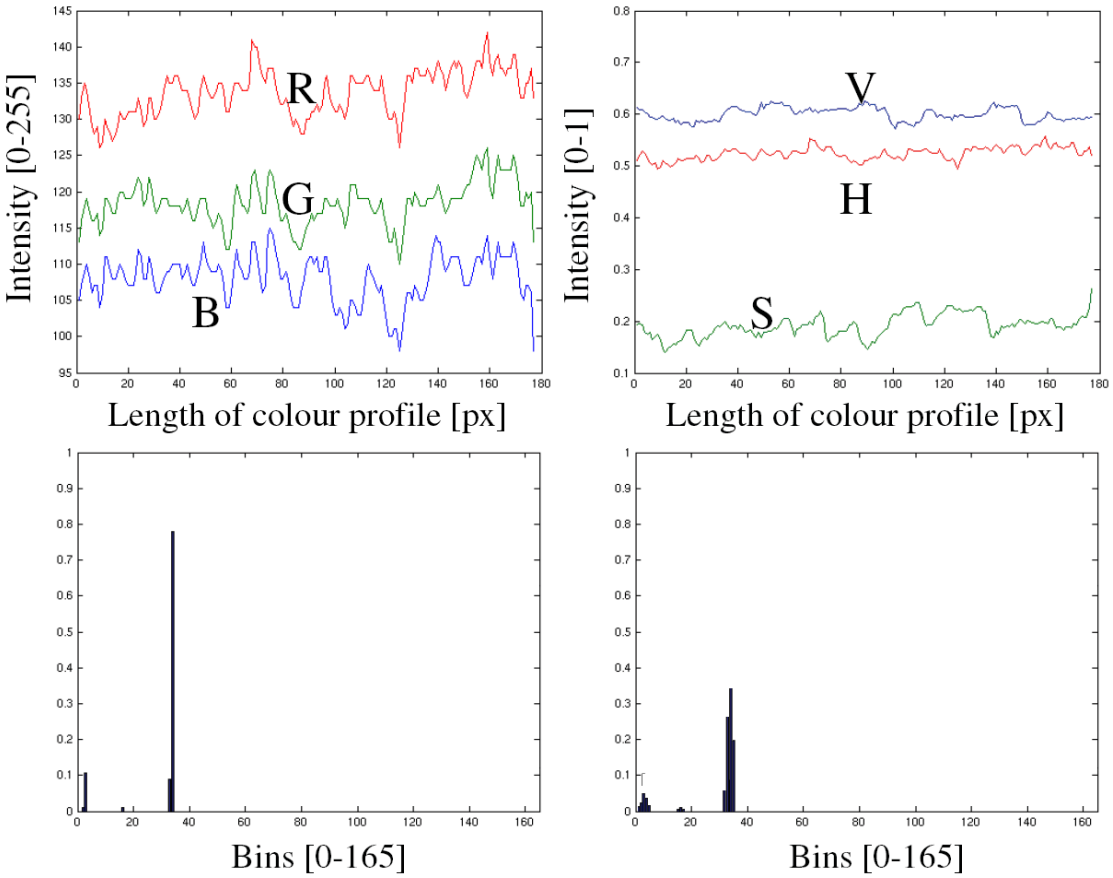

Figure 5.2: Example of a colour profile $\Psi^{R|L}$ along one side of a line segment. Top left: The raw RGB values. Top right: The same profile in the HSV colour space. Bottom left: The profile as unweighted histogram. Bottom right: The profile as weighted histogram.

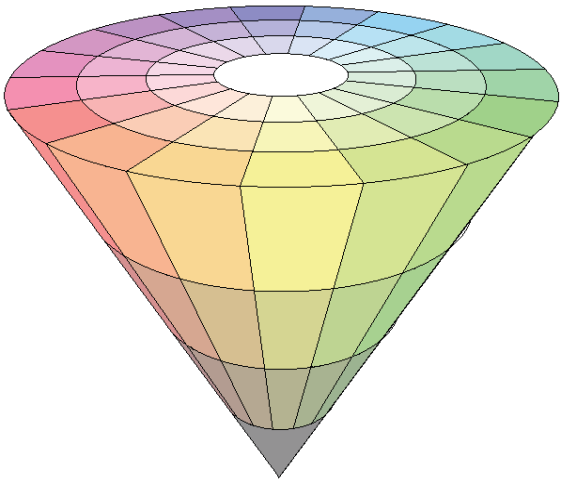

Figure 5.3: The quantised HSV cone. 18 subdivisions for Hue, 3 for Saturation, 3 for Value and 4 greyvalues in the centre of the cone.

colour's contributions to the 8 bins are determined as follows. For reasons of simplicity, lets suppose a quantised, one-dimensional colour space with a total number of two bins: p_1 for the colour $c = 0$ and p_2 for the colour $c = 1$. Suppose we have a colour $c_1 = 0.4$ to be quantised. Traditionally, the colour would be attributed to the closest bin, p_1 for this specific case. Instead, we distribute the colour to both bins p_1 and p_2 according to the proximity of the corresponding bins. As c_1 is closer to p_1 than p_2, this yields a value of 0.6 for p_1 and 0.4 for p_2, respectively. Therefore, c_1 creates the histogram $\mathbf{h_1} = [0.6,\ 0.4]$ instead of $\mathbf{h_1}' = [1,\ 0]$ with the traditional approach. Suppose we also have a second colour c_2 with the value 0.6. This yields $\mathbf{h_2} = [0.4,\ 0.6]$ instead of $\mathbf{h_2}' = [0,\ 1]$. We know that the two colours c_1 and c_2 are only separated by 0.2 units. Even without any knowledge of histogram distance metrics, it is straightforward to see that the histograms $\mathbf{h_1}$ and $\mathbf{h_2}$ are closer to each other than $\mathbf{h_1}'$ and $\mathbf{h_2}'$. Therefore, this binning strategy is expressing the colour distribution more accurately.

5.2.4 Histogram Distance Metric

The difference between two line segments is determined by measuring the similarity of their profile histograms. There are various histogram similarity measures with different properties. They can be divided in two types: The *bin-by-bin* and the *cross-bin* dissimilarity measures. The first type compares only pairs

of bins that have the same index. The Minkowski-form distance d_{L_p} is probably the most used bin-by-bin dissimilarity measure. This distance between histograms $\mathbf{h_1}$ and $\mathbf{h_2}$ is given by

$$d_{L_p} = \left(\sum_{\mathbf{i}} \mid h_{1,\mathbf{i}} - h_{2,\mathbf{i}} \mid^p \right)^{1/p} . \tag{5.2}$$

The L_1 measure is often used to compute the dissimilarity of colour images [Swain and Ballard 1991], but its reliability is limited because the similarity of neighbouring colours is not considered [Stricker and Orengo 1995]. For instance, the distance between yellow and orange is considered equal to the distance between yellow and blue. In general, this is the drawback of all the bin-by-bin type measures like the histogram intersection [Swain and Ballard 1991], which is the same as the Minkowski distance for $p = 1$ when the areas of the two histograms are equal [Stricker and Orengo 1995]. Also the Kullback-Leibler divergence [Kullback 1959] is not an ideal histogram distance metric as it is non-symmetric and sensitive to histogram binning. When using colour similarity measures within the histogram distance computation, a cross-bin dissimilarity measure is more suitable. This type of measure makes use of the distance between bins with different indices. The quadratic distance [Niblack *et al.* 1993] is such a metric; it considers Euclidean distances between colours and is much faster to compute than the Earth Mover's distance [Rubner *et al.* 2000]. This is an important issue, because we have to calculate hundreds of histogram similarities for every line segment. The quadratic distance $d_{1,2}$ between the histograms $\mathbf{h_1}$ and $\mathbf{h_2}$ is given by

$$d_{1,2} = (\mathbf{h_1} - \mathbf{h_2})^\top \mathtt{A} (\mathbf{h_1} - \mathbf{h_2}), \tag{5.3}$$

where $\mathtt{A} = [a_{i,j}]$ is a 166×166 matrix, and its elements $a_{i,j}$ denote the Euclidean distance between the bins c_i and c_j of the palette P in the colour space C. In our case C is the conical representation of the HSV colour space. Therefore, the distance $a_{i,j}$ between two colours $p_i = [h_i,\, s_i,\, v_i]$ and $p_j = [h_j,\, s_j,\, v_j]$, in the quantised HSV colour space is

$$\begin{aligned} a_{i,j} = & \frac{1}{\sqrt{2}} \Big\{ \left(v_i s_i \cos(h_i) - v_j s_j \cos(h_j)\right)^2 + \\ & \left(v_i s_i \sin(h_i) - v_j s_j \sin(h_j)\right)^2 + \left(v_i - v_j\right)^2 \Big\}^{1/2} \end{aligned} \tag{5.4}$$

The distance of two line segments d_s is expressed by the square root of the mean of the histogram distances for both sides. This distance expresses the dissimilarity of two line segments and lies between 0 and 1. The direction of the line segments (dark side on the right hand side) decides on which side is to be compared to which.

5.3 Collinear Line Merging

Line segments are often not fully extracted, but split into several smaller, more or less collinear line fragments. Such fragmentation can lower the matching score considerably. It is due to changes of the gradient magnitude along the actual line, which results in gaps in the edge response. Thus, the snake [Kass *et al.* 1987, Blake and Isard 1998] extracting the line segments is interrupted at those irregularities and splits the actual line segment into smaller pieces. One expects these pieces to occur regularly along the infinite support line $\mathbf{l}$, with small gaps along the direction of their orientation, showing a similar colour profile histogram, having a similar orientation, and pointing in the same direction. Thus, these expected properties can be explored for a line merging cost function.

[Jonk and Smeulders 1995] proposed a scale- and rotation-invariant approach to cluster line segments on the basis of collinearity. In their paper, they consider the relative angle and the gap size between line segments for the construction of a clustering hierarchy. Given a physical line segment that was split into two

Figure 5.4: Left: Collinear line segment merging without taking the appearance into consideration. Right: Merging with the additional appearance information.

separate line segments $\mathbf{l}^{(1)}$ and $\mathbf{l}^{(2)}$. The authors estimate the probability of $\mathbf{l}^{(1)}$ and $\mathbf{l}^{(2)}$ being collinear based on a metric $\lambda(\mathbf{l}^{(i)}, \mathbf{l})$, first introduced by [Nacken 1993]. The infinite support line $\mathbf{l}$ is obtained by orthogonal regression from the end points of the involved line segments. The metric is written as follows.

$$\lambda(\mathbf{l}^{(i)}, \mathbf{l}) = G_{\sigma_\theta}(\theta^{(i)} - \theta) G_{\sigma_w}(d_\perp^{(i)}), \tag{5.5}$$

where $G_\sigma(x)$ denotes a Gaussian function with standard deviation σ. The angle θ is the orientation of $\mathbf{l}$ and $\theta^{(i)}$ the orientation of $\mathbf{l}^{(i)}$. $d_\perp^{(i)}$ is the orthogonal distance from the mid point of the line $\mathbf{l}^{(i)}$ to $\mathbf{l}$. The standard deviations are experimentally chosen around 10° for σ_θ and 3 pixels for σ_w (for 800×600 images).

This line merging method of [Jonk and Smeulders 1995] connects collinear line segments without considering their appearance. In our case, this is dangerous as a given line segment may be occluded or interrupted by another object, or two line segments may just as well be collinear accidentally while belonging to two different objects (see left side of figure 5.4). Therefore, we extend their method using our appearance-based line segment descriptor as additional information. The line segments $\mathbf{l}^{(i)}$ are represented by their colour histograms $\mathbf{h}^{(i)}$ as described in section 5.2. In that same section, the line segments were given a direction according to the brightness change from one side to the other (the dark side is on the right hand side of the line segment).

Given a set of N line segments in one image. Suppose two line segments $\mathbf{l}^{(i)}$ and $\mathbf{l}^{(j)}$, where $i,\ j \in \{1 \ldots N\} \quad \wedge \quad i \neq j$, build with their outer end points a new line segment $\mathbf{l}$ of length l and have a distance d_{ep} between their inner end points. Therefore, the value for d_{ep} is always smaller than the value for l. The lines are described with colour profile histograms $\mathbf{h}^{(i)}$ and $\mathbf{h}^{(j)}$, respectively. The line merging goodness function $M(\mathbf{l}^{(i)}, \mathbf{l}^{(j)})$ is defined as

$$M(\mathbf{l}^{(i)}, \mathbf{l}^{(j)}) = (1 - \frac{d_{\mathrm{ep}}}{l})(1 - d_{s_{i,j}}) \sqrt{\lambda(\mathbf{l}^{(i)}, \mathbf{l}) \lambda(\mathbf{l}^{(j)}, \mathbf{l})}, \tag{5.6}$$

where $d_{s_{i,j}}$ is the dissimilarity of the two line segments $\mathbf{l}^{(i)}$ and $\mathbf{l}^{(j)}$ determined using their colour profile histograms (see previous section). $\lambda(\mathbf{l}^{(i)}, \mathbf{l})$ is the metric mentioned above, expressing the probability of collinearity between the line segment $\mathbf{l}^{(i)}$ and the infinite support line of the new line segment $\mathbf{l}$ [Nacken 1993]. The line segments $\mathbf{l}^{(i)}$ and $\mathbf{l}^{(j)}$ are merged together, if their directions

are not in opposite, and if the merging goodness $M(\mathbf{l}^{(i)}, \mathbf{l}^{(j)})$ is bigger than a certain threshold. This pairwise clustering step is repeated until convergence. Note that for this kind of clustering, the solutions may differ somewhat depending on the order by which line segments are merged. Therefore, the order in which we proceed depends on the length of the individual line segments. As longer line segments are more reliable detected than shorter ones, we first try to merge long line segments and than gradually decrease to the next smaller size.

The merging process can slightly tilt the resulting line segment with respect to the underlying image gradient. Therefore, the resulting line segment is *snapped* by the latter using orthogonal regression of the closest maximum gradient magnitudes in a band of 4 pixels (800×600 images) along the line segment (2 pixels on each side). Quadratic interpolation is used for sub-pixel accuracy. Finally, remaining small line segments (length < 25 pixels for 800×600 images) are discarded.

On average, the number of line segments is reduced by 28% for the merging step and by another 35% during the filtering of short line segments.

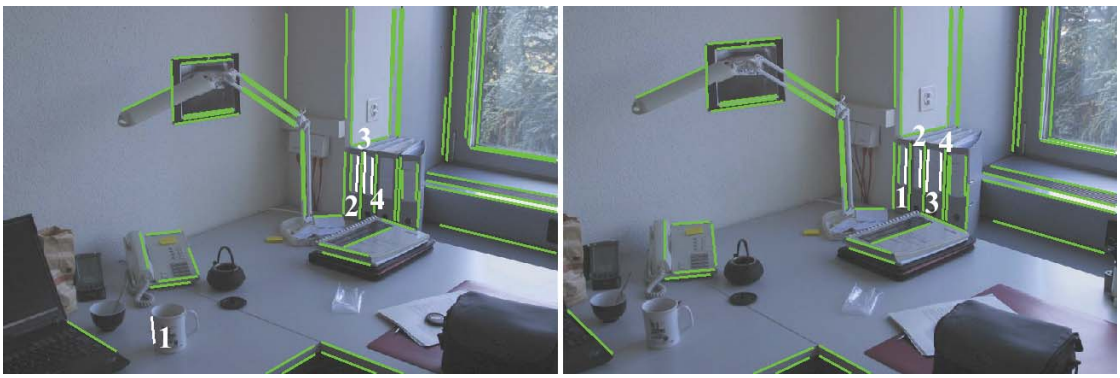

Figure 5.5: Example of matching line segments for the small-baseline case based on appearance only. There are 55 correct matches (green) and 4 mismatches (white).

5.4 Matching

Given two images I_1 and I_2, respectively. In order to establish an initial set of matches, the descriptor of a given line segment $\mathbf{l}_1^{(i)}$ in the first image is compared to all line segments in the second image. A line segment $\mathbf{l}_2^{(j)}$ in I_2

is considered as a candidate match of $\mathbf{l}_1^{(i)}$, if their dissimilarity is below 0.25. Appearance-based matching is often based on a mutually-best strategy. This means that a given line segment $\mathbf{l}_1$ in I_1 matches with a line segment $\mathbf{l}_2$ in I_2 if $\mathbf{l}_2$ is the best candidate of $\mathbf{l}_1$ and vice versa.

We applied this 1-to-1 matching strategy on our line segment descriptors. The results for small-baseline conditions are promising. A simple disparity threshold would suffice to remove the few remaining mismatches for such cases (see figure 5.5). However, the matching performance drops rapidly for wider baselines (see figure 5.6).

For difficult cases, like completely untextured scenes imaged under wide-baseline conditions, the number of mismatches may easily get higher than the number of correct ones (see figures 5.7). Therefore, line segment matching based on appearance alone is not stable enough for poorly-textured scenes that are imaged under wide-baseline conditions.

5.4.1 Soft Matches

We observe for the classic 1-to-1 matching approach a low matching score (see above). First of all, line segments tend to be only weakly distinctive in their appearance, thus several non-corresponding line segments may resemble each other. Second, important changes in the viewing conditions (scale changes, specularities, background changes, etc.) may result in a loss of similarity of

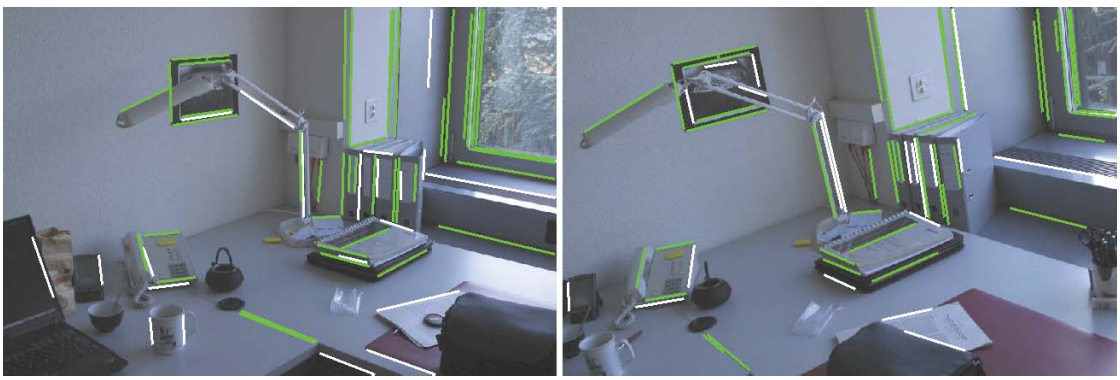

Figure 5.6: A slightly more important view change than the one shown in figure 5.5 results in many more mismatches. There are 40 correct matches (green) vs. 20 mismatches (white)

the correct match. Finally, real scenes often contain repeated or visually similar elements.

Keeping more than just the most similar line segment significantly reduces the chances of missing the correct match. Therefore, for a given line segment $\mathbf{l}_1^{(i)}$, we keep multiple such matches in I_2, so called *soft matches*. We limit the maximum number of soft matches to the 3 candidates with the lowest dissimilarity measure to $\mathbf{l}_1^{(i)}$, if their dissimilarity is below 0.25. Therefore, a given line segment can have 0, 1, 2, or 3 matching candidates.

The dissimilarity threshold of 0.25 was chosen by analysing the maximum dissimilarity of mismatch-free configurations. Considering more than 3 soft matches is dangerous as we observed that the correct match is in most cases among the 3 best candidates and it may lead to erroneous *topological configurations* (explained below) when most line segments are similar to each other.

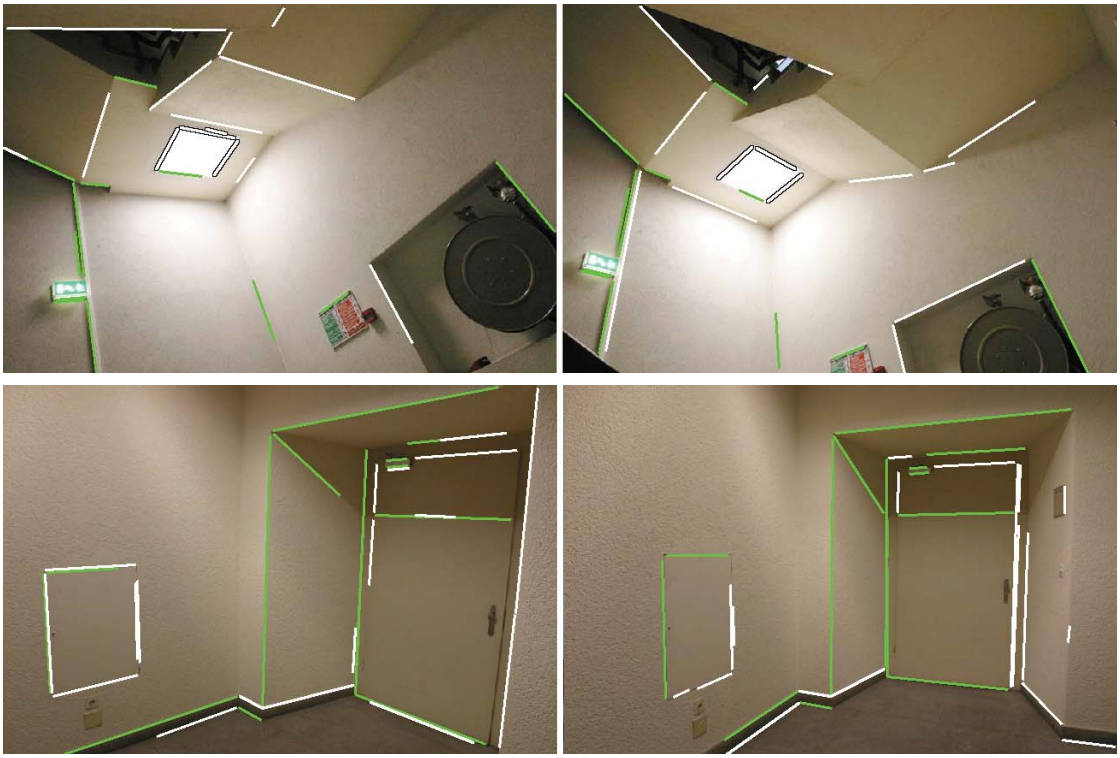

Figure 5.7: Top: Staircase scene with 14 mismatches (white) and 12 correct matches (green). Bottom: Corridor scene with 19 mismatches and 13 correct matches.

5.4.2 Topological Filter

This subsection describes a powerful mismatch filter based on the semi-local spatial arrangement of the features (line segments and interest points) in two views. This is an extension of the topological filter proposed by [Ferrari *et al.* 2003], for interest point triplets. Our filter can handle sets of matches containing only line segments, or both line segments and interest points at the same time. This filtering stage substantially reduces the number of mismatches.

Sidedness Constraint The filter is based on two forms of the *sidedness constraint*. The first form states that, for a triplet of feature matches, the centre of a feature $\mathbf{m}_v^{(1)}$ should lie on the same side of the directed line $\mathbf{l}_v$ going from the centre $\mathbf{m}_v^{(2)}$ of the second feature to the centre $\mathbf{m}_v^{(3)}$ of the third feature, in both views $v \in \{1, 2\}$:

$$\text{side}(\mathbf{l}_1, \mathbf{m}_1^{(1)}) = \text{side}(\mathbf{l}_2, \mathbf{m}_2^{(1)}). \tag{5.7}$$

with

$$\text{side}(\mathbf{l}_v, \mathbf{m}_v^{(1)}) = \text{sign}\left(\mathbf{l}_v \mathbf{m}_v^{(1)}\right) = \text{sign}\left(\left(\mathbf{m}_v^{(2)} \times \mathbf{m}_v^{(3)}\right) \mathbf{m}_v^{(1)}\right) \tag{5.8}$$

Here, points and line segments are used in a homogeneous manner as they both contribute with their centres. The centre of a line segment is defined as its midpoint. The sidedness constraint holds always if the three features are coplanar, and in the vast majority of cases if they are not [Ferrari *et al.* 2003]. As explained in the next subsection, the filter is designed to favour features that respect the constraint in combination with many other feature pairs, while tolerating modest amounts of violations.

The second stage of the filter uses another form of the sidedness constraint. For a *pair* of feature matches, the first being a line segment, the centre of the second feature must be on the same side of the (directed) line segment in both views. Notice that we previously assigned a direction to each line segment (section 5.2.1), so it directly takes up the role of $\mathbf{l}_v$ as in the first form of the constraint. This second stage takes advantage of the fact that a line segment suffices to define a directed line, so the constraint can be verified already for pairs, rather than triplets. Moreover, this second stage uses the whole information provided by the line segment. Figure 5.8 illustrates both forms of the constraint.

These two forms of the sidedness constraint are used each in one of two successive stages (figure 5.9). The first stage uses feature triplets for the verification of the sidedness constraint and eliminates gross mismatches. The second stage, which considers the features pairwise, is very strict towards mismatches. If the second stage was applied directly on an initial set of soft matches, it would get *confused* by the presence of too many mismatches and would eliminate correct matches in turn.

Algorithm A feature triplet or pair including a mismatch is more likely to violate the sidedness constraint. When this happens, it does not tell yet *which* feature is a mismatch. Therefore, for every feature we compute a *violation score* which counts the number of violated constraints for all unordered triplets/pairs including it. A mismatched feature is typically involved in more violations than a correctly matched one. Thus, a match with a high violation score has a higher chance to be incorrect.

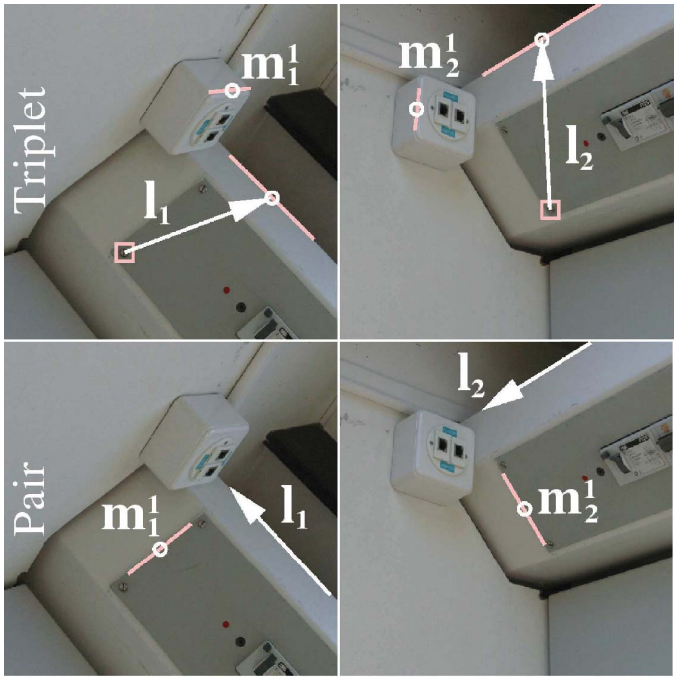

Figure 5.8: Top: a triplet of features (two line segments, with midpoints 'o', and an interest point, marked with a square). $\mathbf{m}^1$ lies on the same side of the directed line $\mathbf{l}$ in both views. Bottom: a pair of line segments, where one considers the side on which the midpoint $\mathbf{m}^1$ lies with respect to the directed line segment $\mathbf{l}$.

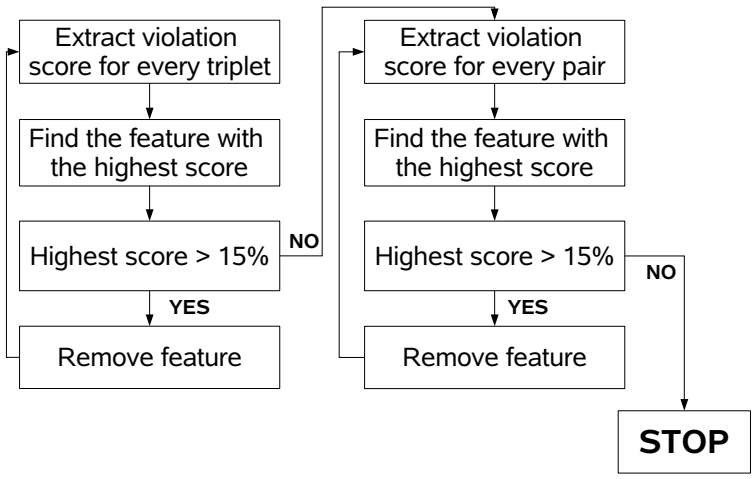

Figure 5.9: Concept of the algorithm for the two stages of the topological filter. The algorithm is similar for both stages and applied consecutively.

The filter algorithm is similar for both stages, which are applied one after the other. Given a set of N feature matches $\mathbf{m}_v^{(i)}$ between views $v \in \{1, 2\}$, with S the number of line segment matches, do:

1. For every feature match $(\mathbf{m}_1^{(i)}, \mathbf{m}_2^{(i)})$, test all sidedness constraints in which it is involved.

2. For each feature, we define a *violation score* V_i by counting the number of violated constraints, divided by the total number in which it is involved. This total number depends on the current stage of the filter. For the triplet stage this is $(N-1)(N-2)/2$, where N evolves over different iterations. For the pair stage, there are three cases: $N-1$ when the feature is a line segment and plays the role of the directed line $\mathbf{l}_v$, or S when the feature is a region, or $S-1$ when it is a line segment playing the second role in the pair. This latter distinction in necessary because of the two different roles a line segment plays in the second stage of the filter: it can provide the directed line, or the centre-point whose sidedness is tested.

3. Find the feature match with the highest violation score V_{max}. If $V_{max} > 0.15$, this match is considered incorrect and removed. Then the algo-

rithm re-iterates from step 1 of the current stage. If $V_{max} \leq 0.15$, or if all matches have been removed, the algorithm terminates.

The threshold 0.15, limiting the permitted overall violation score, was directly adopted from [Ferrari *et al.* 2003] and experimentally verified for the specific case when line segments are considered. This was done on mismatch-free configurations.

After applying the topological filter, there might still be some soft matches left. For further processing, we keep for each feature only the single match with the lowest appearance dissimilarity.

It is important to point out the reasons behind our choice for this filtering strategy. First of all, we cannot compute the fundamental matrix directly from line segment correspondences, and therefore cannot exploit the epipolar constraint as a filter (as opposed to what is done in the region matching literature, e.g. [Matas *et al.* 2002, Mikolajczyk and Schmid 2002, Tuytelaars and Van Gool 2000]). Instead, the topological filter can effectively use the information within the line segment matches to discriminate between correct and incorrect ones.

The second main reason lies in the fact that the proposed filter is insensitive to the exact location of the features, because the number of violated sidedness constraints varies slowly and smoothly for a feature departing from its ideal location. This is particularly important in our context, where the midpoint of a line segment is often inaccurately localised. This ill-localisation is due to the fact that the positions of the end points are not precisely determined. Moreover, since the perspective projection of the midpoint of a line segment in 3D space is not congruent with the midpoint of the projected line segment, the midpoints of two long matched line segments can sometimes represent only an approximative point correspondence.

5.4.3 Correspondence Booster

While [Ferrari *et al.* 2003] use the sidedness constraint only to filter mismatches, we exploit it also for adding new matches.

During the topological filtering, some correct matches are erroneously rejected. Additionally, there might be correct correspondences which have been missed by the initial matcher, because of their high similarity with other line segments.

Based on the second form of the sidedness constraint, such matches can be retrieved and added to the current set. We use this technique to find more line segment matches, but it could also be extended for the generation of new interest point correspondences. However, the use of the epipolar constraint is more appropriate for the generation of new interest point matches.

The algorithm is iterative and starts from the set of matches after applying the topological filter. At each iteration, new line segment matches are added if they respect the topological structure of the current set (i.e. if they have a low violation score). In the next iteration, the updated set of matches serves as reference to calculate the violation scores of potential new matches.

1. For every unmatched line segment in I_1, calculate the violation score of each possible candidate match in I_2, computed with respect to the current configuration of matches. We consider as possible candidates all line segments with an appearance dissimilarity below 0.25. It is advantageous to consider also line segments of I_2 that are already matched, because the match with another line segment may have a lower violation score. The three candidate matches with the lowest violation scores are stored in a waiting room, if this score is below 0.15.

2. All line segment matches in the waiting room are added to the current set of matches

3. The two stages of the topological filter are applied to this extended set of matches. The matches which are rejected are left out of any further consideration. In the case that some line segments still have more than one candidate match, these are eliminated by keeping only the one with the lowest sum of appearance dissimilarity and topological violation score.

4. If the current configuration is the same as at the beginning of this iteration, the algorithm terminates, otherwise it iterates back to the first point.

This method typically substantially increases the number of correct matches. For the example of figure 5.14, the number of matches grows from 21 (16 correct) to 41 (35 correct).

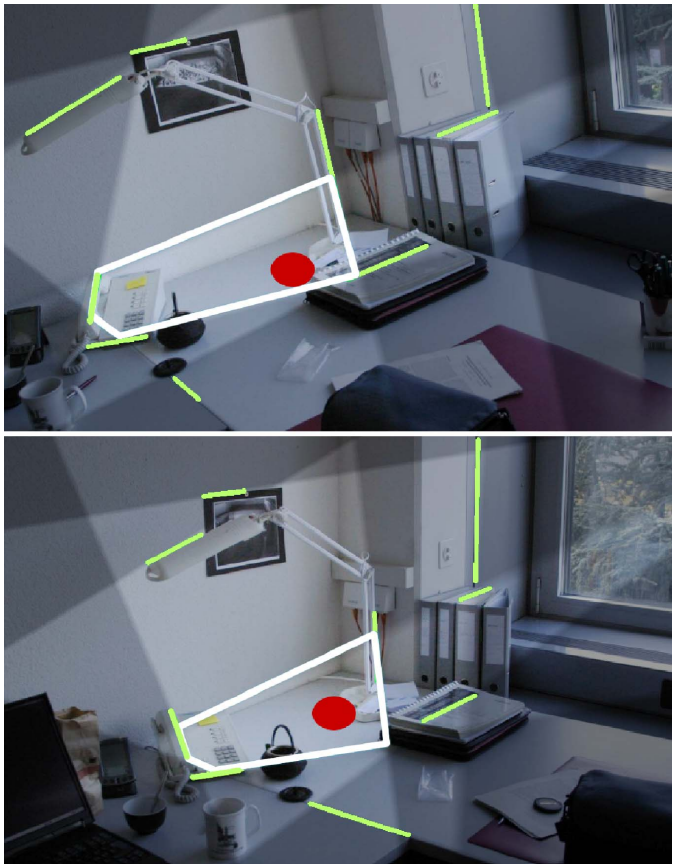

Figure 5.10: The search space for candidate matches of the dot within the highlighted zone in the top image is reduced to the highlighted zone in the bottom image if no sidedness constraint is to be violated.

5.4.4 Vanishing Point Filter

Indoor scenes often contain parallel lines whose perspective projections meet in a *vanishing point*. Suppose we have a bundle of parallel lines in 3D space with a common direction $\mathbf{d}$. That bundle is imaged by a camera with its focal point $\mathbf{C}$ and projection plane $\mathcal{R}$. The projected parallel lines on the projection plane seem to converge at one single point called the 'vanishing point' (see figure 5.11). We can define the vanishing point of a line $\mathbf{l}$ having the orientation $\mathbf{d}$ as the intersection of the image plane $\mathcal{R}$ with the one line having the same direction $\mathbf{d}$ and going through the projection centre $\mathbf{C}$. The vanishing point is the same for all lines which are parallel to $\mathbf{l}$.

In most indoor scenes, one or more distinctive vanishing points are present. They contain important geometrical information about the scene, like about

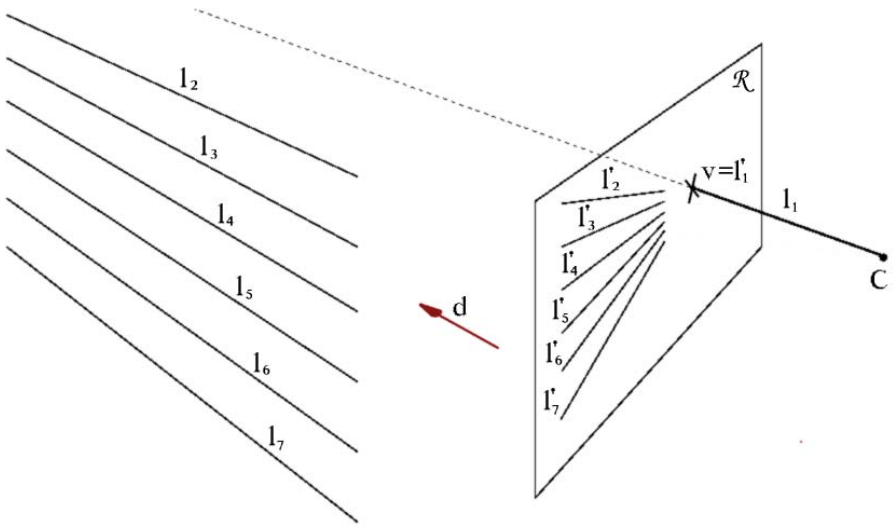

Figure 5.11: Projection of a pencil of parallel lines $\mathbf{l}_1$ ... $\mathbf{l}_7$ with a direction $\mathbf{d}$ on the projection plane $\mathcal{R}$. The line $\mathbf{l}_1$ is going through the projection centre $\mathbf{C}$ and intersects $\mathcal{R}$ in the vanishing point $\mathbf{v}$. In other terms, the vanishing point is the projection of the line with the direction $\mathbf{d}$, passing through the projection centre $\mathbf{C}$

parallel lines, dominant directions and angles, horizon line etc. For environments that are composed of mutually orthogonal planar patches, vanishing points can also be used for applications like camera calibration and 3D reconstruction [Bellutta *et al.* 1989, Cipolla *et al.* 1999, Rother and Carlsson 2002]. Here we use vanishing points to create an additional constraint for the detection of more reliable line segment matches.

Vanishing Point Detection The vanishing points are extracted in both views, I_1 and I_2 simultaneously, using an extended version of the RANSAC-based method proposed by [Rother 2000]. The original approach supposes the range of the camera focal-length to be known in order to use an orthogonality constraint. We changed this constraint in order to deal with non-orthogonal scenes. The orthogonality constraint is built upon the fact that the vanishing points $\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$, and $\mathbf{v}^{(3)}$ for three orthogonal directions build an orthocentric system together with the principal point $\mathbf{c}$ (see figure 5.12). Our adaptation requires that the scene is composed of different wall planes that build a right angle with the floor plane, but they are not necessarily mutually orthogonal. Furthermore, we suppose the principal point to lie in the image centre. Such a configuration allows for the automatic extraction of the vanishing line (line

built by connecting two vanishing points) that belongs to the floor plane, and the estimation of an approximate focal length. Hence, the angles between the vanishing points can be approximated and a new constraint is built that has to hold for I_1 and I_2 simultaneously.

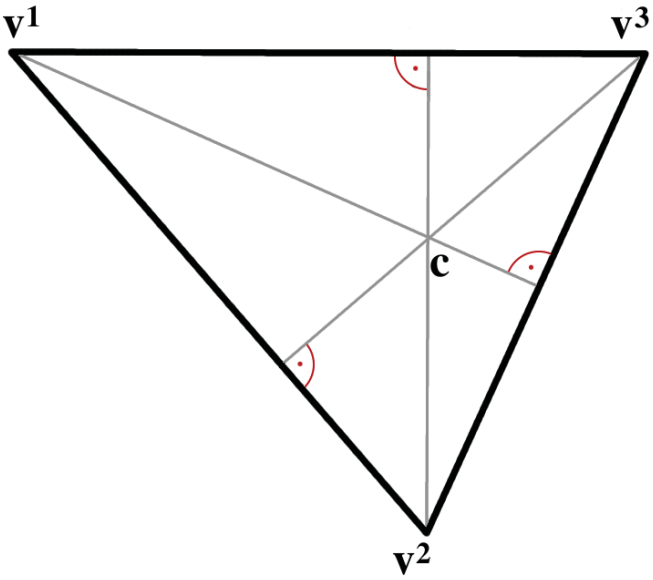

Figure 5.12: The orthogonality constraint for scenes with three mutual orthogonal directions imposes an orthocentric system involving the three vanishing points ($\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$, $\mathbf{v}^{(3)}$) and the principal point $\mathbf{c}$.

In order to verify the orthogonality constraint, we first identify the vanishing point that belongs to the vertical lines. These lines are in general perpendicular to the ground plane (as we suppose walls to be orthogonal to the floor plane). This specific vanishing point is easily identified. Given three vanishing points $\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$, $\mathbf{v}^{(3)}$, and the principal point $\mathbf{c}$ as the centre of the image. If the vanishing point $\mathbf{v}^{(3)}$ corresponds to the vertical direction, and $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$ to two horizontal directions, the lines $\mathbf{s} = \mathbf{v}^{(1)} \times \mathbf{v}^{(2)}$ and $\mathbf{t} = \mathbf{v}^{(3)} \times \mathbf{c}$ must form a right angle. As the principal point lies not always exactly in the image centre, the obtained angle between $\mathbf{s}$ and $\mathbf{t}$ is not perfectly orthogonal. Therefore, we consider the most perpendicular vanishing point as the one belonging to the vertical lines.

The vanishing point detection algorithm proceeds as follows. First, a sample set of 6 line segment matches are randomly selected in order to compute the three vanishing points $\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$, and $\mathbf{v}^{(3)}$ in both images simultaneously. We

suppose the intersection of the first two lines as the vanishing point $\mathbf{v}^{(1)}$ and the intersection of the two remaining line pairs as $\mathbf{v}^{(2)}$ and $\mathbf{v}^{(3)}$, respectively.

Then, the total number of inliers is computed for the 4 different combinations: ($\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$, $\mathbf{v}^{(3)}$), ($\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$), ($\mathbf{v}^{(1)}$, $\mathbf{v}^{(3)}$) and ($\mathbf{v}^{(2)}$, $\mathbf{v}^{(3)}$). An inlier is a line segment match where both line segments have a certain distance d_{vp} to one and the same vanishing point simultaneously. Therefore, the maximum distance of the individual line segments involved in a match is considered. The distance d_{vp} is defined as the separation in pixels of the end points to the line going through the vanishing point and the mid point of the line segment, see figure 5.13. This distance has been chosen to be below 5 pixels for 800×600 images. It was evaluated based on the distances of manually selected lines to their corresponding vanishing points.

Finally, the best solution is stored if it fulfils the modified orthogonality constraint. All steps are repeated N times with

$$N = \log(1-p)/\log(1-(1-\epsilon)^s), \tag{5.9}$$

where s is the number of samples, in our case the 6 line segment matches. The parameter p is the probability that at least one of the random samples of s is free from outliers. Usually p is chosen at 0.99 [Hartley and Zisserman 2004]. The parameter ϵ is the proportion of outliers $\epsilon = 1-(\mathrm{numberofinliers})/(\mathrm{totalnumberoflineseg}$ computed at each iteration.

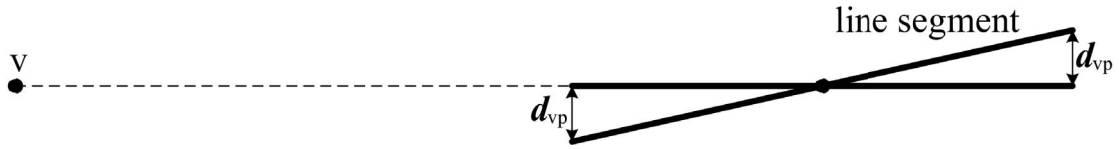

Figure 5.13: The distance d_{vp} of a line segment to a vanishing point.

In the case when three vanishing points are detected, the procedure is repeated with the remaining lines (if available) in order to find more than just three vanishing points. This is done because we assume that for some architectural scenes, the walls are not mutually orthogonal and may provide more than just two horizontal vanishing points. If this is the case, the vanishing line of the floor plane, needed for the verification of the orthogonality constraint, is more easily identified than with only three vanishing points. When less than three vanishing points are detected, the orthogonality constraint cannot be verified.

Once our method has terminated, we end up with a set of matched vanishing points. Unmatched and conflicting line segments are considered for each image separately. If their distance d_{vp} to a certain vanishing point is below a threshold (again 5 pixels), they are labelled with the corresponding vanishing point number. It is possible that a line segment votes for no vanishing point. Such line segments are assigned to a virtual one (not existing).

Filter Algorithm The idea of the vanishing point filter is very easy. Given a subset $\mathcal{L}_{\mathbf{v}_1}$ of line segments in image I_1 intersecting in a vanishing point $\mathbf{v}_1$. The corresponding set $\mathcal{L}_{\mathbf{v}_2}$ of matched line segments in image I_2 should hence intersect at the corresponding vanishing point $\mathbf{v}_2$. Line segment pairs that vote for different vanishing points are considered as mismatches. However, it occurs that a line lies on or very close to a vanishing line. Such a line may equally vote for either of the two vanishing points which build the vanishing line. Therefore, it is possible that the lines involved in a correct match vote for opposite vanishing points. Discarding such cases would lead to falsely-rejected, correct matches. The *genuine* vanishing point is determined as follows. The maximum distance d_{vp} of the lines to each of the candidate vanishing points is determined. If this distance is smaller than the threshold used for the determination of the inliers (5 pixels for 800×600 images), the involved lines are attributed to the vanishing point opposite to the one for which d_{vp} is maximum. The metric used for the distance measure is the same as defined in [Rother 2000] and shown in figure 5.13.

The correspondence booster from section 5.4.3 is extended as follows. If an unmatched line segment in I_1 belongs to a matched vanishing point $\mathbf{v}_1$, the candidate match is supposed to belong to the corresponding vanishing point $\mathbf{v}_2$ in I_2. Therefore, the search space is limited to line segments voting for $\mathbf{v}_2$. All in all, three conditions have to be fulfilled for a line pair to be considered a potential match: First, a dissimilarity lower than 0.25. Secondly, a violation score smaller than 15% in the overall topological configuration. Thirdly, the involved lines have to vote for the same vanishing point correspondence. In general, we obtained 10% more matches while reducing the number of mismatches with about 20% compared to the original booster.

5.5 Results

We report results on 3 example scenes, imaged under wide-baseline conditions. For the detection and description of the interest points, we used our SURF features at a moderate threshold (300). All experiments have been made with the same thresholds as specified (15% for the maximum violation score and 0.25 for the dissimilarity). The average computation time for the matcher was 4 seconds on a modest workstation (Pentium IV at 3 GHz, line detection included). The examples were chosen such that they show different characteristics of the individual matching stages.

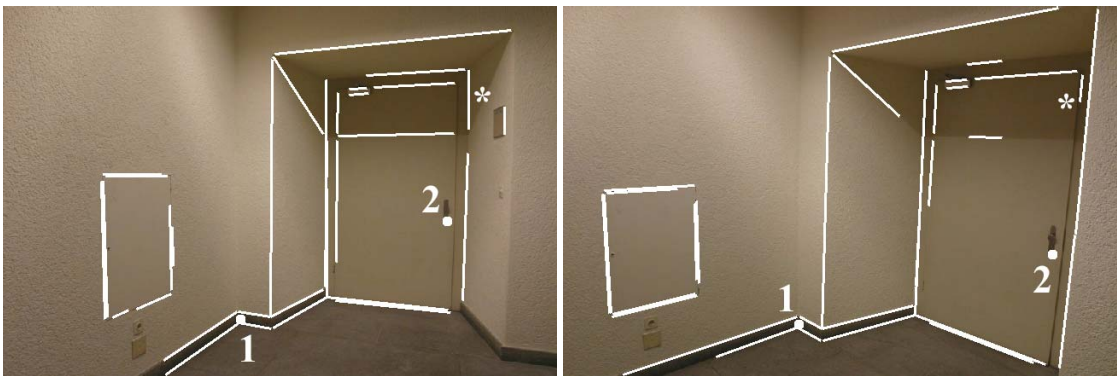

Figure 5.14: Corridor scene where our matcher, including booster, produced 31 geometrically correct matches and 1 mismatch (marked '*'). Moreover, two interest points were detected and correctly matched (numbers 1 and 2) using SURF.

The first scene depicts the corner of a corridor, and has almost no texture (figure 5.14). This example shows a typical property of the boosting stage. Instead of finding more matches, it reduces the number of mismatches while leaving the number of correct matches constant. Our matcher finds 33 matches after the topological filter (30 correct). After the boosting stage, one mismatch is removed and another one after adding the vanishing point filter and booster. The final number of line segment correspondences is 31. Notice that the influence of interest point correspondences is negligible (2 correct matches).

Figure 5.15 presents another texture-less scene, consisting of a corner of an office ceiling. Besides the viewpoint change, an additional challenge is posed by the strong in-plane camera rotation. For this example, the booster stage shows its convergence stability. Our method produces only 3 matches (all incorrect) after the topological filter. The booster finds a total of 25 matches (20

Figure 5.15: Corner scene. After the different matching stages, we end up with 23 correct matches and 1 mismatch (marked '*'). SURF identified one interest point correspondence.

correct). After introducing the vanishing points, 23 matches were found and only one mismatch remained. This example shows how the last two stages can substantially boost the number of correct matches, while reducing the number of mismatches. Only one interest point match is found for this scene.

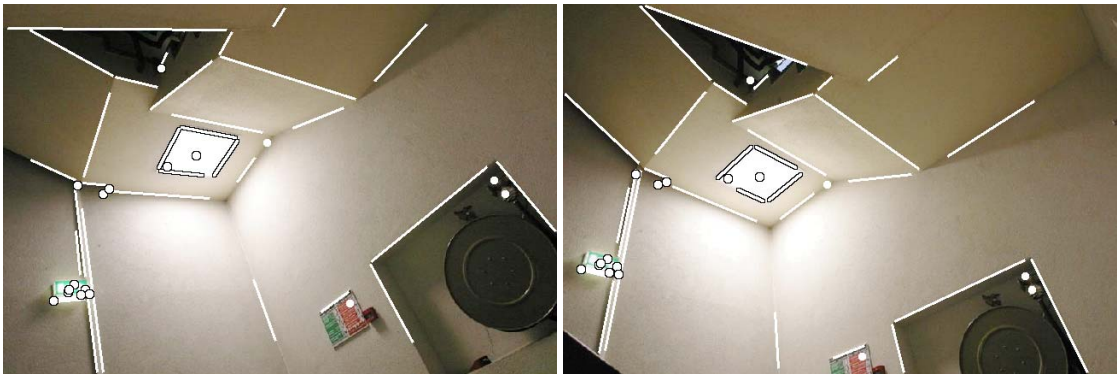

Figure 5.16: Staircase scene. The booster produced 31 geometrically correct matches and no mismatch. Moreover, 17 correct interest point matches were identified using SURF.

Figure 5.16 shows the last scene, demonstrating the benefits of considering line segments and interest points at the same time to verify the topological constraints. The interest point matcher (SURF) yields 17 matches (17 correct). By running our method including the interest point matches, we obtain after the topological filter 28 line segment matches (17 correct). The boosting stage ends up with 31 correct matches and not a single mismatch. These numbers remain constant after the vanishing point filter. When considering the

line segments alone, we end up with only 26 line segment matches, where 1 is a mismatch. Notice that for the other scenes, the number of detected interest points is not important enough to influence the outcome of the individual matching stages. Classical, point-oriented stereo matching methods would return basically empty-handed.

5.6 Conclusion and Outlook

In this chapter, we presented an appearance-based descriptor for line segments. It considers the colour neighbourhood along the line segments using histograms instead of the raw intensity values. We used the descriptor in a first step to enhance an established line merging scheme. Instead of *blindly* merging collinear line segments, the appearance of the involved segments is considered. As a result, the number of candidate matches for a given line segment is reduced. This on the other hand reduces the number of ambiguities significantly.

The classic 1-to-1 approach often fails for images taken under wide-baseline conditions. Therefore, we have presented a new wide-baseline stereo matcher capable of working with line segments alone, or combined with interest point matches. The experiments support the approach and show that it can find a good number of line segment correspondences also on untextured scenes, imaged under general wide-baseline conditions, and without any prior knowledge about the scene or camera positions (e.g.: in contrast to [Schmid and Zisserman 1997]). Moreover, the system can effectively exploit the complementary information provided by line segments and interest points, and hence can handle a wider range of scenes than is possible with either cue alone.

Still an open issue is the development of a performant line segment detector. Current approaches often fail to robustly detect the same line segment in two images taken under only slightly different conditions (viewpoint, image-rotation, scale, lighting). As for the descriptors, future work could focus on using the gradients along the line segments as additional information. Moreover, the colour quantiser could be chosen based on an initial global analysis of the colour distribution in the two images. This would probably increase the quality of the initial set of candidate matches.

A potential improvement to the matcher could be to apply a stage of finding more matches also for interest points, or to extend it to take advantage of the

epipolar constraint once the fundamental matrix has been computed. Moreover, soft matches could be considered also for the interest points based on their respective distance in descriptor space. The three closest candidates could then be considered to build the topological constraints.

Future work could focus on applying a similar scheme for matching curves. We made some promising tests for camera self-calibration from curves in two views.

III

from Points and Lines

Outline

In the previous two parts, we tackled the problem of feature correspondences (interest points and line segments) between two images and showed some practical applications for interest point correspondences. Another important application for feature matches between two or more images is camera self-calibration and 3D reconstruction (structure from motion). The accuracy of the camera calibration is crucial for the quality of the 3D model. It depends on the constraints imposed on the cameras, the number of images, the number of correct matches between those images, and the accuracy in terms of localisation of the interest points.

This part of the thesis has two goals. The first one is to test our SURF features for tasks related to structure from motion and compare the accuracy of the Fast-Hessian detector against other scale and rotation invariant interest point detectors for the two-view case. Furthermore, the SURF detector/descriptor scheme is integrated into a professional 3D-reconstruction tool in order to handle the wide-baseline case for more than two views. This tool automatically calibrates the cameras and estimates a dense 3D model of the imaged scene. The accuracy of the obtained dense 3D model serves as a qualitative evaluation for the accuracy of the Fast-Hessian interest points.

The use of interest points requires a high amount of texture in the imaged scene in order to detect enough features that support the estimation of the epipolar geometry. Man-made environments are typically scenes where not enough interest points are detected because of lacking texture. However, line segments may still be abundantly detectable. Therefore, the second goal of this part is to solve the structure-from-motion problem for line segment correspondences between two views in a practical sense. In general, line segments are inaccurately and incompletely detected (as discussed in the previous part). Furthermore, there is no geometrical constraint for line segment matches between only two views. Hence, structure-from-motion from line segments in two views is an extremely

challenging task and not solvable for the general case. Therefore, we suppose the scene to be piecewise planar as it is often the case for architectural indoor scenes. Based on this assumption, we propose a novel approach to detect planar intersections (junctions) of line segment correspondences. These are point correspondences and can be used to estimate the epipolar geometry and therefore to calibrate the cameras.

In chapter 6, we introduce the theoretical basis for the structure-from-motion problem from two images. Furthermore, the SURF features are tested quantitatively for the two-view case and qualitatively for the N-view case. The images are supposed to be taken under wide-baseline conditions. Then, the problem of camera self-calibration, bundle adjustment and 3D reconstruction from line segments in two images is addressed in chapter 7. There, we propose a novel method to identify polyhedral junctions resulting from line segment intersections. At the same time, the images are segmented into planar polygons. This is done using an algorithm based on a Binary Space Partitioning (BSP) tree. The junctions are matched end points of the detected line segments and hence can be used to obtain the epipolar geometry. The essential matrix is considered for metric camera calibration. A bundle adjustment is performed on line segments and the camera parameters. Prior to this, the number of unknowns is reduced by a maximum flow algorithm for better stability. Finally, a piecewise-planar 3D reconstruction is computed based on the segmentation of the BSP tree. The system's performance is tested on some challenging examples. In this part, we also discuss the advantage of using line segment and point correspondences in a unified manner.

6

From Point Matches to 3D

Finding the 3D locations of individual point correspondences between images of a static scene is a well studied problem and many different methods for self-calibration exist (see [Hartley and Zisserman 2004]). The choice of a particular method depends on the number of given input images and the constraints on the camera. There are mainly two types of constraints: a parameter is either known, or it is fixed across views, but its value is unknown. Notice that the images have to be taken under specific conditions in order to fulfil the desired constraints. For more detailed information about such cases, we refer the reader to [Hartley and Zisserman 2004].

Whatever method is chosen, a typical part is the estimation of the epipolar geometry. It geometrically connects two or more views and is computed based on image correspondences, for this chapter these are interest point matches. Once the epipolar geometry is known, it can eventually be used, depending on the camera constraints, to derive some of the *internal* or *external* camera parameters. With calibrated cameras, the interest points are projected in 3D space using the technique of triangulation. The resulting 3D structure is by far not a complete model of the scene. The construction of a complete dense model from two or more images without any a priori knowledge is, even with many image correspondences and perfectly calibrated cameras, a sophisticated problem. The difficulty consists in finding the specific location in 3D space for every visible pixel in a reference image, also for homogeneous regions [Strecha *et al.* 2003, Pollefeys *et al.* 2004, Strecha *et al.* 2004, Cornelis and Van Gool 2005]. However, the accuracy of the camera calibration and the initial point model strongly influence the quality of the dense 3D model.

Here, we address the theoretical part of the two-view case as we present some quantitative results of SURF for this specific case. Furthermore, it serves as the theoretical basis for the next chapter, where the challenging problem of structure-from-motion from line segment correspondences between two views is addressed.

This chapter is organised as follows. First, we introduce the projective camera model. Then, we discuss the constraints imposed on the cameras. Afterwards, we continue with the estimation of the epipolar geometry, the camera calibration, and the 3D reconstruction of individual interest point correspondences. Then, the performance of SURF for tasks related to camera self-calibration and 3D reconstruction is demonstrated by means of two different evaluation approaches. The first one considers two views of a known scene. These are matched using different interest point detection schemes and the respective cameras are calibrated. The resulting 3D points are then compared to their 'real' location yielding a quantitative evaluation of different interest point detectors. The second approach is based on the N-view case. The SURF features are integrated in a professional 3D-reconstruction tool in order to handle the wide-baseline case. The accuracy of the obtained dense 3D model serves as a qualitative evaluation.

6.1 The Projective Camera Model

Most cameras can be modelled using the pinhole-camera model. It consists of a retinal plane $\mathcal{R}$ (image) and a projection centre $\mathbf{C}$ (focal point). The orthogonal projection of $\mathbf{C}$ on $\mathcal{R}$ is defined as the principal point $\mathbf{c}$. The distance between the projection centre and the principal point is called the focal length f. The projection of a point $\mathbf{M}$ in the scene is defined as the intersection of the retinal plane $\mathcal{R}$ and the line passing trough $\mathbf{M}$ and the projection centre $\mathbf{C}$ (see figure 6.1).

Considering the projection $\mathbf{m} = (x, y)^\top$ of the scene point $\mathbf{M} = (X, Y, Z)^\top$ with the focal length $f = 1$, we can write the following relation, using the homogeneous representation of point coordinates, if the world coordinate system is aligned with the camera, as shown in figure 6.1.

$$\mathbf{m} \sim [\mathtt{I} \mid \mathbf{0}] \tag{6.1}$$

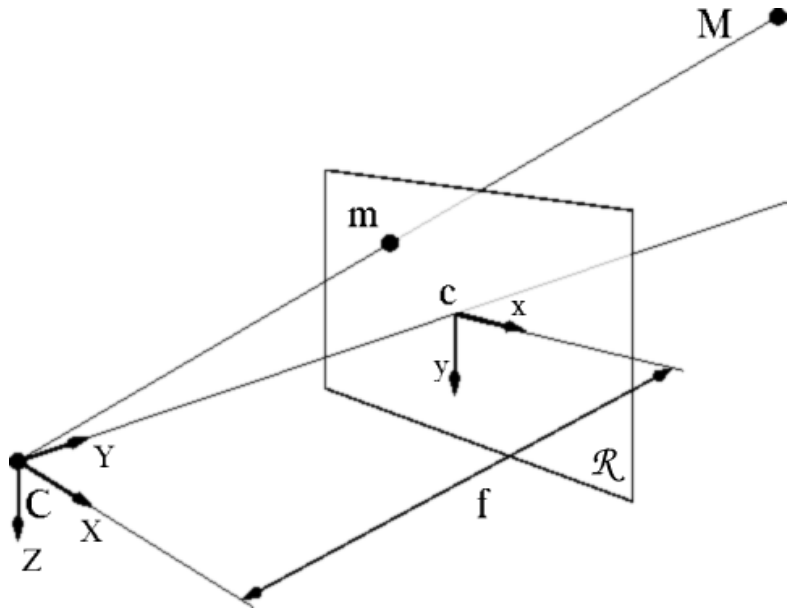

Figure 6.1: Perspective projection with pinhole-camera model. The scene point $\mathbf{M}$ is projected on the retinal plane $\mathcal{R}$ and results in the image point $\mathbf{m}$.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{6.2}$$

The matrix I denotes the identity matrix and $\mathbf{0}$ is a zero vector. Notice that the sign $\sim$ means equal up to a scalar that is usually equal to the inverse of the third component of $\mathbf{m}$. Thus, the coordinates of the projected point $\mathbf{m}$ can be written as $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$.

In practice, a real camera has a focal length different from 1 and the pixels of its chip are not always perfectly square. Furthermore, the principal point of the camera is typically not used as the origin. Also, image coordinates are expressed in pixels, not the metrical units of x and y. Therefore, we need a transformation matrix that transforms the coordinates $(x,\, y)^\top$ into pixel coordinates in the image. This matrix is called the calibration matrix K of the camera.

Lets call p_x and p_y the width respectively the height of a pixel on the chip in a metric unit, α the skew angle in radians (see figure 6.2), and $\mathbf{c} = (c_x,\, c_y)^\top$ the coordinates of the principal point in pixels with respect to the origin of the image. Hence, we can write $f_x = \frac{f}{p_x}$ respectively $f_y = \frac{f}{p_y}$ for the focal length

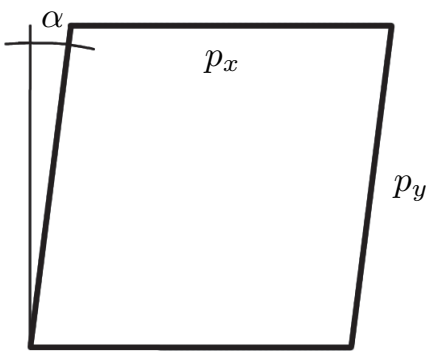

Figure 6.2: The skew angle α, the height p_y, and the width p_x of a pixel.

in pixels, and $s = \tan(\alpha) f_y$ the skew, also in pixels. The general form of the calibration matrix, representing the *internal* camera parameters, can therefore be written as follows.

$$\mathtt{K} = \begin{bmatrix} f_x & s & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix} \tag{6.3}$$

In order to handle the two-view case, we impose constraints on the internal camera parameters (one could also impose constraints on the scene, like orthogonality, known point locations in 3D space, etc.). For most modern digital cameras, the skew parameter is very close to zero. Calibration algorithms using calibration grids often attempt to ensure that $s = 0$, sometimes at the cost of solving a nonlinear optimisation problem [Faugeras 1993].

Furthermore, it has become well established that self-calibration of the principal point $\mathbf{c}$ is difficult and only necessary if the reconstruction has to be very accurate. The 3D reconstruction error after *triangulation*, as defined in [Hartley and Sturm 1995], is not dramatically affected by errors in the assumption that the principal point remains fixed at the centre of the images.

The aspect ratio ($\epsilon = f_x / f_y$), which is often assumed 1, is hardly affected by focus and zoom changes. It is possible to calibrate for this parameter once, and reliably use the value found at any other occasion.

Therefore, for the two-view case, we assume the skew to be zero, the aspect ratio known, and the principal point in the centre of the image. Furthermore,

we suppose the focal length f to be equal for both views, and the images to be taken with the same camera. Hence, the calibration matrix for all views is defined as follows.

$$\mathtt{K} = \begin{bmatrix} \epsilon f & 0 & c_x \\ & f & c_y \\ & & 1 \end{bmatrix} \tag{6.4}$$

We denote $\mathtt{P}$ the projection matrix of a general projective camera that maps scene points $\mathbf{X}$ to image points $\mathbf{x}$, where $\mathbf{x} \sim \mathtt{P}\mathbf{X}$.

As we deal with two input images I_1 and I_2, two projection matrices are required and denoted as $\mathtt{P}_1$ and $\mathtt{P}_2$, respectively. For the first image, we consider the projection centre $\mathbf{C}$ of the camera in the origin of the world coordinate system and the axes thereof aligned with those of the camera. Therefore, the metric projection matrix of the first camera is $\mathtt{P}_1 = \mathtt{K}[\mathtt{I} \mid \mathbf{0}]$. The second image has to be taken from a different position than the first one. This camera transformation is expressed as a rotation $\mathtt{R}$ followed by a translation $\mathbf{t}$. Thus, the metric projection matrix $\mathtt{P}_2$ for the second image is written as follows.

$$\boxed{\mathtt{P}_2 = \mathtt{K}[\mathtt{R} \mid \mathbf{t}]} \tag{6.5}$$

The translation vector $\mathbf{t}$ is equal to the position of the projection centre of the camera in the world coordinate system. For a metric 3D reconstruction, the focal length f, the rotation matrix $\mathtt{R}$ and the translation vector $\mathbf{t}$ remain to be determined. This is possible through the estimation of the epipolar geometry.

6.2 Epipolar Geometry

The back-projection of an image point $\mathbf{x}$ from one image into 3D space has an infinite number of solutions. The corresponding scene point lies somewhere on a straight line that goes through the projection centre $\mathbf{C}$ of the camera and the image point $\mathbf{x}$ on the retinal plane $\mathcal{R}$. This line is called the projection ray $\mathbf{l}_p$ of the scene point $\mathbf{X}$. In order to define the exact location of $\mathbf{X}$ on $\mathbf{l}_p$, we need a second image of that same point from a different position. Two views of the same rigid scene taken from different view points are related through the epipolar geometry. It can be calculated from point correspondences between at least two views, or from line correspondences between at least three views.

The new projection centre $\mathbf{C}'$ and the projection ray $\mathbf{l}_p$ build a plane called the epipolar plane, denoted Π_e (see figure 6.3). The epipolar plane intersects with the image planes $\mathcal{R}$ and $\mathcal{R}'$ to form lines, the so-called epipolar lines $\mathbf{l}_e$ and $\mathbf{l}'_e$. The "real" scene point $\mathbf{X}$ lies somewhere on the projection ray $\mathbf{l}_p$. Thus, its correspondence in the second image $\mathcal{R}'$ has to lie somewhere on the epipolar line $\mathbf{l}'_e$. The line defined by the two camera centres $\mathbf{C}$ and $\mathbf{C}'$ is called the baseline.

Multiple scene points create multiple epipolar planes as illustrated in figure 6.4 and thus, multiple epipolar lines. They intersect for each image in one specific point that is called *epipole* and denoted $\mathbf{e}$ and $\mathbf{e}'$, respectively. The epipoles are the projections of the camera centre $\mathbf{C}$ in I_2 and $\mathbf{C}'$ in I_1, respectively.

With the 3D reconstruction of image points in mind, one can easily see that points located somewhere on the baseline are, even for widely separated views, not defined in space. In fact, image points located close to the respective epipoles are difficult to reconstruct. It also follows that images with the same projection centre (zero translation) contain no spatial information, even if the rotation is important.

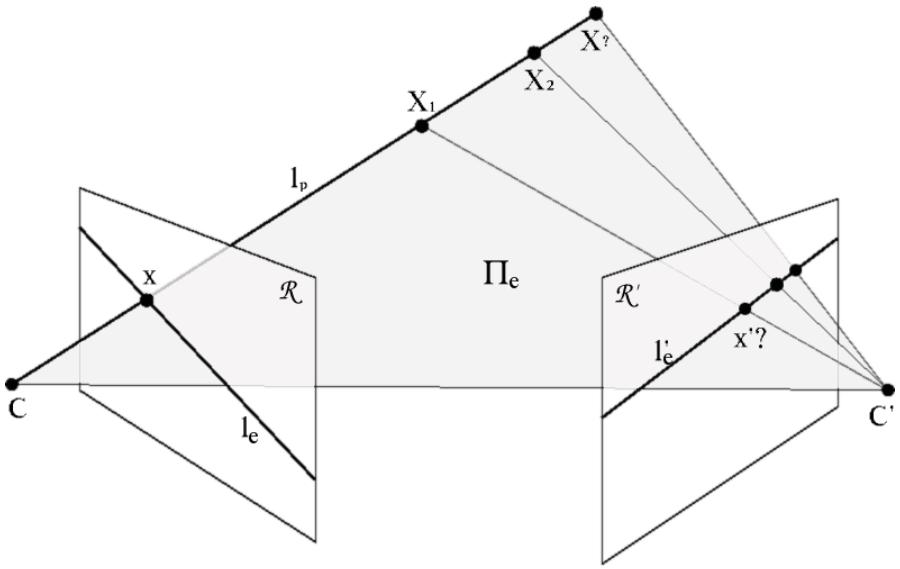

Figure 6.3: The correspondence of the image point $\mathbf{x}$ in the second image has to lie somewhere on the epipolar line $\mathbf{l}'_e$. For an accurate estimation of the scene point $\mathbf{X}$, its projection in the second image has to be known

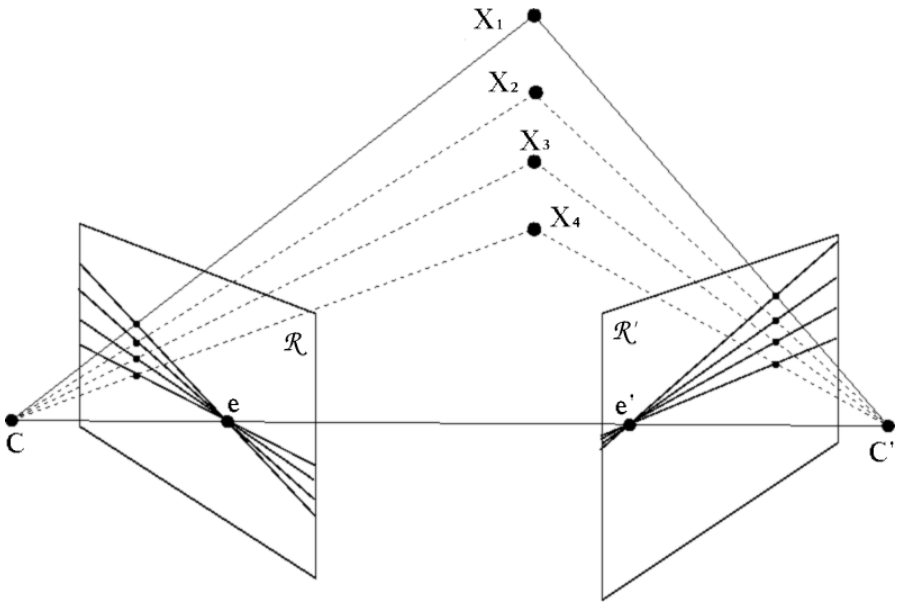

Figure 6.4: Multiple points in space create multiple epipolar planes and a pencil of epipolar lines centred in the epipole. The intersections between the different epipolar lines determines the epipoles $\mathbf{e}$ and $\mathbf{e}'$

The epipoles contain information about the extrinsic camera parameters, namely the position of the camera centre $\mathbf{C}$ and the orientation of the optical axis. However, this information cannot be directly retrieved.

At this point, we define the projection matrices $\mathtt{P}$ and $\mathtt{P}'$ for the first and second view, respectively. In contrast to the previously defined metric projection matrices $\mathtt{P}_1$ and $\mathtt{P}_2$, they may still have a projective ambiguity. The epipolar line $\mathbf{l}_e$ in figure 6.3 is defined by the image point $\mathbf{x}$ and the epipole $\mathbf{e}$. Thus, we can say that $\mathbf{l}_e = [\mathbf{e}]_\times \mathbf{x}$, where

$$[\mathbf{e}]_\times = \begin{bmatrix} 0 & -w_e & y_e \\ w_e & 0 & -x_e \\ -y_e & x_e & 1 \end{bmatrix} \tag{6.6}$$

is the skew-symmetric matrix, representing the cross product with $\mathbf{e} = (x_e,\, y_e,\, w_e)^\top$. Moreover, $\Pi_e = \mathtt{P}^\top \mathbf{l}_e$ for the first and $\Pi_e = \mathtt{P}'^\top \mathbf{l}'_e$ for the second camera. It follows,

$$\begin{aligned} \mathtt{P}'^{\top}\mathbf{l}'_e &= \mathtt{P}^{\top}\mathbf{l}_e \\ \mathbf{l}'_e &= \left(\mathtt{P}'^{\top}\right)^{\dagger}\mathtt{P}^{\top}\mathbf{l}_e \\ \mathbf{l}'_e &= \left(\mathtt{P}'^{\top}\right)^{\dagger}\mathtt{P}^{\top}\left[\mathbf{e}\right]_{\times}\mathbf{x}' \\ \mathtt{F} &= \left(\mathtt{P}'^{\top}\right)^{\dagger}\mathtt{P}^{\top}\left[\mathbf{e}\right]_{\times} \end{aligned} \tag{6.7}$$

where $\left(\mathtt{P}'^{\top}\right)^{\dagger}$ indicates the pseudo-inverse of the transposed camera projection matrix for the second view $\mathtt{P}'$.

$\mathtt{F} = \left(\mathtt{P}'^{\top}\right)^{\dagger}\mathtt{P}^{\top}\left[\mathbf{e}\right]_{\times}$ is the *fundamental matrix* that determines for each point $\mathbf{x}$ in the first image its corresponding epipolar line $\mathbf{l}'_e$ in the second image and vice versa. The fundamental matrix was introduced by [Longuet-Higgins 1981] and has the following properties.

$$\boxed{\begin{aligned} \mathbf{l}'_e &= \mathtt{F}\mathbf{x} \\ \mathbf{x}'^{\top}\mathtt{F}\mathbf{x} &= 0 \end{aligned}} \tag{6.8}$$

These conditions are valid for any pair of corresponding points $\mathbf{x}$ and $\mathbf{x}'$. If the image point $\mathbf{x}$ is congruent with the epipole $\mathbf{e}$, the product $\mathtt{F}\mathbf{e}$ is equal to zero. For the epipolar line $\mathbf{l}'_e$ and the epipole $\mathbf{e}'$ in the second image, we have similar properties.

$$\boxed{\begin{aligned} \mathtt{F}^{\top}\mathbf{x}' &= \mathbf{l}_e \\ \mathtt{F}^{\top}\mathbf{e}' &= 0 \end{aligned}} \tag{6.9}$$

Notice that $\mathtt{F}$ is of rank 2. This is a useful constraint for the estimation of the fundamental matrix. Traditionally, the fundamental matrix is estimated using a classical RANSAC scheme that randomly selects 7 sample points equally distributed over the whole image space. The fundamental matrix is computed for this minimal number of points and the number of inliers is determined. The best solution is iteratively evaluated and refined using a minimisation algorithm on all inliers [Hartley and Zisserman 2004].

6.2.1 Retrieving the Projection Matrices

One of the most significant properties of the fundamental matrix is that it determines the projection matrices of the two views up to an unknown projective ambiguity. Let $\mathbf{k}'$ be a line in the second image not containing the epipole $\mathbf{e}'$, and Π a plane obtained by back-projecting the image line in space. Moreover, let $\mathbf{X}_\Pi$ be a point located on the plane Π and $\mathbf{x}_\Pi$ its projection in the first image. The epipolar line $\mathbf{l}'_e$, obtained with the relation $\mathbf{l}'_e = \mathtt{F}\mathbf{x}_\Pi$ intersects with $\mathbf{k}'$ in the second image. This intersection represents the projection of the point $\mathbf{X}_\Pi$ in the second image $\mathbf{x}'_\Pi$. Thus, $\mathbf{x}'_\Pi \sim [\mathbf{k}']_\times \mathtt{F}\mathbf{x}_\Pi$. The choice

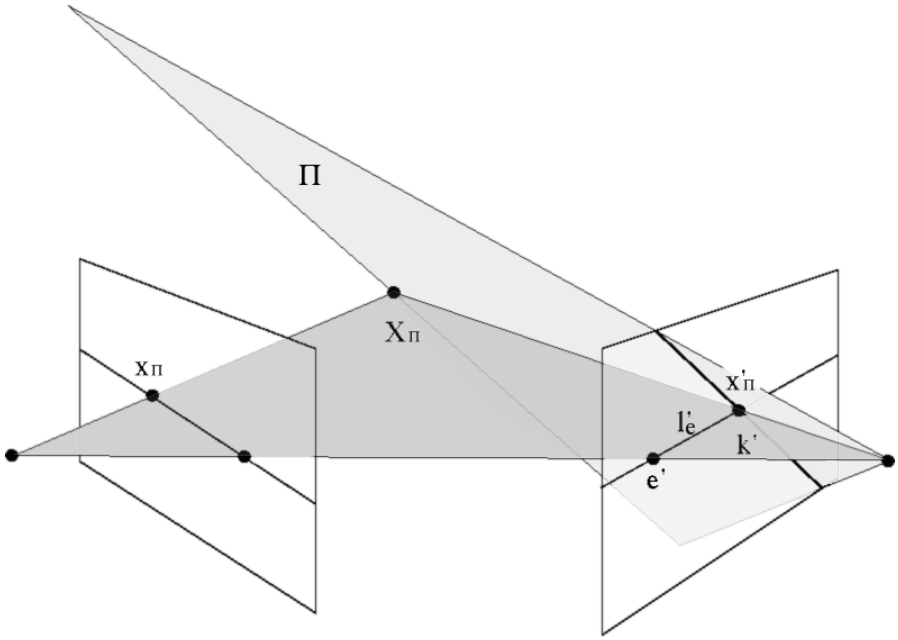

Figure 6.5: A point $\mathbf{x}_\Pi$ is transferred via the plane Π to a point $\mathbf{x}'_\Pi$ on the second image plane. It can also be seen as a Homography $\mathtt{H} = [\mathbf{k}']_\times \mathtt{F}$ of a point from one plane to another.

of the plane Π is projectively equivalent for any point $\mathbf{x}_\Pi$, except the epipole itself. The plane is simply used here as a means of defining a point map from one image to another. In order to avoid that the second epipole $\mathbf{e}'$ lies on the line $\mathbf{k}'$, the line with the parameters of $\mathbf{e}'$ itself is a good choice for $\mathbf{k}'$, since $\mathbf{k}'^\top \mathbf{e}' = \mathbf{e}'^\top \mathbf{e}' \neq 0$. Hence, the relation (homography) between the projections of the scene point $\mathbf{X}$ in the first and second image becomes:

$$\mathbf{x}' \sim [\mathbf{e}']_\times \mathtt{F}\mathbf{x} \tag{6.10}$$

Suppose that the first projection matrix is known as $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$, and the second projection matrix has the form $\mathtt{P}' = [\mathtt{A} \mid \mathbf{a}]$. Combining (6.10) with $\mathbf{x} \sim \mathtt{P}\mathbf{X}$, we are able to find $\mathtt{A}$.

$$\begin{aligned} \mathbf{x}' &\sim [\mathbf{e}']_\times \mathtt{F}\mathtt{P}\mathbf{X} \\ \mathbf{x}' &\sim [\mathbf{e}']_\times \mathtt{F}\,[\mathtt{I} \mid \mathbf{0}]\,\mathbf{X} \\ \mathbf{x}' &\sim \left[[\mathbf{e}']_\times \mathtt{F} \mid \mathbf{0}\right]\mathbf{X} \\ \mathbf{x}' &\sim \mathtt{P}'\mathbf{X} \\ \Rightarrow \mathtt{A} &= [\mathbf{e}']_\times \mathtt{F} \end{aligned} \tag{6.11}$$

Because of its multiplication with a zero vector, the fourth component of the vector $\mathbf{X}$ is not taken into consideration. Therefore, we can conclude that the result $\mathtt{A} = [\mathbf{e}']_\times \mathtt{F}$ is correct up to an ambiguity factor, which will be determined later in this section.

In order to estimate the translational part $\mathbf{a}$ of the second projection matrix $\mathtt{P}'$, we simply choose the projection of the first image centre $\mathbf{C} = (0,\, 0,\, 0,\, 1)^\top$, which is known to be the epipole $\mathbf{e}'$:

$$\mathtt{P} = \left[[\mathbf{e}']_\times \mathtt{F} \mid \mathbf{e}'\right] \tag{6.12}$$

There is another ambiguity λ deriving from the unknown distance between both camera centres. $\mathbf{C}$ and $\mathbf{C}'$. Suppose the two pairs of projection matrices $\tilde{\mathtt{P}}_1 = [\mathtt{I} \mid \mathbf{0}]$, $\tilde{\mathtt{P}}'_1 = [\mathtt{A}_1 \mid \mathbf{a}_1]$, and $\tilde{\mathtt{P}}_2 = [\mathtt{I} \mid \mathbf{0}]$, $\tilde{\mathtt{P}}'_2 = [\mathtt{A}_2 \mid \mathbf{a}_2]$. In order to find the ambiguity λ, we suppose the two pairs of projection matrices belong to the same fundamental matrix $\mathtt{F}$. Furthermore, we have

$$\begin{aligned} \mathbf{l}' = [\mathbf{e}']_\times \mathbf{x}' &= \mathtt{F}\mathbf{x} \\ [\mathbf{e}']_\times \mathtt{P}'\mathbf{X} &= \mathtt{F}\mathtt{P}\mathbf{X} \\ \mathtt{F} &= [\mathbf{e}']_\times \mathtt{P}'\mathtt{P}^\dagger \end{aligned} \quad , \tag{6.13}$$

and we know that $\mathbf{e}'$ is in our case the projection of the origin of the world coordinate system in the second image. As we saw before, this corresponds to the transitional part $\mathbf{a}_1$ and $\mathbf{a}_2$ in $\tilde{\mathtt{P}}'_1$ and $\tilde{\mathtt{P}}'_2$ respectively.

It results that $\mathtt{F} = [\mathbf{a}_1]_\times \mathtt{A}_1 = [\mathbf{a}_2]_\times \mathtt{A}_2$ and $\mathbf{a}_1^\top \mathtt{F} = \mathbf{a}_2^\top \mathtt{F} = \mathbf{0}$, since $\mathbf{a}_1^\top [\mathbf{a}_1]_\times \mathtt{A}_1 = \mathbf{0}$. Hence, $\mathbf{a}_2$ has to be $\lambda \mathbf{a}_1$ so that the cross product $\mathbf{a}_2^\top [\mathbf{a}_1]_\times$ is equal to zero. It follows

$$\begin{aligned} [\mathbf{a}_1]_\times \mathtt{A}_1 &= [\lambda \mathbf{a}_1]_\times \mathtt{A}_2 \\ [\mathbf{a}_1]_\times \mathtt{A}_1 &= [\mathbf{a}_1]_\times \lambda \mathtt{A}_2 \\ [\mathbf{a}_1]_\times (\mathtt{A}_1 - \lambda \mathtt{A}_2) &= \mathbf{0}. \end{aligned} \tag{6.14}$$

This means that each column of $(\mathtt{A}_1 - \lambda \mathtt{A}_2)$ has to be collinear to $\mathbf{a}_1$ in order to get a cross product equal to zero. Therefore, $(\mathtt{A}_1 - \lambda \mathtt{A}_2) = \mathbf{a}_1 \mathbf{v}^\top$ for a given vector $\mathbf{v} \in \mathbb{R}^3$. We deduce that for any fundamental matrix, the following two projection matrices exist.

$$\boxed{\begin{aligned} \mathtt{P} &= [\mathtt{I} \mid \mathbf{0}] \\ \mathtt{P}' &= \left[[\mathbf{e}']_\times \mathtt{F} - \mathbf{e}' \mathbf{v}^\top \mid \lambda \mathbf{e}' \right] \end{aligned}} \tag{6.15}$$

It is possible to compute $\mathtt{P}'$, up to the unknown parameters λ and $\mathbf{v}$, with two images of the same scene from two slightly different viewpoints (non-zero baseline), and to reconstruct the scene in 3D space. The unknown parameters have four degrees of freedom (λ and the coefficients of $\mathbf{v}$), hence the 3D reconstruction will be a projective representation of the scene. However, it is possible to upgrade the projective reconstruction to a metric one using the constraints on the camera matrices, introduced in section 6.1.

6.3 Essential Matrix

The *essential matrix* was introduced as a special form of the fundamental matrix by [Longuet-Higgins 1981][1]. The essential matrix is defined as follows.

$$\boxed{\mathtt{E} = \mathtt{K}'^\top \mathtt{F} \mathtt{K}} \tag{6.16}$$

As the fundamental matrix can also be written as $\mathtt{F} = \mathtt{K}'^{-\top} [\mathbf{t}]_\times \mathtt{R} \mathtt{K}^{-1}$ (see [Hartley and Zisserman 2004]), it is straightforward to see that $\mathtt{E} = [\mathbf{t}]_\times \mathtt{R}$.

[1] The authors presented both the fundamental matrix and the essential matrix in the same paper.

The rotation matrix $\mathtt{R}$ is of full rank and $[\mathbf{t}]_\times$ the skew symmetric matrix corresponding to the cross product with the vector $\mathbf{t}$ is of rank 2. Hence, the essential matrix $\mathtt{E}$ is of rank 2 as well. The singular values of the rotation matrix are all equal to one. Hence, the singular values of the essential matrix have to be equal to the singular values of the skew symmetric matrix. More precisely, the first two singular values s_1 and s_2 are both equal to the length of the translation vector $\mathbf{t}$, and the last one, s_3, is zero ($s_1 = s_2,\ s_3 = 0$).

Our constraints on the camera parameters and the above properties of the singular values allow for a direct search of the focal length, similar to [Dick *et al.* 2000]. According to our constraints on the camera matrix, the focal length is the only remaining unknown of our $\mathtt{K}$ and is supposed to be equal for both cameras. In the search for its true value, the focal length is varied in order to minimise the penalty function in equation (6.17). This is a simplified version of the one used in [Mendonca and Cipolla 1999].

$$\boxed{\mathcal{C} = \frac{s_1 - s_2}{s_1 + s_2}} \tag{6.17}$$

The minimisation is carried out as follows. First, we set the focal length to the value 1 and compute the camera matrix as specified in equation (6.17). The value 1 is chosen because the cost function converges faster to a minimum from a low initial value than for a big one (see figure 6.7). Secondly, the essential matrix is computed according to equation (6.16). Finally, the singular values are computes using singular value decomposition (SVD), and this yields the cost function $\mathcal{C}$. Levenberg-Marquardt [Hartley and Zisserman 2004] is used to find the best value of the focal length that minimises the non-linear cost function. After minimising the above cost function, we end up with the focal length of the cameras. Hence, the camera matrix $\mathtt{K}$ is known. Moreover, the essential matrix defines the extrinsic camera parameters up to a four-fold ambiguity. The SVD yields the the following representation of the essential matrix

$$\mathtt{E} = \mathtt{U} \cdot \mathrm{diag}(1,\ 1,\ 0)\mathtt{V}^\top, \tag{6.18}$$

where $\det(\mathtt{U}) > 0$ and $\det(\mathtt{V}) > 0$. If the first camera matrix has the canonical form $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$, there are four possible choices for the second projection matrix $\mathtt{P}'$ (compare figure 6.6),

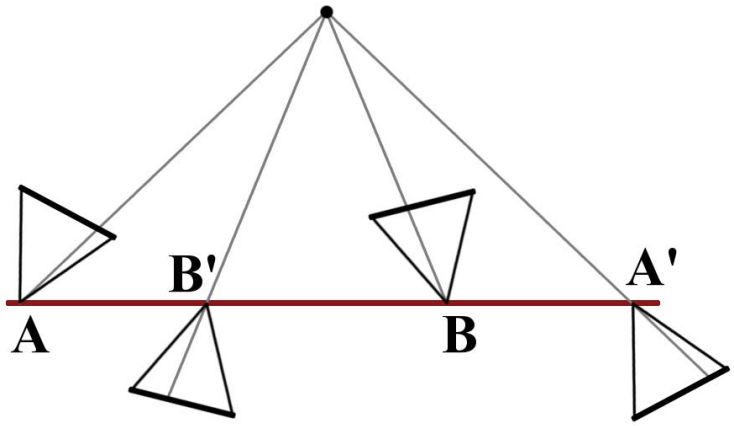

Figure 6.6: The four possible solutions for the second projection matrix $\mathtt{P}'$ are $(A,\, B)$, $(A,\, B')$, $(A',\, B)$, and $(A',\, B')$. Note that for only one configuration $(A,\, B)$, the space point is in front of both cameras (cheirality constraint). This constraint is easily verified.

$$\begin{aligned}\mathtt{P}'_1 &= [\mathtt{R}_a \mid \mathbf{u}] \\ \mathtt{P}'_2 &= [\mathtt{R}_a \mid -\mathbf{u}] \\ \mathtt{P}'_3 &= [\mathtt{R}_b \mid \mathbf{u}] \\ \mathtt{P}'_4 &= [\mathtt{R}_b \mid -\mathbf{u}],\end{aligned} \tag{6.19}$$

with $\mathtt{R}_a = \mathtt{U}\mathtt{W}\mathtt{V}^\top$, $\mathtt{R}_b = \mathtt{U}\mathtt{W}^\top\mathtt{V}^\top$, $\mathbf{u}$ the last column vector of the matrix $\mathtt{U}$, and

$$\mathtt{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.20}$$

One solution corresponds to the correct solution $(A,\, B)$, as shown in figure 6.6. The other three solutions are obtained by reflecting the individual cameras about the baseline, supposing the scene point fixed. For the correct solution, all points should lie in front of both cameras simultaneously except in the case of a mismatch. This solution can be found using the cheirality constraint that verifies if a given scene point lies in front or behind a given camera [Hartley 1993].

According to [Nistér 2004], this is implemented as follows. A given point match pair $\mathbf{m}$ and $\mathbf{m}'$ is triangulated using the projection matrices $\mathtt{P}$ and $\mathtt{P}'_1$ as described above. This yields the space point $\mathbf{M} = (M_1,\, M_2,\, M_3,\, M_4)$, and its

back-projections $\hat{\mathbf{m}} = (\hat{m}_1, \hat{m}_2, \hat{m}_3)$ and $\hat{\mathbf{m}}' = (\hat{m}'_1, \hat{m}'_2, \hat{m}'_3)$, respectively. If $c_1 = M_3 M_4 < 0$, the point is behind the first camera. If $c_2 = \hat{m}'_3 M_4 < 0$, the point is behind the second camera. If $c_1 > 0$ and $c_2 > 0$, $\mathtt{P}'_1$ and $\mathbf{M}$ correspond to the true configuration. If $c_1 < 0$ and $c_2 < 0$, $\mathtt{P}'_2$ is the correct solution. However, if $\mathbf{M}$ is triangulated and back-projected using the projection matrices $\mathtt{P}$ and $\mathtt{P}'_3$. $c_1 c_2 < 0$. If $M_3 M_4 > 0$, $\mathtt{P}'_3$ is correct, otherwise $\mathtt{P}'_4$ is the right choice. For stability reasons (mismatches), all points that are used for the estimation of $\mathtt{E}$ are also used to select the correct configuration with the cheirality constraint. As we suppose the presence of a higher number of correct matches than mismatches, we vote for the solution with the highest number of interest points in front of both cameras.

Once the correct projection matrix $\mathtt{P}'$ has been determined, the rotation matrix $\mathtt{R}$ and the translation vector $\mathbf{t}$ of the second camera can be found. Therefore, the metric projection matrices $\mathtt{P}_1$ and $\mathtt{P}_2$ can easily be deduced.

$$\begin{aligned} \mathtt{P}_1 &= \mathtt{K}[\mathtt{I} \mid \mathbf{0}] \\ \mathtt{P}_2 &= \mathtt{K}[\mathtt{R} \mid -\mathtt{R}\mathbf{t}] \end{aligned} \tag{6.21}$$

Note that the focal length has not to be initialised beforehand, since there are no local minima of the cost function in equation (6.17) as shown in figure 6.7.

The 3D reconstruction of the individual point matches is obtained by triangulation using the projection matrices. For more information, see [Hartley and Sturm 1995, Hartley and Zisserman 2004].

6.4 Algorithm

The algorithm for the camera self-calibration and 3D reconstruction is divided in the following steps.

1. Interest points are identified and matched between two input images using a given interest point detector and descriptor. Traditionally, the Harris corner detector [Harris and Stephens 1988] is applied and the matches are found using normalised cross-correlation. Moreover, the location of the interest point is adapted in order to get a more accurate interest point location and therefore a more accurate reconstruction of the point match.

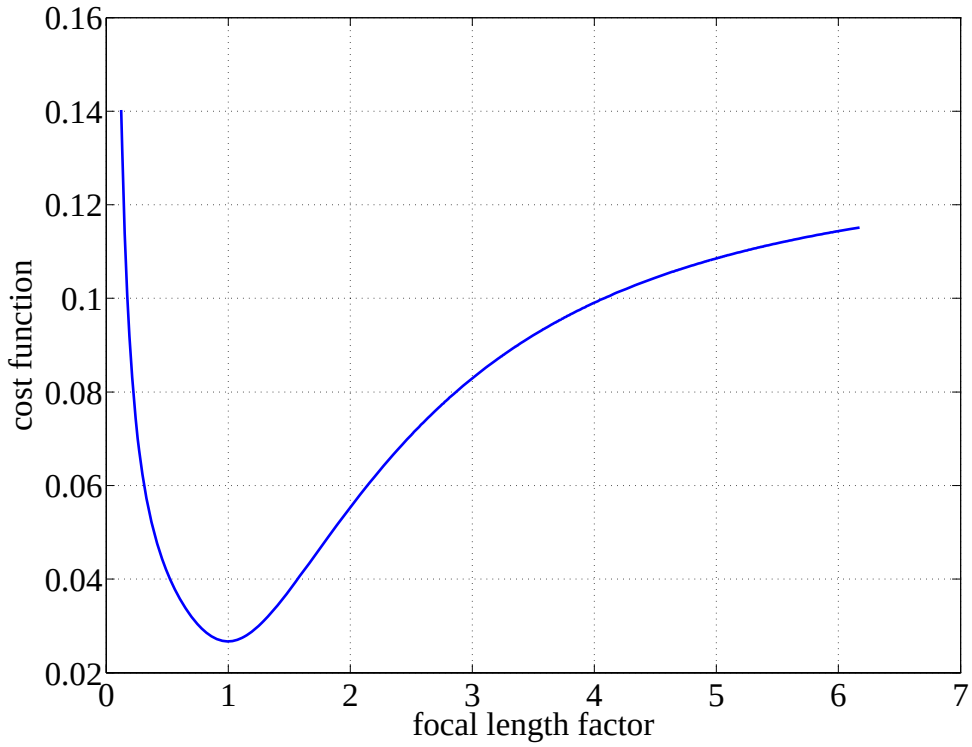

Figure 6.7: Minimisation of the cost function in equation (6.17). The x-axis represents the focal length factor, this means 1 is the *real* focal length and 2 is two times this value. It clearly shows that there is only one global minimum and that an initialisation with a low value is favourable.

However, this is not possible for wide-baseline conditions, as the robustness of the normalised cross-correlation is very sensitive to view changes [Mikolajczyk and Schmid 2005]. Therefore, we use the SURF detector and descriptor for obtaining the image correspondences. We suppose that the location of the SURF interest points is less accurate than the adapted Harris locations. Therefore, we expect the results to be less accurate in this regard.

2. The fundamental matrix is estimated using a classical RANSAC scheme [Fischler and Bolles 1981]. Mismatches are filtered out with the epipolar constraint. This means that a matching interest point has to lie within a certain band around the corresponding epipolar line in the other image. Furthermore, the cheirality constraint is used to filter out mismatches that are accidentally not violating the epipolar constraint. The fundamental

matrix is reestimated with all inliers in order to get a more accurate solution.

3. Based on the additional assumption made on the camera intrinsics, the Essential matrix is determined by minimising the cost function in equation (6.17). From the essential matrix, the metric projection matrices are derived.

4. All interest point correspondences are projected in 3D space using the technique of triangulation.

In general, the reprojected 3D points do not lie exactly on the measured image points. This is due to the fact that the projected optical rays do not intersect in a uniquely-defined 3D point. Moreover, the camera parameters are not always perfectly retrieved. Therefore, an additional bundle adjustment step is performed in order to get a better estimate of the 3D points and the camera parameters simultaneously, see [Laveau 1996, Triggs *et al.* 2000, Hartley and Zisserman 2004, Pollefeys *et al.* 2004] for more details. The basic idea of the bundle adjustment is to minimise the reprojection error (distance between detected interest point and reprojected 3D point) in the individual images by adjusting the location of the 3D points and/or the camera parameters. This minimisation is usually carried out using the Levenberg-Marquardt method [Hartley and Zisserman 2004]. The robustness of this step depends on the number of residuals (reprojection error) and the degrees of freedom (coordinated of the 3D points and varying camera parameters). If the number of residuals is bigger than the degrees of freedom, the minimisation algorithm can converge to a meaningful solution. For two views with unknown location of the 3D points and unknown camera parameters, the degrees of freedom exceed the number of residuals. Therefore, bundle adjustment makes no sense because there are too many degrees of freedom to yield a stable solution. We will see in the next chapter, that bundle adjustment is possible for the special case of line matches between two views.

6.5 Results

We evaluate the accuracy of our Fast-Hessian detector, presented in chapter 2, for the application of camera self-calibration and 3D reconstruction. The first

evaluation compares different state-of-the-art interest point detectors for the two-view case. A known scene is used to provide some quantitative results. The second evaluation considers the N-view case for camera self-calibration and dense 3D reconstruction from multiple images, some taken under wide-baseline conditions.

2-view Case In order to evaluate the performance of different interest point detection schemes for camera calibration and 3D reconstruction, we created a controlled environment. An good scene for such an evaluation are two highly textured planes forming together a right angle (measured 88.6° in our case), see figure 6.8. The images are of size 800×600 and the number of matches are for all interest point detectors empirically set to 800 by adjusting the threshold for the detection step (corner or blob response). The matching was carried out using the SURF-128 descriptor for all interest points. All matches were correct. The location of the two planes was evaluated using RANSAC. In order to find a plane in the reconstructed 3D point cloud, we randomly select 3 points which determine a 3D plane. The solution with the highest number of inliers within a certain neighbourhood is selected and refined using a least-square fit. This steps are repeated in order to find the second plane among the remaining 3D points (outliers of the first estimation). The evaluation criteria are the angle between the two planes, the mean distance and the variance of the reconstructed 3D points to their respective planes for different interest point detectors.

Table 6.1 shows these quantitative results for our two versions of the Fast-Hessian detector (FH-9 and FH-15), the DoG features of SIFT [Lowe 2004], and the Hessian- and Harris-Laplace detectors proposed by Krystian Mikolajczyk and Cordelia Schmid [Mikolajczyk and Schmid 2004]. The FH-15 detector clearly outperforms its competitors.

Figure 6.9 shows the orthogonal projection of the Fast-Hessian (FH-15) features for the reconstructed angle. Curiously, the theoretically more correct approaches like the Harris- and Hessian-Laplace detectors perform worse than the approximations (DoG and the SURF features).

Another example is shown in figure 6.10 where two images of a temple and the respective 3D position for the point matches between two views are shown. Unfortunately, sparse 3D reconstructions are visually not as appealing as it is the case for dense 3D models. Therefore, we limit the results for the two-view case to these two examples.

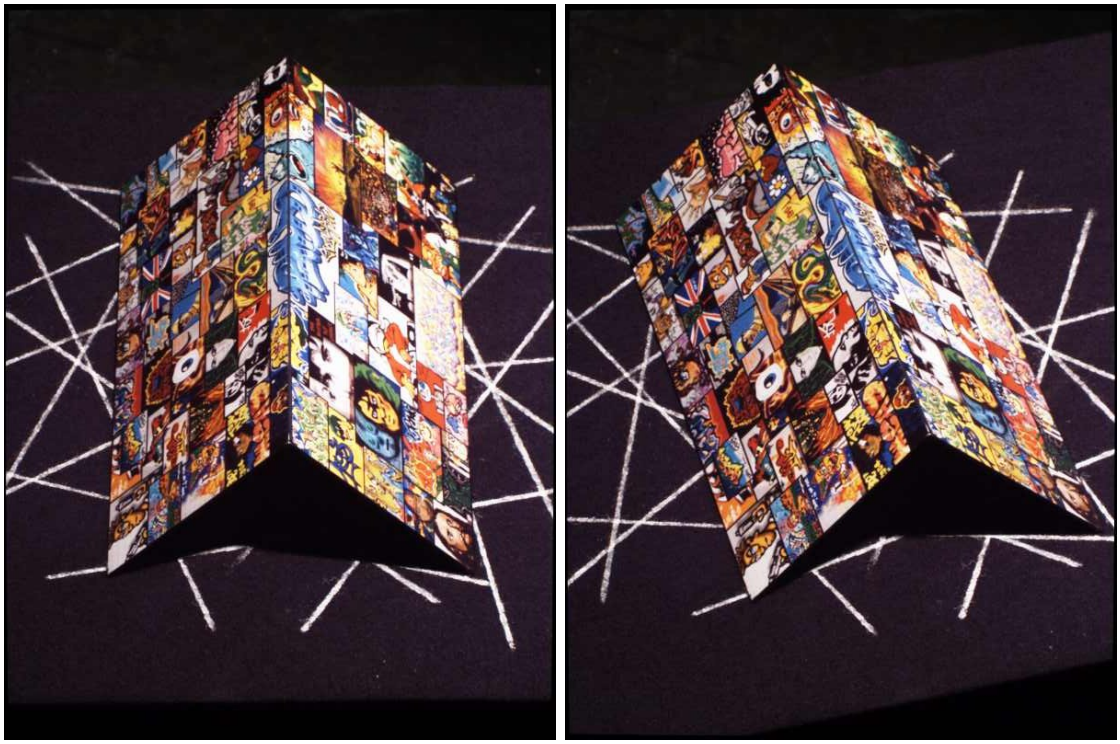

Figure 6.8: Input images for the quantitative detector evaluation. This represents a good scene choice for the comparison of different types of interest point detectors, as its components are simple geometric elements.

N-view Case The SURF detection and description algorithm has been integrated with the Epoch 3D Webservice of the VISICS research group at the K.U. Leuven[2]. This webservice allows users to upload sequences of still images to a server. There, the calibration of the cameras and dense depth maps

[2] `http://homes.esat.kuleuven.be/~visit3d/webservice/html`

detector	angle	mean dist	variance
FH-15:	88.5°	1.14 px	1.23 px
FH-9:	88.4°	1.64 px	1.78 px
DoG:	88.9°	1.95 px	2.14 px
Harris-Laplace:	88.3°	2.13 px	2.33 px
Hessian-Laplace:	91.1°	2.85 px	3.13 px

Table 6.1: Comparison of different interest point detectors for the application of camera calibration and 3D reconstruction. The true angle is $88.6°$

Figure 6.9: Orthogonal projection of the reconstructed angle shown in figure 6.8.

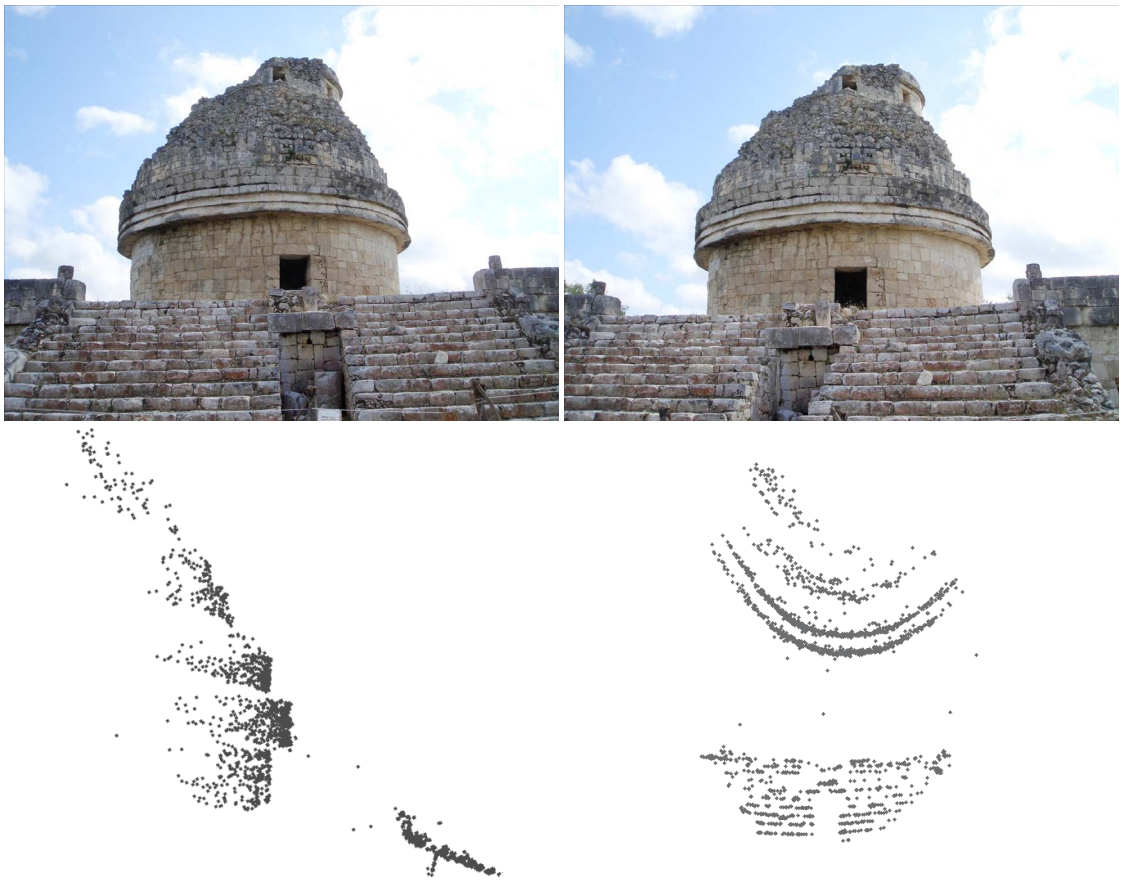

Figure 6.10: Top: Input images for camera calibration and sparse 3D model. Bottom: Sparse 3D model viewed from the side (left) and from top (right). The images were taken by Maurizio Forte, CNR-ITABC, Rome).

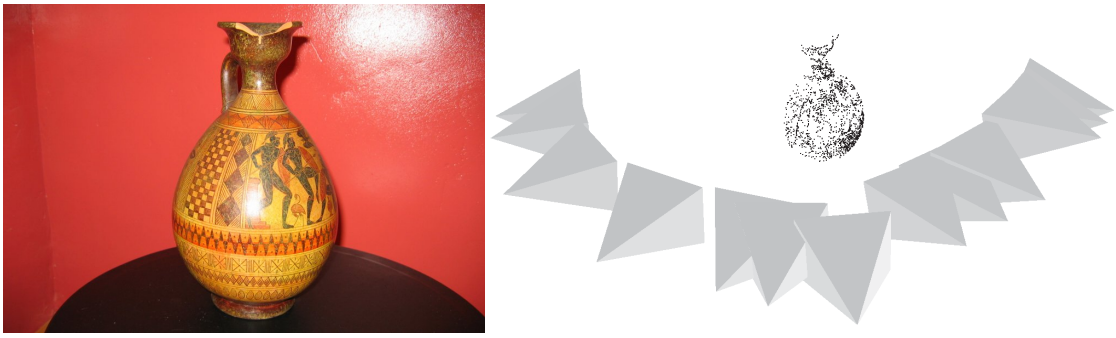

Figure 6.11: 3D reconstruction with KU-Leuven's 3D webservice. Left: One of the 13 input images for the camera calibration. Right: Position of the reconstructed cameras ant sparse 3D model of the vase.

are computed automatically using these images only [Vergauwen and Van Gool to appear]. During the camera calibration stage, features need to be extracted and matched between the images. It was found that the use of SURF features improved the results of this step for many uploaded image sets, especially when the images were taken quite far apart. The normal procedure using Harris corners and normalised cross correlation of image windows has problems matching such wide-baseline images. Furthermore, the SIFT detection and description scheme failed on some image sequences, where SURF succeeded to calibrate all the cameras accurately.

For the example in figure 6.11, the traditional approach managed to calibrate only 6 from a total of 13 cameras. Using SURF however, all 13 cameras could be calibrated. The vase is easily recognisable even with a sparse 3D model.

Figure 6.12 shows a typical wide-baseline problem: three images, taken from different, widely separated view points. It is a challenging example, as three images represent the absolute minimum number of images needed for an accurate dense 3D reconstruction. The obtained 3D model can be seen in figure 6.12 (bottom). In general, the quality of the camera calibration can best be appreciated on the basis of the quality of such resulting dense models. This is a typical example where the traditional Harris corner detector and the Normalised-Cross-Correlation matcher fail. Moreover, the computation time is drastically reduced using SURF instead.

6.6 Conclusion and Outlook

This chapter presented the theoretical basis for camera self-calibration and 3D reconstruction for interest point correspondences between only two views of a static scene. In order to succeed, some specific constraints have to be imposed on the cameras. In order to loosen these constraints, more images must be considered. An subject that has not been touched is the presence of radial distortion that may result in erroneous camera parameters and therefore in an inaccurate 3D locations. Current self-calibration methods are very sensitive to such artefacts, inaccurate interest point locations, and mismatches. However, for the N-view case, the radial distortion has been considered.

We evaluated SURF on a simple scene for which the desirable position of the reconstructed interest points could be evaluated. The Fast-Hessian detector turned out to be the most accurate for this single example. However, SURF turned out to be a competitive alternative for the N-view case of widely separated views. SURF manages to provide accurate results where other fail.

Nevertheless, more complete tests should be performed in order to completely evaluate the performance of SURF for structure from motion.

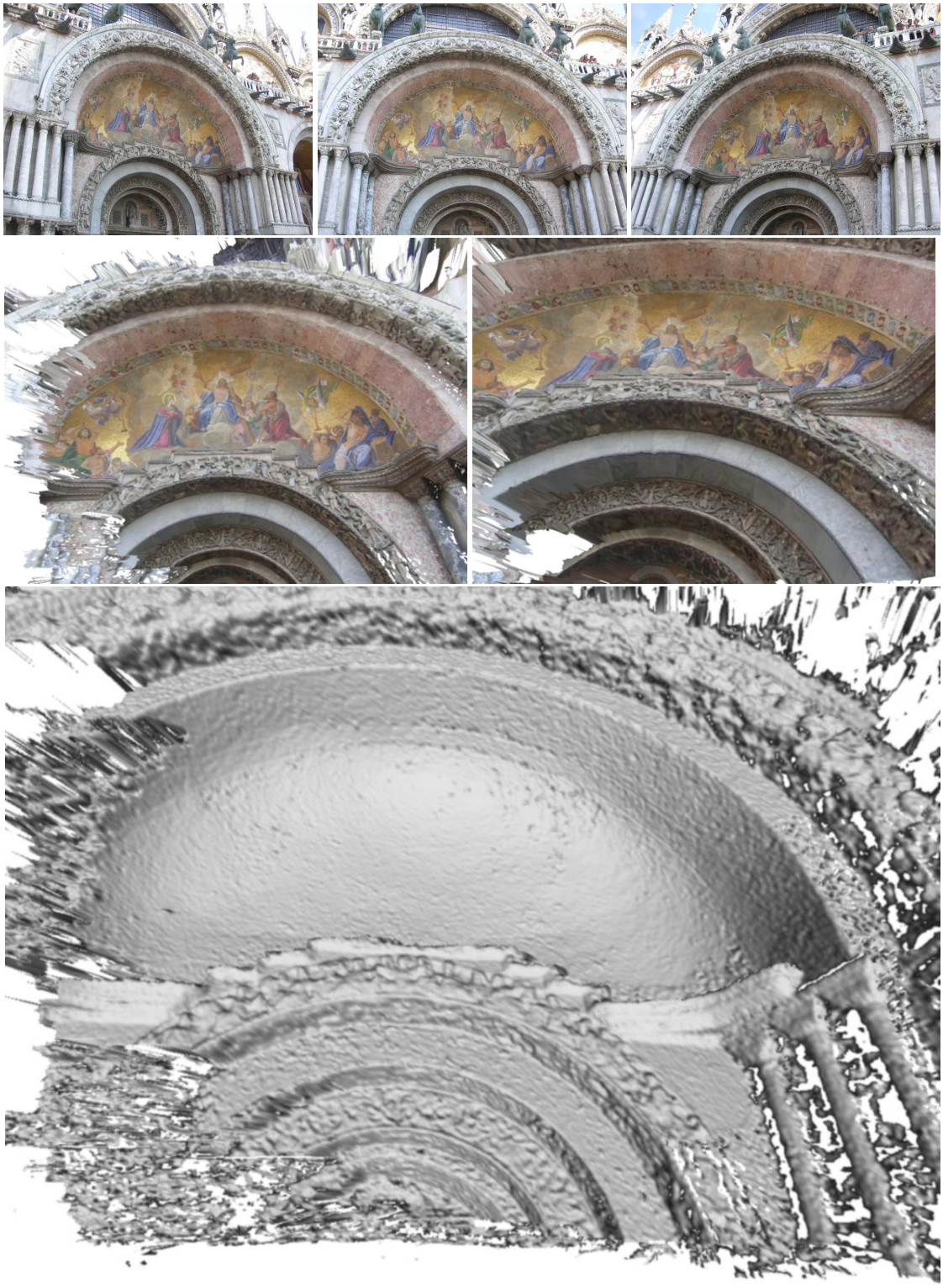

Figure 6.12: 3D reconstruction with KU-Leuven's 3D webservice. Top row: The 3 input images of a detail of the San Marco Cathedral in Venice. Middle row: Samples of the textured dense reconstruction. Bottom row: un-textured dense reconstruction. The quality of the dense 3D model is directly reflecting the quality of the camera calibration. The images were taken by Maurizio Forte, CNR-ITABC, Rome).

7

From Line Segments to 3D

The automatic 3D modelling of indoor environments is useful for applications like robot navigation, real estate web presentation, and surveillance. This chapter addresses the problem of automatically reconstructing indoor scenes that are supposed to be piecewise planar. In particular, we focus on stereo views of extremely poorly-textured scenes, imaged under wide-baseline conditions, but with some constraints on the cameras (known principal point, aspect ratio and skew), as discussed in the previous chapter. This case is important from a practical point of view, and it largely defies existing methods.

Traditional approaches fail on poorly-textured scenes due to an insufficient number of point matches. However, even in such cases, line segments may still abound for indoor scenes. The problem is that there is no geometrical constraint for line segments in two views, if their end points are not detected with good consistency, which they typically are not. Moreover, even with a calibrated stereo system, the reconstruction of line segments is still ill-conditioned, especially when they lie in or close to epipolar planes.

We present a fully automated pipeline for the 3D reconstruction of such weakly textured indoor scenes, from camera calibration and bundle adjustment, up to the recovery of a complete 3D model.

7.1 Related Work

Most approaches for the automatic reconstruction of piecewise planar scenes use homographies with plane sweeping [Baillard and Zisserman 2000, Werner

and Zisserman 2002]. In poorly textured scenes, however, the image-based comparisons of the different planes are unreliable. Also, for a robust pixel-wise mapping through homographies, plane sweeping depends on well reconstructed features and accurate camera calibration, for which more than two views are recommended.
[Dick *et al.* 2000] suggest a method based on Bayesian inference. Point matches are used for the generation of a few rough plane hypotheses. Under the assumption of orthogonality, these are then subject to a model-based selection. The obtained model is semi-automatically refined with a set of predefined insets like windows, using plane parallax. The assumption of orthogonality is in practice often not fulfilled.
[Bartoli *et al.* 2001, Bartoli and Sturm 2003] studied the problem of structure-from-motion for piece-wise planar scenes, using points on planes. They focus on a minimal parametrisation using 2D entities. The final reconstruction step still requires some input from the user. Yet another strand are intentionally semi-automatic approaches [Debevec 1996, Cipolla *et al.* 1999], which provide very convincing results. The drawback, however, is the substantial amount of required user interaction.

Stereo reconstruction with line segments is a more challenging task than with interest points. At least three views are needed to arrive at a geometrical constraint, via the trifocal tensor [Hartley 1994]. Fairly reliable reconstructions can be obtained with lines in multiple views [Taylor and Kriegman 1995]. [Zhang 1995] was the first to tackle the two-view case. He assumes that two matched line segments contain the projection of a common part of the corresponding line segment in space (overlap). Besides the fact that the focal length has to be known (which we will derive from the images), this method is not stable enough for our purposes. This is due to the weak gradients in images of homogeneous scenes, which result in inaccurate and incomplete line segment detection. Therefore, the reprojected overlap is often too small. This artefact is even more pronounced in wide-baseline conditions. A similar approach to the one of Zhang was followed by [Montiel *et al.* 2000]. In contrast to the two previous methods, our approach tends to be based on more stable data. It is based on points (derived from the line segments), which are constrained in two images by the epipolar geometry.

The work of [McLauchlan *et al.* 2000] uses, similar to our method, intersections of lines – we will refer to them as *junctions* – to reconstruct indoor environments. However, they use fully calibrated image sequences. We deal

with only two partially calibrated views. This is useful for practical cases like robot kidnapping.

7.2 Overview of our Approach

Given a set of line segment correspondences between two views of a piecewise planar scene, we present a novel approach, based on a Binary Space Partitioning (BSP) tree that identifies candidate junctions as intersections of the infinite support lines of matched line segments. At the same time, the scene is segmented into potentially planar polygons. Junctions, derived from intersections of matched line segments, provide point correspondences, which allow for the estimation of the epipolar geometry.

Rerunning the algorithm of the BSP tree with the epipolar constraint yields a better segmentation and a more reliable set of junctions. The latter are matched end points of the extended line segments. As mentioned in section 6.4, bundle adjustment for two views on the camera and the 3D points makes no sense because of the high number of unknowns. The property of collinear junctions is used to reduce this number and therefore turns the bundle adjustment considerably more stable. In this manner it becomes possible to indirectly perform a bundle adjustment on line segments from only two views.

As the lines bordering a potentially planar polygon may be involved in the construction of multiple different planes, a set of these planes is constructed. The ideal planes are chosen such that the overall scene contains as few discontinuities as possible. The plane delineations are inferred directly from the polygons found by the BSP tree, resulting in a cleaned-up 3D model of the scene.

7.3 Junction Detection

It is difficult to estimate the epipolar geometry for images of poorly textured scenes, due to the low number of detected image features (interest points and line segments). For the same reason, the estimation of the fundamental matrix, using homography-based feature clustering [Sinclair *et al.* 1993], is not stable enough. The number of detected features hardly suffices to estimate a homography. Therefore, we propose a novel method, using polyhedral junctions, which is better suited and more robust for our type of scenes.

By 'junctions' we mean the projections of real scene corners like the ones arising from the intersection of three scene planes, or corners of a door or window frame, etc. Let $\mathbf{c_V}$ denote a V type junction, $\mathbf{c_T}$ a T junction, and $\mathbf{c_Y}$ a Y junction (see figure 7.4 for some examples). Junctions involving more than three lines will be treated the same way as Y junctions.

To identify these junctions, we consider the intersection of two or more pairs of line segment correspondences and the Harris corner response map R_H for both images. If the intersection of two lines coincides with a Harris corner, then it is most probable that the intersection is a junction.

Taking all possible line intersections into account would lead to far too many false junctions because R_H is not distinctive enough. Furthermore, as junctions arise from intersections of coplanar lines, it would be advantageous to perform a planar segmentation beforehand or at the same time. Finally, for a proper delineation of the planar regions, complete line segments are preferable. However, even after the merging step, explained in section 5.3, most line segments are still too short. We therefore suggest the use of a Binary Space Partitioning tree (BSP tree for short, cf. e.g. [Cormen *et al.* 1990]) to solve these problems simultaneously. First, it casts a hypothesis of the real line segment by elongation of the detected one. This segments the images into potentially planar polygons and thereby detects junctions. The resulting polygons provide a good starting point for the 3D reconstruction, as they already cast a hypothesis about the scene planes. Moreover, the search space of all intersections is drastically reduced. Once a line has partitioned the image, it can't be crossed anymore.

7.3.1 BSP Tree

A BSP tree is a recursive subdivision of the image into multiple regions. The root of the tree represents a region corresponding to the entire image. At each node, a region is split along a matched line segment into two sub-regions, which in turn represent the basis for the next splitting (branches of the tree), see figure 7.1 and 7.3.

This is done for all matched line segments in a reference view and transferred into the other image via the line segment correspondences. Line segments crossed by a partition (producing a T junction) are split into two separate parts (one for each sub-region), see figure 7.2. This results in a segmentation of the images into a set of polygons $\mathcal{P}$, as well as a set of potential junctions, $\mathcal{C}$. The

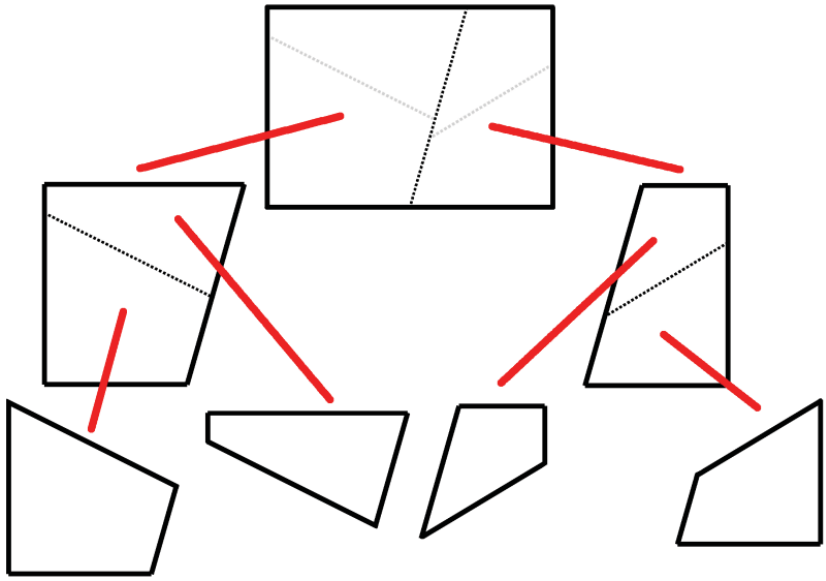

Figure 7.1: Schematic representation of a BSP tree. At each node, a region is split along a matched line segment into two sub-regions. The root of the tree is the initial image. The arborescent structure is clearly visible.

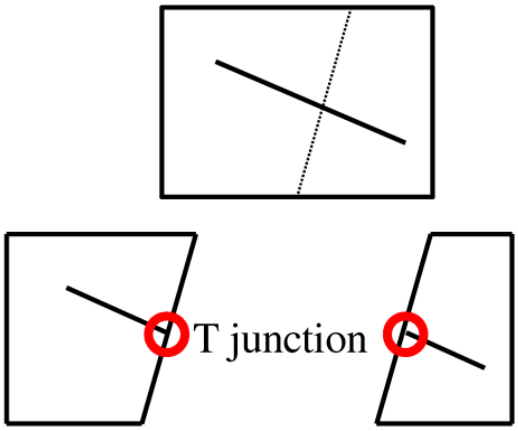

Figure 7.2: The generation of a T junction takes place when a detected line segment (top) is split by a partition in two parts.

outcome (see figure 7.4) strongly depends on the insertion order of the line segments into the tree, see subsection 7.3.2.

The drawback of this method is that the partitioning lines create more polygons than necessary and sometimes falsely split planar regions. This is taken care of as follows.

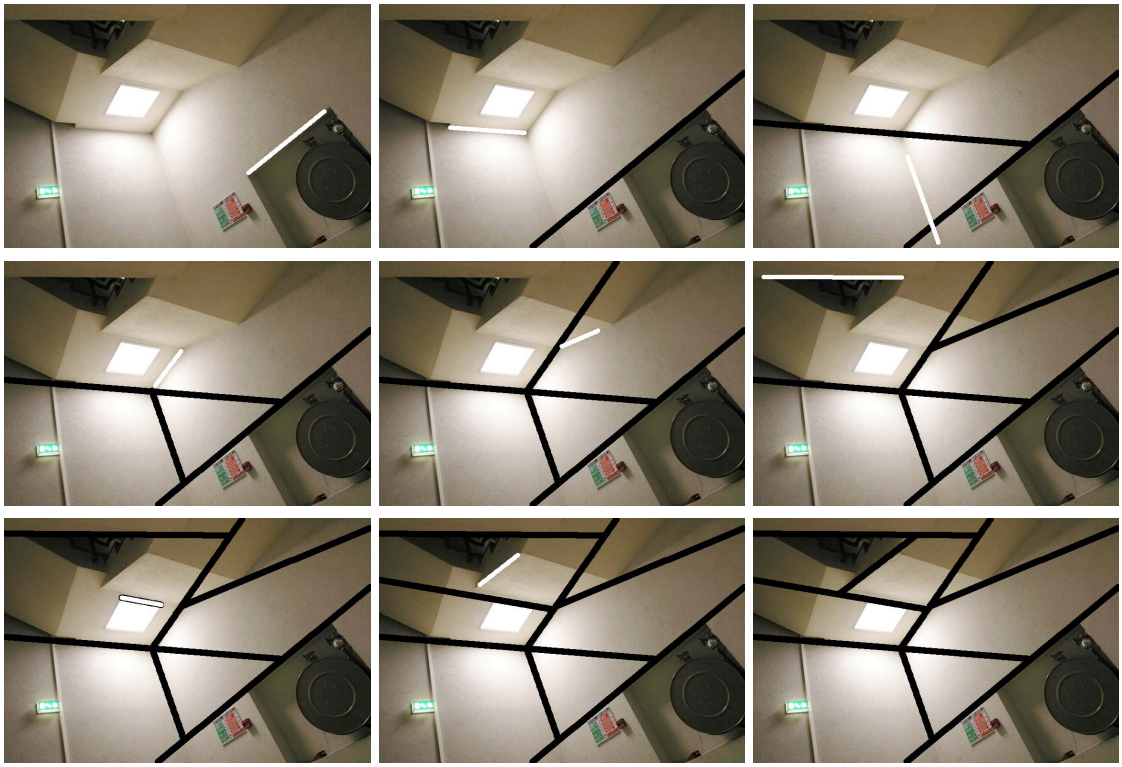

Figure 7.3: The generation of a BSP tree. The lines (black) are partitioning the image according a detected line segment (white), visible in each previous image. It represents a possible order in which lines could be inserted in a BSP tree.

Before evaluating the quality of the obtained space partitioning, the potential junctions (so far, intersections of two lines) are subject to an identification process that detects V, T and Y junctions. A Y junction is found when three lines intersect within a reasonably small neighbourhood in both images simultaneously. To distinguish between regular line intersections, and V and T junctions, the binarised Harris-corner-response maps $R_{H_{1,2}}$ for both images, I_1 and I_2, are considered. The strength $s(\mathbf{x})$ of a corner response is measured with

$$s(\mathbf{x}) = \det(\mathtt{A}(\mathbf{x})) - \alpha \mathrm{trace}^2(\mathtt{A}(\mathbf{x})), \tag{7.1}$$

where $\mathtt{A}(\mathbf{x})$ is the local second moment matrix of the Gaussian weighted image gradients ($\sigma = 2$) in a 9×9 window around the point $\mathbf{x}$. The parameter α is set to 0.06. These settings provided stable results in a previous interest point evaluation [Schmid *et al.* 2000]. Thus, the binarised Harris-corner-response map is written as

$$R_{H_{1,2}}(\mathbf{x}) = \begin{cases} 1 & \text{if } s(\mathbf{x}) > t \\ 0 & \text{otherwise} \end{cases} \tag{7.2}$$

The threshold t for $R_{H_{1,2}}$ is kept rather low (lack of texture), in our case $t = 80$. The Harris detector was selected because it detects corners and not blob-like structures as does SURF's Fast-Hessian detector. Furthermore, as the focal length is fixed for both views, a scale invariant approach is not required.

A V or T junction is detected if it lies near a Harris response $R_H = 1$ in at least one image. As it happens rarely that three or more lines intersect in one point in both images, Y junctions do not need a confirmation of the Harris response map. A T junction is a special case of a V junction. These are confirmed if the potential junction lies on the detected part of a line segment.

As the Harris response map is a rather weak constraint, some false junctions may be detected. In order to avoid such cases, we make two practical assumptions. First, we assume that a Y junctions result from the intersection of three

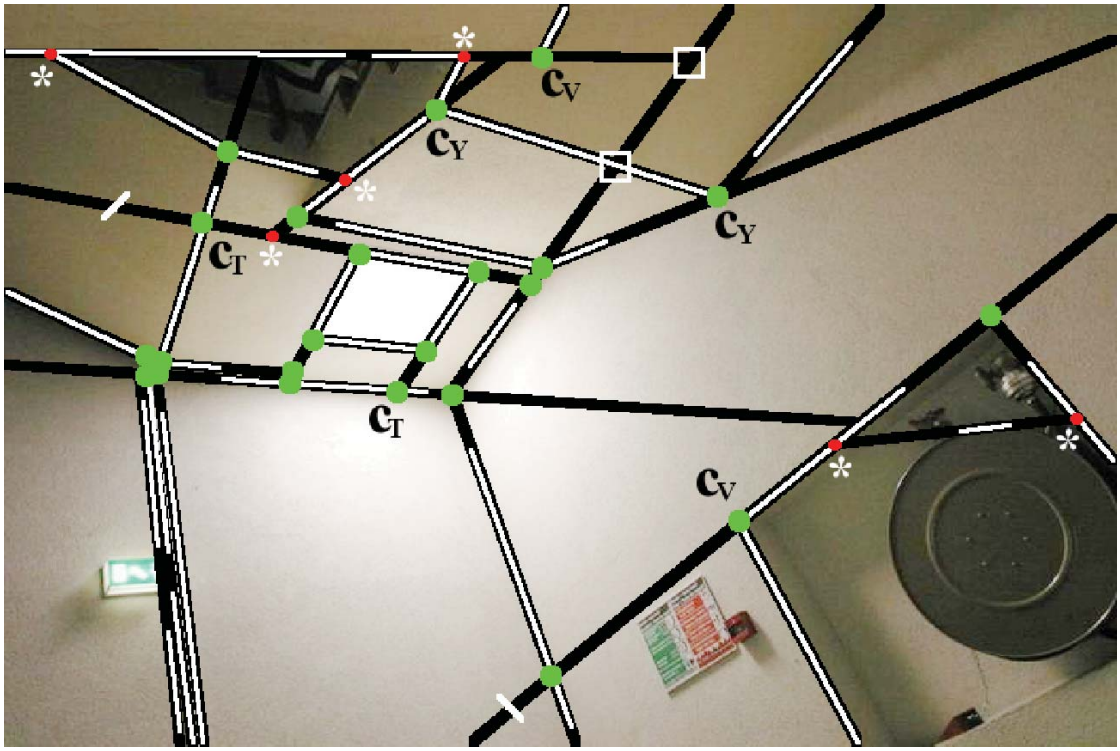

Figure 7.4: The concept of a BSP tree. The partitioning lines (black) are the elongations of the detected line segments (white). The BSP tree generates a segmentation into polygons and provides junctions (points). The junctions (asterisks) have no supporting Harris response and were therefore rejected. Later, this rejection will be based on the epipolar geometry. Intersections beyond the last Y junction are discarded (box), likewise for the partitioning lines after T junctions (cancelled lines).

scene planes. However, by the nature of the BSP tree, Y junctions lie on a partition line, which actually represents on one side of the Y junction (where the line segment was detected) the intersection of two planes, but may split a valid plane on the other side. This is not the case when it is involved in another Y junction, see figure 7.5. Hence, intersections with this line beyond the last Y junction are discarded, see figure 7.4. Second, we define T junctions as intersections of a partitioning line with the detected part of a line segment. We assume that in this case, the intersected line is the boundary of two different planes. Thus, all intersections involving the partitioning line beyond a T junction are discarded.

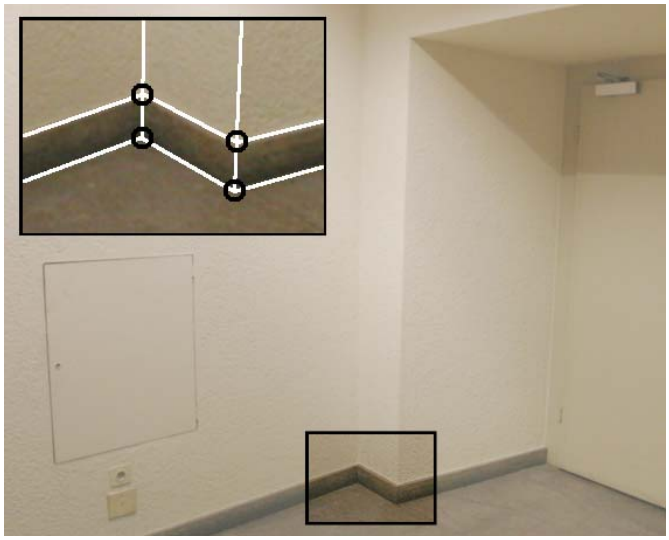

Figure 7.5: Multiple Y junctions may appear successively for some scenes. Therefore, the junctions created with the same partitioning line are not considered anymore beyond the last Y junction (seen from the detected line segment belonging to the partitioning line).

7.3.2 Segmentation Selection

As mentioned above, the obtained polygons $\mathcal{P}$ and junctions $\mathcal{C}$ depend on the insertion order of the lines into the tree. In order to choose the best segmentation out of a set of BSP trees, a quality measure of the latter has to be introduced.

The goal is to select the BSP tree with the highest number of detected junctions while keeping the remaining intersections (not upgraded to junctions)

low. Therefore, the set of intersections and detected junctions are subject of an evaluation step. For that purpose, we assign a weight to each type of junction.

V and T junctions are both detected the same way, therefore they are weighted equally. As Y junctions give important topological information about the scene, they are rewarded substantially better. These considerations yield equation 7.3 as measure for the quality of an intersection or junction. As we will formulate a minimisation problem, lower values are better.

$$f(\mathbf{c}) = \begin{cases} -3 & \text{if } \mathbf{c_Y} \text{ or } R_H(\mathbf{c_{V,T}}) = 1 \text{ in both images} \\ -1 & \text{if } R_H(\mathbf{c_{V,T}}) = 1 \text{ in one image} \\ 1 & \text{otherwise} \end{cases} \tag{7.3}$$

The weights -3, -1, and 1 were chosen experimentally. These values do not have a high impact on the results as long as the weight for Y junctions is smaller than the one for V and T junctions. A global cost function G_f is inferred by summing up all local measurements and adding an empirical penalty term for the number of invalid splittings S_{inv},

$$G_f(\mathcal{C}) = 10^9 S_{\text{inv}} + \sum_{\mathbf{c}\in\mathcal{C}} f(\mathbf{c}). \tag{7.4}$$

Invalid splittings are polygons which are built by line segments voting for non-collinear vanishing points. Such polygons are unlikely to be planar and therefore penalised with a term that then dominates the entire cost function. The second term is the sum of all local costs from equation (7.3) (and equation (7.5) explained later).

In order to get the best segmentation, we search for the BSP tree minimising G_f. Exploring the entire set of possible BSP trees with N elements would take $C_n = \frac{1}{N+1}\binom{2N}{N}$ evaluations at least[1]. As this is not polynomial in N and hence infeasible, heuristics are used. We tested three common randomised algorithms, i.e. genetic algorithms, evolutionary strategies, and a stochastic hill-climber [Michalewicz and Fogel 2002]. The latter was chosen, as it outperformed the others for all tested scenes.

Each algorithm was applied 100 times on 5 sample scenes. The algorithms stopped after 100 iterations. Figure 7.6 shows the results for two sample scenes. The ground truth was evaluated manually based on the retrieved line

[1] C_n is the n^{th} Catalan number.

segment correspondences. The overall cost function G_f is minimised by evaluating the best insertion order of the partitioning lines in the BSP tree.

The genetic algorithm considers a pool with a certain population of genomes (insertion orders). In our case, we start with a population of 50 randomly selected individuals. At each iteration, the 10 best and worst individuals are determined by means of the cost function G_f. New genetic combinations are found through crossover and mutation at each iteration. Only the 10 best individuals are considered for crossover. Crossover means that two genomes are split in each two parts at the same random location. Then the parts are crossed resulting in two offsprings. An offspring can therefore inherit a "good gene" from both parents. The 10 worst genomes are discarded, and the remaining genomes are mutated. This is carried out by swapping two randomly selected genes. After the 100 iterations, the best solution is selected.

The evolutionary strategy is basically a genetic algorithm without the step of crossovers. The fact that it performs better than the genetic algorithm is most likely due to the doubtful effect of crossovers. A drawback of the evolutionary strategy is that it may be trapped in a local minimum as it considers only solutions with a lower cost function.

The stochastic hill-climber aims at escaping local minima by choosing a better solution only with a certain probability. It selects randomly a initial insertion order v_c and a new insertion order v_n in the neighbourhood of v_c. This is achieved by randomly swapping two elements (partitioning lines). It then iteratively compares v_c and v_n based on a probability measure that is per definition $p = (1 + \exp(\frac{G_f(v_n) - G_f(v_c)}{T}))$ [Michalewicz and Fogel 2002]. We chose $T = 1.5$ which results in a 88% chance of choosing the better solution if the difference in the cost function G_f is -3 (a new trihedral junction). If this probability is higher than a normal distributed random probability, v_n becomes the current insertion order v_c.

The algorithm terminates after a maximum of 100 iterations or if there is no change during 20 iterations. This proved to be enough for our experiments to identify most junctions. After the segmentation, we end up with a hypothetical set of matched junctions and planar polygons. The former can be used for the estimation of the fundamental matrix.

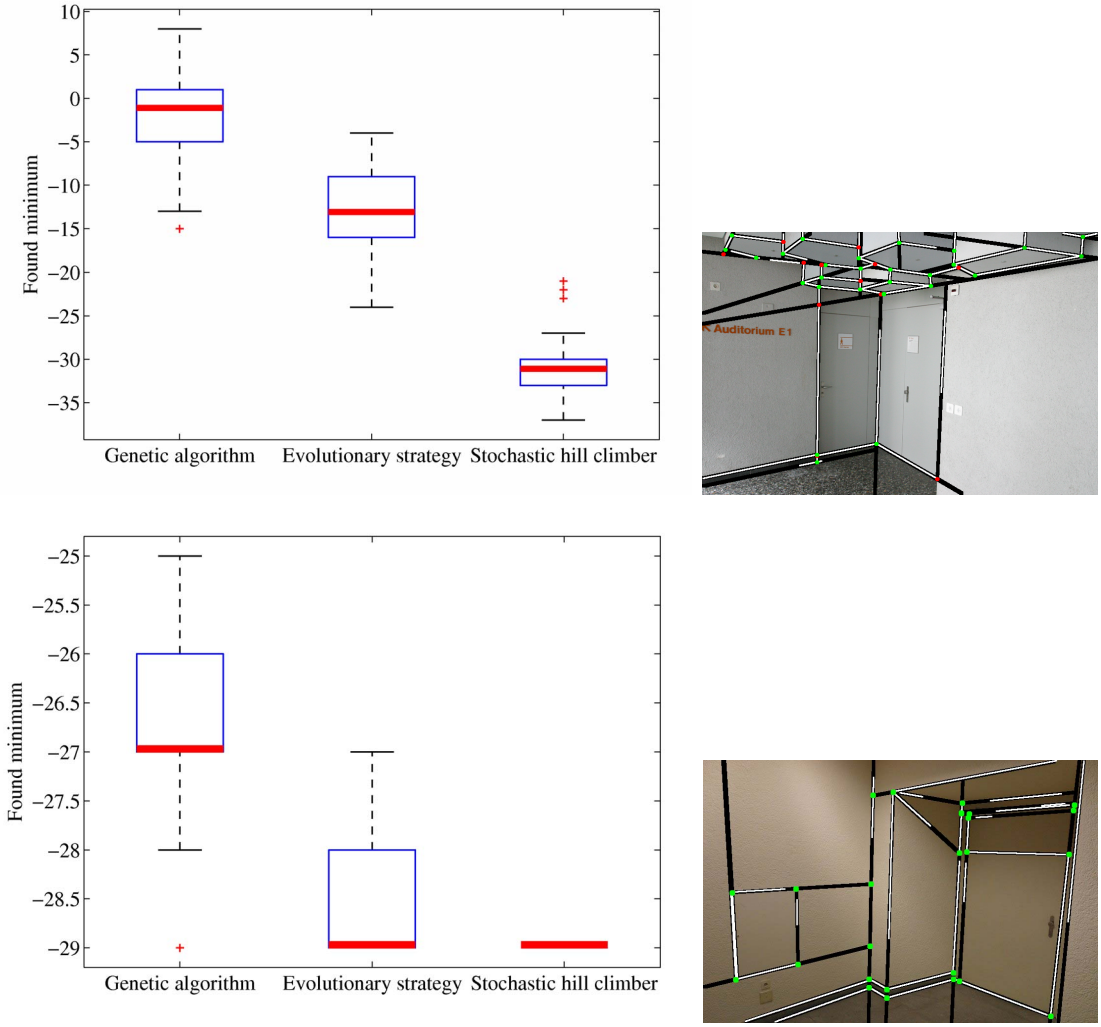

Figure 7.6: Comparison of three heuristics. Each algorithm was ran 100 times for each sample scenes. The result after 100 iterations was considered. The ground truth of the minimum is -36 for the hexagon scene (top left) and -29 for the door scene (bottom left). The thick line in the middle of the box (in the graph on the right side located on the bottom of the boxes) represents the median. The box encompasses the interval from the lower to the upper quartile. The whiskers represent the lowest and the highest values that are not outliers (crosses). On the right side of each plot, a typical solution is shown.

7.4 Epipolar Geometry Estimation

Our next step is to use the detected junctions of the previous section – which come with their corresponding points in the second image – to extract the fundamental matrix. As we deal with piece-wise planar scenes, the presence of a

dominant plane is probable. This may happen when the majority of the point correspondences lie on one physical plane. Using a classical RANSAC scheme [Hartley and Zisserman 2004] for the estimation of the epipolar geometry, as explained in section 6.2, could yield a fundamental matrix which is only valid for the points on that specific plane. The points that lie outside the plane are considered as outliers and discarded. The resulting fundamental matrix cannot be used for 3D reconstruction, as it contains only planar information.

The method proposed by [Chum *et al.* 2005] solves that problem by automatically switching between the standard seven-point (see section 6.2) and a plane-and-parallax solution [Irani and Anandan 1996, Hartley and Zisserman 2004]. There, six matches (4 coplanar and 2 other points) are enough to estimate the fundamental matrix. As it is the case for the standard seven-point algorithm, a random sample of 7 point correspondences is selected at each iteration. If 5 or more correspondences of the sample are consistent with a homography, the plane-and-parallax algorithm is used to robustly estimate the epipolar geometry.

With the fundamental matrix estimated, see figure 7.7-7.9 for examples, the junction detection is repeated with a different cost function based on the epipolar constraint. This additional execution provides a higher number of junctions and extracts mostly correct planar polygons. This is explained next.

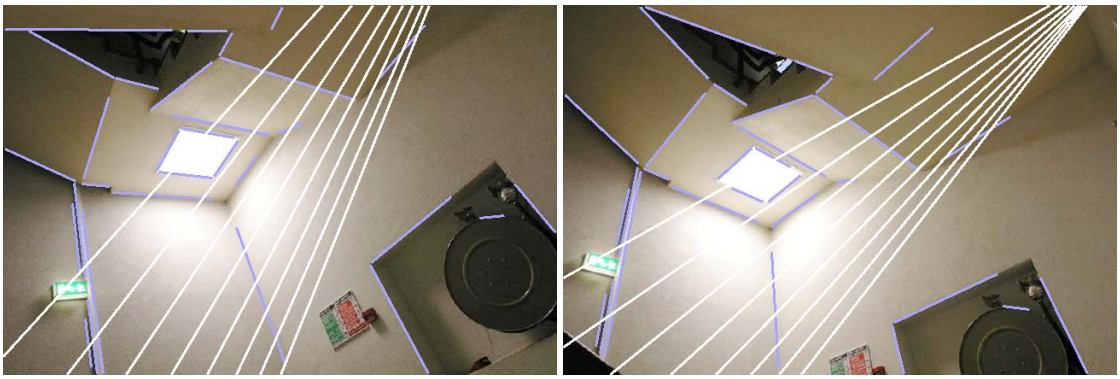

Figure 7.7: Estimated epipolar geometry for the staircase scene.

The epipolar geometry is a much stronger constraint than the Harris response map. Therefore, R_H is no longer considered for the junction detection. The distance of a given junction to its epipolar line $d_{\mathbf{c}}$ in both images is taken as a measure of quality. Y junctions have been proved to be robustly detected even

without any geometrical constraints. Thus, the tolerated deviation of Y junctions to the epipolar line is rather coarse (a value of 4 for 800×600 images). V and T junctions however are less reliable, especially in the case when one of the involved lines is near an epipolar line. Hence, we introduce a weighting term $(d_{\mathbf{c_{V,T}}} < 1.5 \sin \alpha)$ to restrict the epipolar constraint according to the minimal angle α between the participating lines and the epipolar line. These considerations yield the following new cost function for the segmentation selection.

$$f(\mathbf{C}) = \begin{cases} -3 & \text{if } d_{\mathbf{c_Y}} < 4 \\ -1 & \text{if } d_{\mathbf{c_{V,T}}} < 1.5 \sin \alpha \\ 1 & \text{otherwise} \end{cases} \tag{7.5}$$

The values 1.5 an 4 were chosen based on the variance of the distances between the detected junctions and their epipolar lines. Moreover, they were verified experimentally. Again, the exact values for the weights were chosen empirically, and they have no important impact on the results as long as the weights for the Y junctions are clearly smaller than the one for the V and T junctions.

7.5 Bundle Adjustment

The cameras are calibrated with the approach presented in the previous chapter. The camera calibration resulting from line intersections is often not very accurate. In general, a bundle adjustment is not applicable to line segments

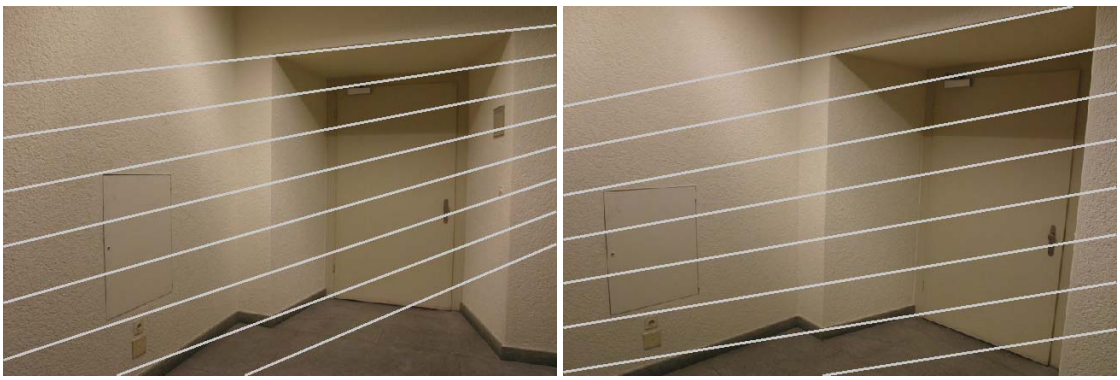

Figure 7.8: Estimated epipolar geometry for the door scene.

Figure 7.9: Estimated epipolar geometry for the corner scene.

in two views, due to their inaccurately detected end points and the number of unknowns compared to the number of known parameters. However, when considering the additional connectivity information, gained from junctions, it becomes possible. In our case, this can be seen as the optimisation of a connected wireframe model against the detected line segments. The wireframe model is obtained by connecting the individual line segments in 3D space at the location of junctions.

For that task, we consider the 3D locations of the junctions and the camera parameters as unknowns. As the junctions are not the result of direct image measurement, their reprojection errors should not be considered as residuals. Instead, the junctions are used to form scene lines, whose reprojection $\mathbf{l}'$ is compared against the original line segment end points by measuring their orthogonal distances to $\mathbf{l}'$. These distances form the residuals.

7.5.1 Junction Parametrisation

A line segment, fully defined by two points in 3D, forms a one-dimensional parameter space. If a line segment votes for a vanishing point, a single point and the direction with unit length induced by the vanishing point suffice. As explained in section 5.4.4, the vanishing point is the projection of the intersection of parallel lines with a certain direction $\mathbf{d}$ at infinity. When the camera parameters are known, this direction $\mathbf{d}$ can be easily estimated. These properties make it possible to define additional junctions, incident with such a fully determined line, by a single parameter instead of three. As the stability of the

bundle adjustment increases by removing gauge freedoms, we aim at applying as many parametrisations as possible.

This yields two conditions. First, each junction is either involved in defining a line (parameter space) or parametrised by a defined line. Second, each line defined by two junctions can not parametrise more than $N_C - 2$ of its incident junctions, where N_C is the total number of incident junctions. If the line votes for a vanishing point, this is $N_C - 1$. As junctions are incident with more than one line, we need to find an assignment of junctions to lines that minimises the total number of parameters for the bundle adjustment. This problem can be cast into a graph formulation, where we seek for the maximum flow from the source s to the destination d, see figure 7.10. At each edge, the flow, an integer number, is bounded by the capacity. This capacity is used to model the above conditions.

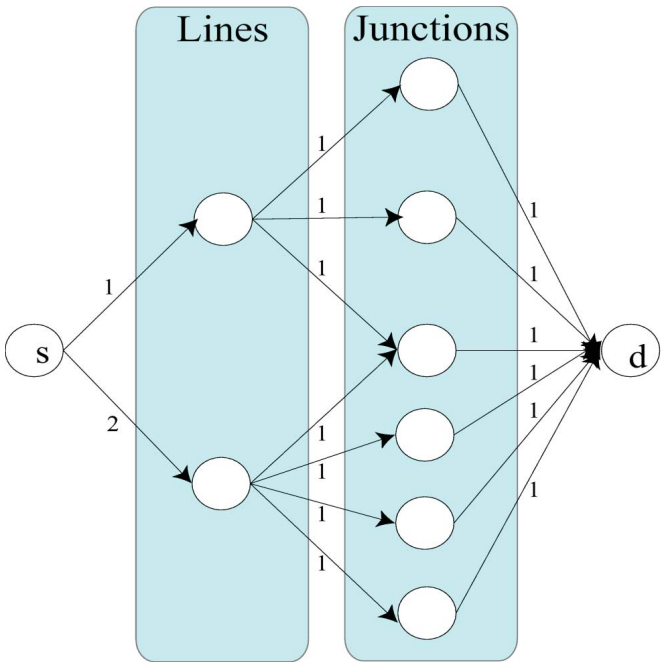

Figure 7.10: Example of a dependency graph for finding parametrised junctions. Lines (left column) are connected to incident junctions (right column). The capacity of edges from the source s indicate how many junctions can be parametrised by a line. The capacity of edges to the drain d (always 1) guarantees that each junction is only parametrised by a single line at the time.

The graph can be split into three sets of edges. The first set is comprised of the edges between the source and the set of all lines. The capacity is chosen $N_C - 2$, encoding the condition that each line can not parametrise more than

$N_C - 2$ of its N_C incident junctions. An exception are the lines defined by a vanishing point direction, which have a capacity of $N_C - 1$. The second set indicates which lines are incident with which junctions. The third set is between all junctions and the destination vertex. The capacity of 1 for the edges of the second and third set encodes the first condition that each junction must be parametrised by at most one line. Finding the maximum flow starting from the source corresponds to finding the maximum number of fulfilled conditions, and hence a minimum number of total parameters. The maximum flow is determined using the Edmonds-Karp algorithm, cf. e.g. [Cormen *et al.* 1990].

After running the algorithm, a flow of 1 between a line and a junction indicates the parametrisation of the junction by the connected line.

7.5.2 Camera Parameters

Instead of using all eleven values of the projection matrix, we reduce this number via a quaternion $\mathbf{q} = (\phi_0, \phi_1, \phi_2, \phi_3)$ for the representation of the rotation matrix [Horn 1987]. The estimation of a quaternion is computationally less expensive than deriving the rotation angles from the Eulerian rotation matrices. Moreover, there are no ambiguities concerning the sequence by which the rotations are applied, and hence no problems with calculating the inverse or derivatives. $\mathbf{q}$ can be calculated from R using equations (7.6)–(7.9).

$$\phi_0 = \frac{\sqrt{r_{11} + r_{22} + r_{33} + 1}}{2} \tag{7.6}$$

$$\phi_1 = \frac{r_{32} - r_{23}}{4\phi_0} \tag{7.7}$$

$$\phi_2 = \frac{r_{13} - r_{31}}{4\phi_0} \tag{7.8}$$

$$\phi_3 = \frac{r_{21} - r_{12}}{4\phi_0} \tag{7.9}$$

Hence, this gives us, including the translation, seven unknowns for the camera related part of the bundle adjustment step.

The sum of the squared residuals is minimised using Levenberg-Marquardt. An example for the improvement achieved by the bundle adjustment is shown in figure 7.11.

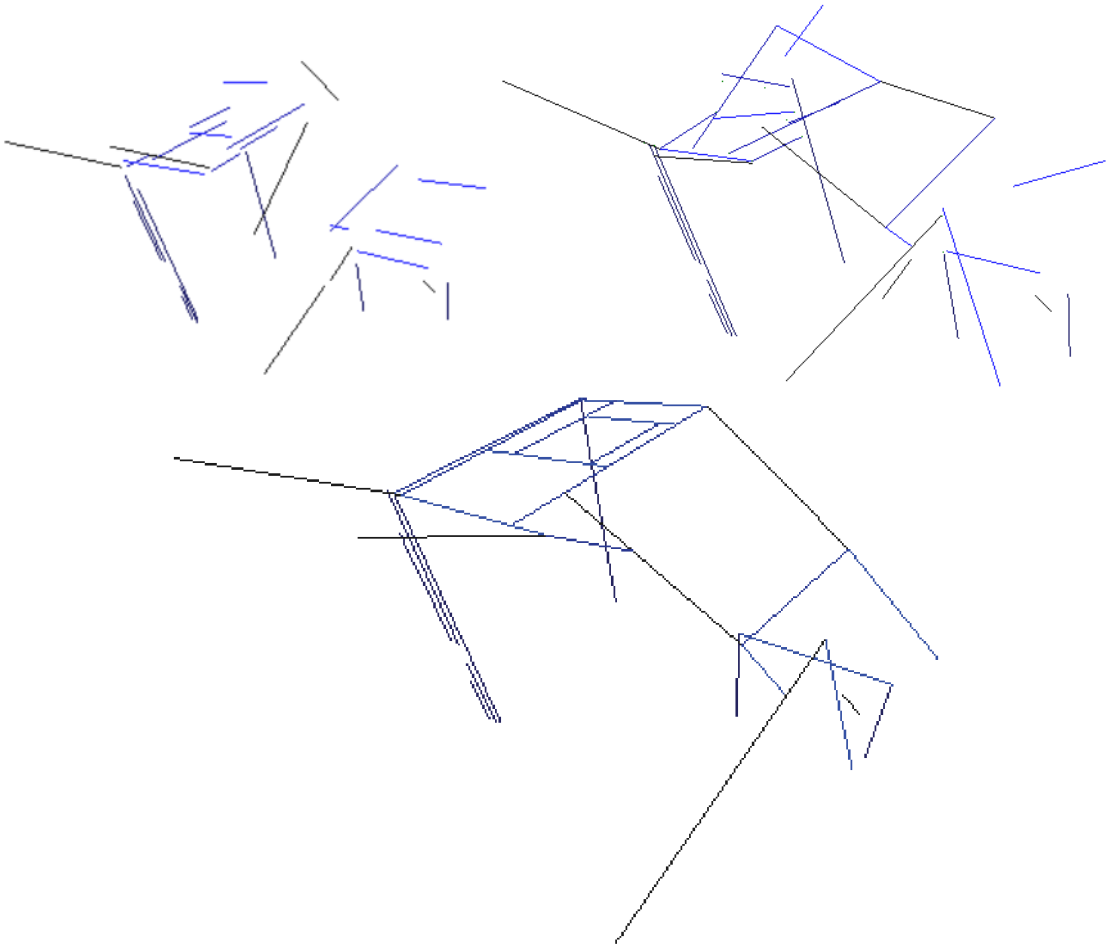

Figure 7.11: Top left: Metric reconstruction of the staircase scene before the additional information of junctions. The scene information is unusable for coplanar grouping. Top right: reconstructed wireframe model after detection of junctions and with connected lines. Bottom: reconstruction after the bundle adjustment.

7.6 Reconstruction

With the improved camera calibration and the scene segmented into planar polygons, the piece-wise planar reconstruction can take place. It consists of finding the *support lines* (defined below) of a polygon. All partitioning lines involved in forming a polygon $p \in \mathcal{P}$ are potential support lines.

In the best case, all these lines $\mathcal{L}_p$ can be used to create the corresponding scene plane and the respective delineation. In practice however, the reconstructed lines are not always coplanar, due to either occlusions or numerical errors, see figure 7.12. Hence, there are multiple possible subsets $\mathcal{T}_p^{(i)} \subset \mathcal{L}_p$ of two or

more lines to form different, but geometrically possible planes for a polygon. We call these lines the support lines.

Figure 7.12: Multiple solutions exist to form valid planes from two or more of the bordering lines of a polygon (black). The correct solution would include lines B, C and D. However, due to the fact that the corner on the right side of the door frame is occluded by the wall, another incorrect but geometrically valid solution would be the plane defined by the lines A and C. The planes formed with A and B, or A and D are geometrically invalid. There is not enough texture available for a pixel-wise comparison.

In order to find the best subsets of support lines, and to reject invalid subsets, we consider the homography transfer error $h_t(\mathcal{T}_p^{(i)})$ of the end points from the first image to the second to determine all valid subsets for each polygon. A subset is valid if this error is below a certain threshold. We allow a maximum total transfer error of 5 pixels.

$$h_t(\mathcal{T}_p^{(i)}) = \sum_{\mathbf{l} \in \mathcal{T}_p^{(i)}} d(\mathbf{H}\mathbf{a_1}, \mathbf{a_2}) + d(\mathbf{H}\mathbf{b_1}, \mathbf{b_2}), \tag{7.10}$$

where $\mathbf{a_{1,2}}$ and $\mathbf{b_{1,2}}$ are the two end points of a detected and merged line segment $\mathbf{l} \in \mathcal{T}_p^{(i)}$ in I_1 and I_2, respectively.

To obtain a scene with a low number of discontinuities, each polygon p is assigned a subset from $\mathcal{T}_p^{(i)}$ such that neighbouring polygons share lines whenever possible. This is done using a simple iterative greedy strategy. For each polygon p, it tries all possible subsets $\mathcal{T}_p^{(i)}$. The subset that shares its lines with the highest number of polygons is chosen. This process is repeated as long as there is a positive change in the total number of shared lines. If in one iteration, no polygon finds a subset increasing the number of common lines (lowering the number of discontinuities), the algorithm is terminated.

In the final 3D model, the delineations of the planes are provided by the bordering lines of the polygons found during the segmentation. This information is already available and avoids intersecting the individual planes.

7.7 Results

We tested our method on some challenging examples. Figure 7.13 shows the 3D reconstruction of the scene that features the ceiling of a staircase. This scene is interesting because some planes are almost parallel to each other. It shows clearly that our approach is not dependent on mutually orthogonal planar patches, but can handle more general kinds of piecewise planar scene.

Figure 7.14 shows two wide-baseline images of a door. Again, not all planes are completely orthogonal. Moreover, it is a good example for the kind of scenes that can be handled with our method. It contains almost no texture at all and the viewpoint change is important. Traditional structure-from-motion approaches [Hartley and Zisserman 2004] are not able to provide any solution for such kind of extreme cases.

The red-door scene of figure 7.15 consists of mutually orthogonal planes, but it is less textured than all the other scenes and therefore clearly shows the performance of the system. Note that due to the perspective distortion in the shown images (not in the 3D reconstruction), the walls do not appear orthogonal. However, even if the information is quite limited by the low number of features, the use of only two views, and the constraints on the cameras, our results show a high accuracy. This is mainly due to the bundle adjustment step (see figure 7.11). Table 7.1 shows some quantitative results.

Figure 7.13: Final 3D reconstruction of the stair case scene (see figure 7.7). A few gaps remain, but subsets were assigned correctly.

The corridor in figure 7.16 shows the level of detail that is provided by our method. The details of the door frame on the right hand side are perfectly reconstructed. Even though this scene contains some textured patches, they could not have been matched correctly using interest point detector/descriptor, as they are not distinctive enough (repeated objects). The most difficult case for our method is represented in figure 7.17. The reconstruction of the small hexagons are a big challenge, as they are responsible for occlusions and, again, repeated objects.

figure:	7.13	7.14	7.15	7.17
ang.dev.$[^\circ]$ before:	2.5	5.3	4.7	2.6
ang.dev.$[^\circ]$ after:	2.4	3.3	2.1	2.2
leng.dev.$[\mathrm{cm}]$ before:	-	6.1	5.4	1.3
leng.dev.$[\mathrm{cm}]$ after:	-	4.9	4.6	1.8

Table 7.1: Standard deviation for the reconstructed angles and lengths, before and after bundle adjustment. Notice that some lengths could not have been measured: figure 7.13 (no access to the ceiling structures). For these scenes, we only compared the obvious orthogonal angles, but notice that we do not impose orthogonality as a constraint in our method.

In general, for the bundle adjustment step, we observed a reduction of the number of unknowns of about 25% by the application of our max-flow algorithm. This important reduction has a remarkable impact on the quality of the reconstruction (see figure 7.11 and table 7.1).

The actual algorithm is to some extent robust against mismatches, as wrongly matched lines are not expected to yield junctions. The computation time of the presented examples lies between 2–4 minutes each. The same settings were used for all scenes.

Figure 7.14: Top row: Input images of the door scene. Bottom row: Two views of the final 3D reconstruction.

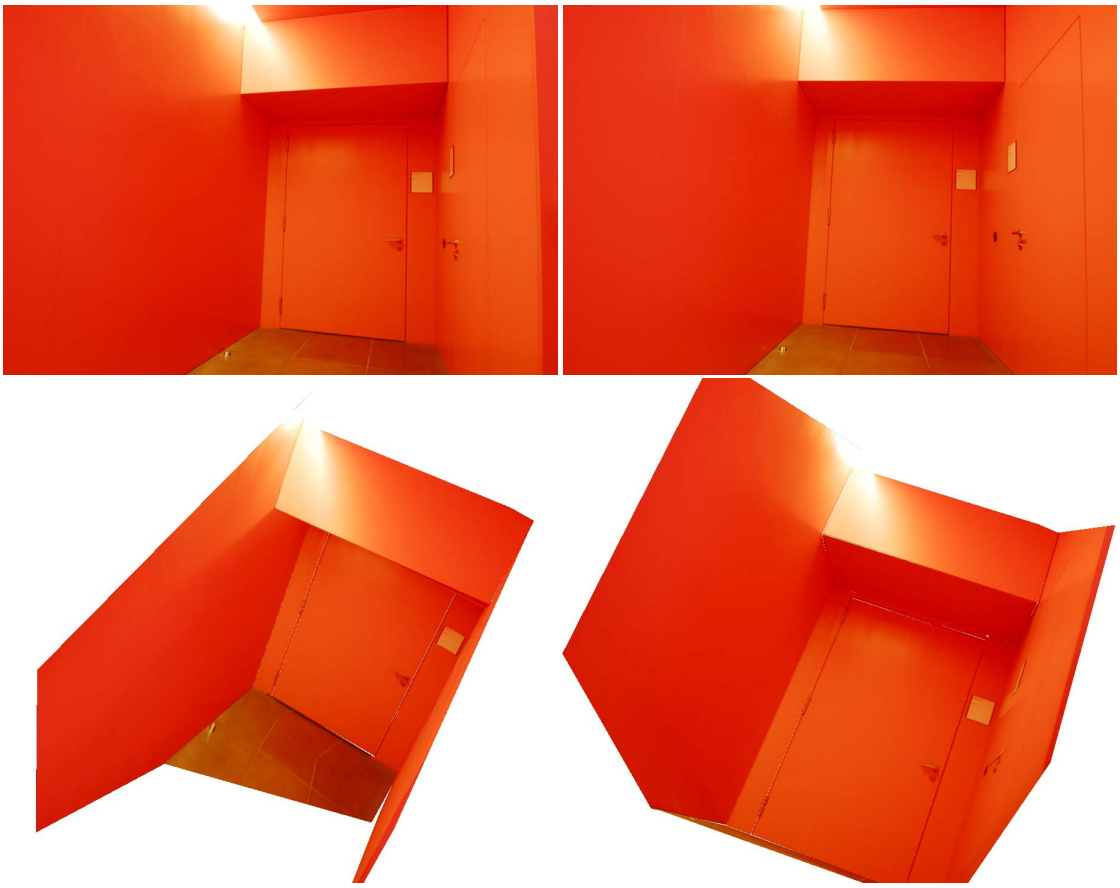

Figure 7.15: Top row: Input images of the red-door scene. Bottom row: Two views of the final 3D reconstruction.

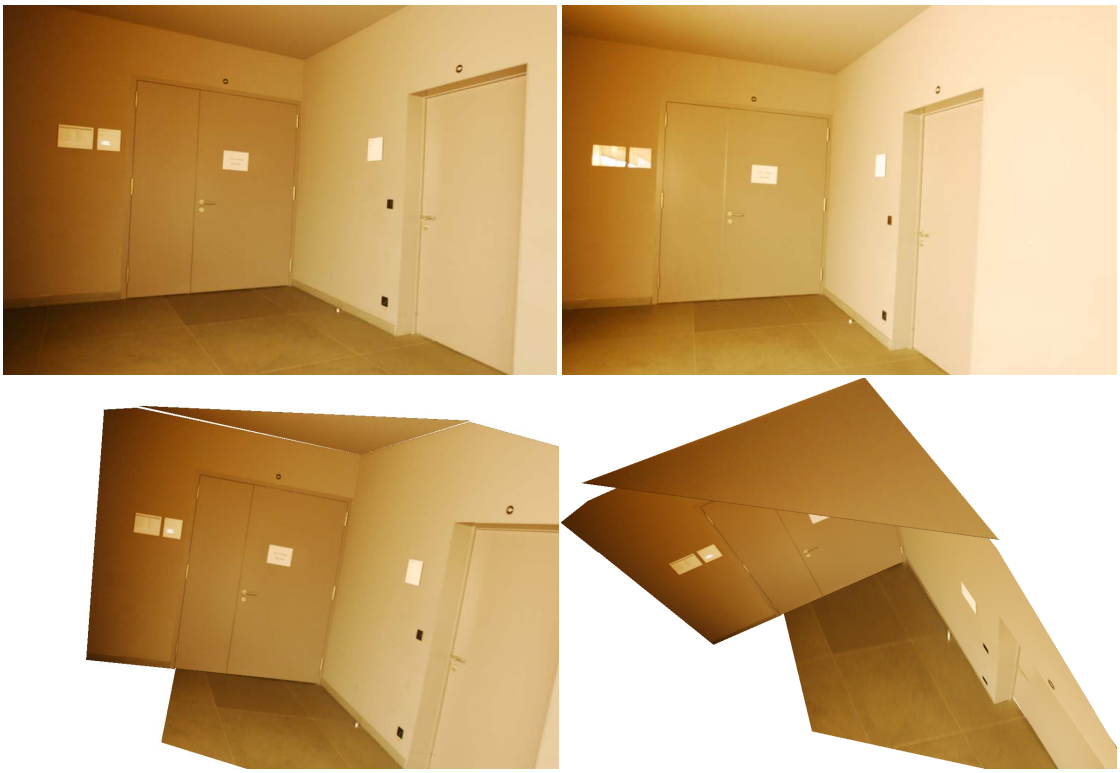

Figure 7.16: Top row: Input images of the corridor scene. Bottom row: Two views of the final 3D reconstruction. The detail of the door frame on the right side are perfectly reconstructed despite inaccurate line segment detection.

Figure 7.17: Top row: Input images of the hexagon scene. Bottom row: Two views of the final 3D reconstruction. The hexagons on the ceiling are correctly reconstructed. Some gaps are present, which are mainly due to occlusions.

7.8 3D From Points and Line Segments

In many cases, the simultaneous use of points and line segments is very helpful for the quality of the outcome of our method. As already shown in chapter 5, the combination of line segments and interest points yields at the same time fewer mismatches and more correct matches. Also, for the estimation of the fundamental matrix, a higher number of interest point correspondences (SURF features or junctions derived from matched line segments) increases the chances of success.

For the majority of the presented examples, the number of extracted and matched interest points is far below the required number for the estimation of the fundamental matrix (7 interest point correspondences). This is due to the absence of texture in most of these scenes. Some examples, like the staircase scene (figure 7.18) provide sparse locally textured regions that result in a sufficient number of correctly matched interest point correspondences for the estimation of the fundamental matrix. However, as we are dealing with piecewise planar scene and the textured patches may lie all on a single plane, the spatial information provided by the latter is often not sufficient. Therefore, the epipolar geometry estimated from interest point correspondences alone is often not correct (see figures 7.18 and 7.19 top).

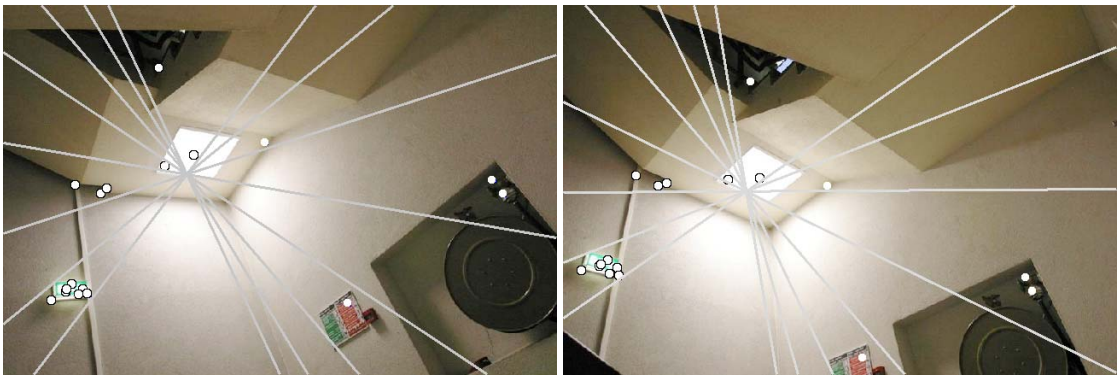

Figure 7.18: False epipolar geometry estimation for the staircase scene using 17 interest point correspondences alone.

The simultaneous use of interest point correspondences and junctions derived from line segment matches on the other hand delivers a correct epipolar geometry (see figure 7.19 middle). Moreover, as more points are used to estimate the epipolar geometry, the result is more accurate. As soon as the epipolar

Figure 7.19: False epipolar geometry for the blackboard scene using 18 interest point correspondences alone. The interest points do not provide enough spatial information in order to estimate a correct epipolar geometry.

geometry is known, the focal length of the camera can be determined given our assumptions about the camera intrinsics, and a piecewise planar 3D model can be constructed (see figure 7.19 bottom). The blackboard scene is not only challenging because of the interest point correspondences that are all located in a single plane, but also because it more or less contains only two dominant

planes. This is the minimal number of planes a piecewise planar scene must contain in order to make sense for camera calibration and 3D reconstruction. Furthermore, the scene is a challenging example as the cupboards on the right hand side contain repeated objects and the doors show specular reflection artefacts.

The simultaneous use of interest points and line segments is also possible to a certain extent for the bundle adjustment step. However, the more unparametrised interest points that are considered for the minimisation of the reprojection error, the higher the number of unknowns. This may result in a less stable solution or even in a solution that does not converge anymore. For that reason, we did not consider such unparametrised interest points for the bundle adjustment.

7.9 Conclusion and Outlook

This chapter tackled the problem of automatic structure-from-motion of poorly-textured indoor scenes from two uncalibrated images, some viewed under wide-baseline conditions. We proposed a novel method for planar segmentation and junction detection at the same time. An existing camera self-calibration scheme was applied to derive the external camera parameters as well as the focal length. We then showed a new strategy for bundle adjustment using a maximum-flow algorithm to reduce the number of unknowns and therefore, render the process far more stable or even possible. These methods enable the 3D reconstruction of such difficult scenes.

We also showed the advantage of using interest point and line segment correspondences in a unified manner. Future work could aim at estimating a dense 3D model using the piecewise planar hypothesis and traditional dense stereo methods at the same time for more complete 3D reconstructions.

Possible improvements would be to test the performance of the method for multiple views. Furthermore, traditional structure-from-motion methods could be extended with our method in order to broaden the range of possible scenes for dense 3D reconstruction. The system could automatically switch to the additional use of line segments as soon as not enough interest point correspondences were found. This would result in a broader range of scenes to be tackled.

A

Fast Non-Maximum Suppression

The detection of local maxima is part of multiple computer vision applications. This process is commonly called *Non-Maximum Suppression* (NMS). It is widely used for the detection of interest points in image space. Recently, the demand for scale-invariant versions of such detectors [Bay *et al.* 2006, Brown *et al.* 2005, Lowe 2004, Mikolajczyk and Schmid 2002] has increased. Their localisation is performed over a three-dimensional space (image- *and* scale-space). Hence, efficient implementations of NMS are required.

A.1 Straightforward Implementation

In general, the NMS can be summarised in a few words. A given pixel p is considered a local maximum, if the intensities of the pixels in a certain neighbourhood around p are smaller than the intensity value of p. The neighbourhood of p consists in the 1D case of the N pixels on its left and right side (referred to as a $(2N+1)$-neighbourhood), in the 2D case of the quadratic $(2N+1)\times(2N+1)$ region, and in the 3D case of the cubic $(2N+1)\times(2N+1)\times(2N+1)$ region centred around p. For a better comprehensibility, we first study the 1D case which is then extended to solve the 2D and 3D case.

The straightforward implementation of NMS consists of two nested loops. The outer loop sifts through all the pixels incrementally. The inner loop compares the $2N$ neighbours with the current pixel of the outer loop (central pixel). As soon as a neighbouring intensity exceeds the intensity of the central pixel, the inner loop is aborted and the next pixel is considered by the outer loop. This is the mostly used implementation of the NMS. However, it is far from being

optimal. The expected number of comparisons per pixel is of $1 + \ln(2N)$. For the worst case, the complexity is even higher, i.e. $O(N)$.

A.2 Block Algorithm

Based on the fact that two local maxima are separated by at least $N + 1$ pixels, a more efficient NMS algorithm partitions the signal into blocks of length $N + 1$ (still considering the 1D case). As there can be only one local maximum in such a block, the algorithm searches within each block for the pixel with the highest intensity value. Then, the full $(2N + 1)$-neighbourhood of this pixel is tested for an element with a higher intensity value. Hereby, the block, containing the candidate pixel itself, can be skipped, because all its elements are by construction already smaller than the considered pixel.

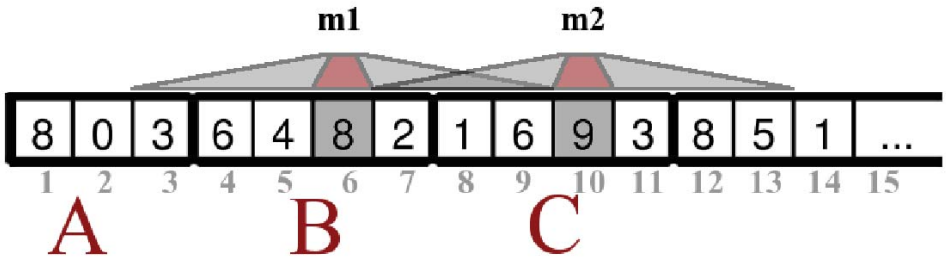

Figure A.1: Functional example of the 1D block algorithm. Two maxima **m1** and **m2** are detected in a 7-neighbourhood.

Figure A.2 shows a functional example for NMS with $N = 3$. The first block $\mathbf{A}$, which of size N only, is used to the second block with. It is therefore not meant to contain a local maximum (border conditions). The second block $\mathbf{B}$ contains a maximum $\mathbf{m1}$ on position 6. In order to confirm it as a local maximum for a 7-neighbourhood, only position 3, 8 and 9 have to be considered. In block $\mathbf{C}$ is another maximum located. Again, only position 7, 12 and 13 have to be considered for a final decision. One could imagine to enhance this algorithm even further by storing the previous block-maximum ($\mathbf{m1}$ in our case). A given candidate ($\mathbf{m2}$) is only compared to the values of the elements in the previous block (position 7), if the previous maximum ($\mathbf{m1}$) is higher than the candidate ($\mathbf{m2}$). However, in our case, where N is small, the profits in efficiency are outweighed by the time used for memory manipulations.

Algorithm	worst case	average case
1D Straightfwd	N	$\approx 1 + \ln(N) + \ln(2)$
1D Block	$2 - \frac{1}{N+1}$	$\approx 1 + \ln(2 - \frac{1}{N+1})$
2D Straightfwd	N^2	$\approx 1 + 2\ln(N) + \ln(2)$
2D Block	$4 - \frac{4}{N+1}$	$\approx 1 + 2\ln(2 - \frac{1}{N+1})$
3D Straightfwd	N^3	$\approx 1 + 3\ln(N) + \ln(2)$
3D Block	$\approx 8 - \frac{12}{N+1} - \frac{6}{(N+1)^2}$	$\approx 1 + 3\ln(2 - \frac{1}{N+1})$

Table A.1: Worst- and average-case complexities for different implementations of the NMS.

The average-case complexity, $1 + \ln(2 - \frac{1}{n+1})$, of the 1D block algorithm is clearly smaller than the one for the straightforward algorithm. The same is true for the worst-case complexity i.e. $2 - \frac{1}{n+1}$. Upgrading the block algorithm to the 3D case, the differences become even clearer (see table A.1).

B

Quadratic Interpolation

Interpolation is widely used for interest point localisation with sub-pixel accuracy. After the non-maximum suppression, the found maximum p_0 is subject to a quadratic interpolation considering the direct neighbours in image space and scale in order to find the 'real' position $\mathbf{x}_m$ of the interpolated extremum m. Here, we consider the one-dimensional case in order to show that m cannot be located farther than half-way the distance from the detected maximum p_0 to one of its direct neighbours p_{-1} and p_1.

Given the extracted maximum p_0 at location $x = 0$, and its direct neighbour intensities p_{-1} at $x = -1$ and p_1 at $x = 1$ (see figure B.1). As p_0 is a local maximum, its intensity is always higher than the neighbouring intensities p_{-1} and p_1.

In order to retrieve the interpolated extremum, a parabola of the form $p(x) = ax^2 + bx + c$ is fit through the detected intensities. Therefore, the parameters of the parabola are written as

$$a = \frac{p_1 - 2p_0 + p_{-1}}{2} \tag{B.1}$$

$$b = \frac{p_1 - p_{-1}}{2} \tag{B.2}$$

$$c = p_0. \tag{B.3}$$

It is straightforward to see that the extremum of the parabola $p(x)$ is located at the position x_m

$$x_m = \frac{p_{-1} - p_1}{2(p_1 - 2p_0 + p_{-1})} \tag{B.4}$$

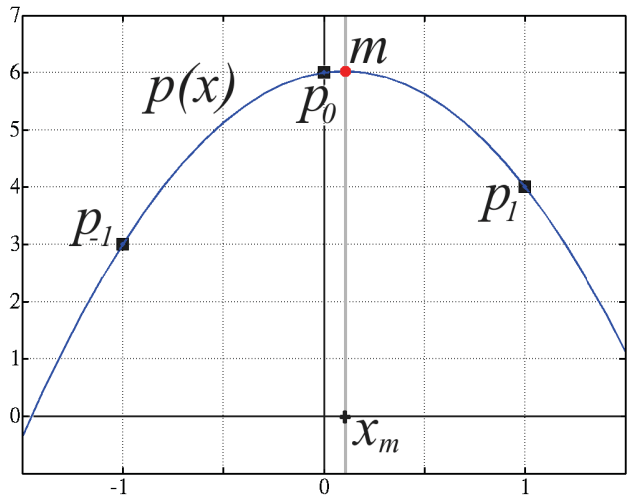

Figure B.1: Quadratic interpolation with a parabola through the detected maximum p_0 and its direct neighbours p_{-1} and p_1. The parabola's extremum m is located at x_m.

In order to determine the interval of equation (B.4), we set the detected maximum p_0 at the fix value 1 and one of the neighbouring intensities close to p_0, i.e. $p_0 - \epsilon$, where ϵ is a infinitely small number. There are two extreme cases. The first one supposes $p_{-1} = 1 - \epsilon$ and p_1 a free parameter. The second one supposes the inverse, i.e. $p_1 = 1 - \epsilon$ and p_{-1} free. Hence, the outer limits l_1 and l_2 of the interval of x_m are determined as follows.

$$l_1 = \lim_{\epsilon \to 0} \frac{1 - p_1 - \epsilon}{2(p_1 - 1 - \epsilon)} = -\frac{1}{2} \tag{B.5}$$

$$l_2 = \lim_{\epsilon \to 0} \frac{p_{-1} - 1 + \epsilon}{2(p_{-1} - 1 - \epsilon)} = \frac{1}{2} \tag{B.6}$$

Therefore, the interval of the extremum resulting from quadratic interpolation is $(-0.5;\, 0.5)$. Notice that this property also holds for the interpolation of extracted minima.

C

Test Images

The test images are of two different types of scenes: structured and textured. They represent different image transformations, like Zoom and Rotation, Image Blur, Affine transformation etc. The purpose is to allow a complete evaluation of the robustness of a given interest point detector and descriptor towards the mentioned types of transformations.

Figure C.1: Graffiti sequence: Affine transformation, in-plane rotation and lighting change

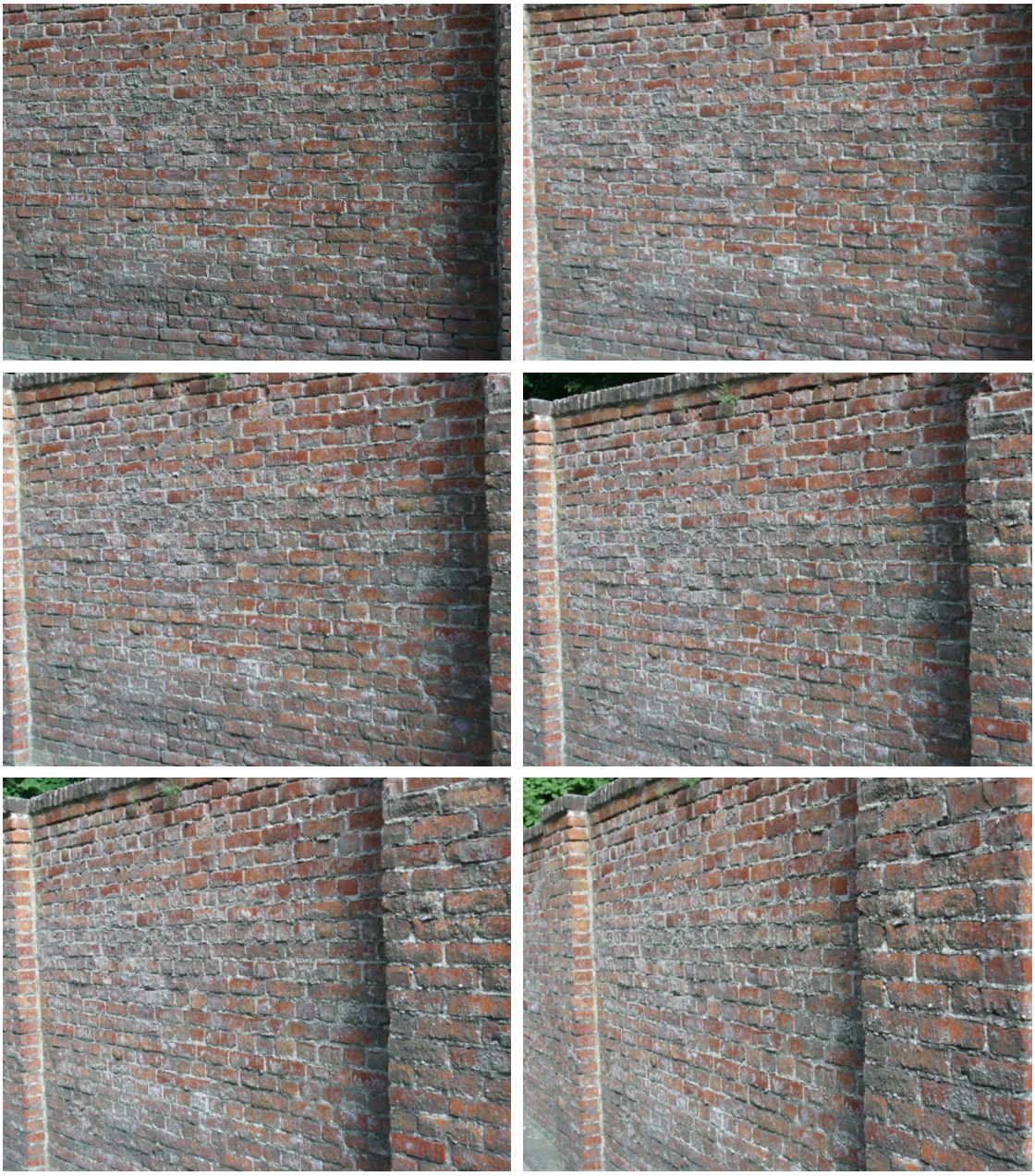

Figure C.2: Wall sequence: Affine transformation

Figure C.3: Boat sequence: Zoom and rotation

Figure C.4: Bark sequence: Zoom and rotation

Figure C.5: Bikes sequence: Image blur

Figure C.6: Trees sequence: Image blur

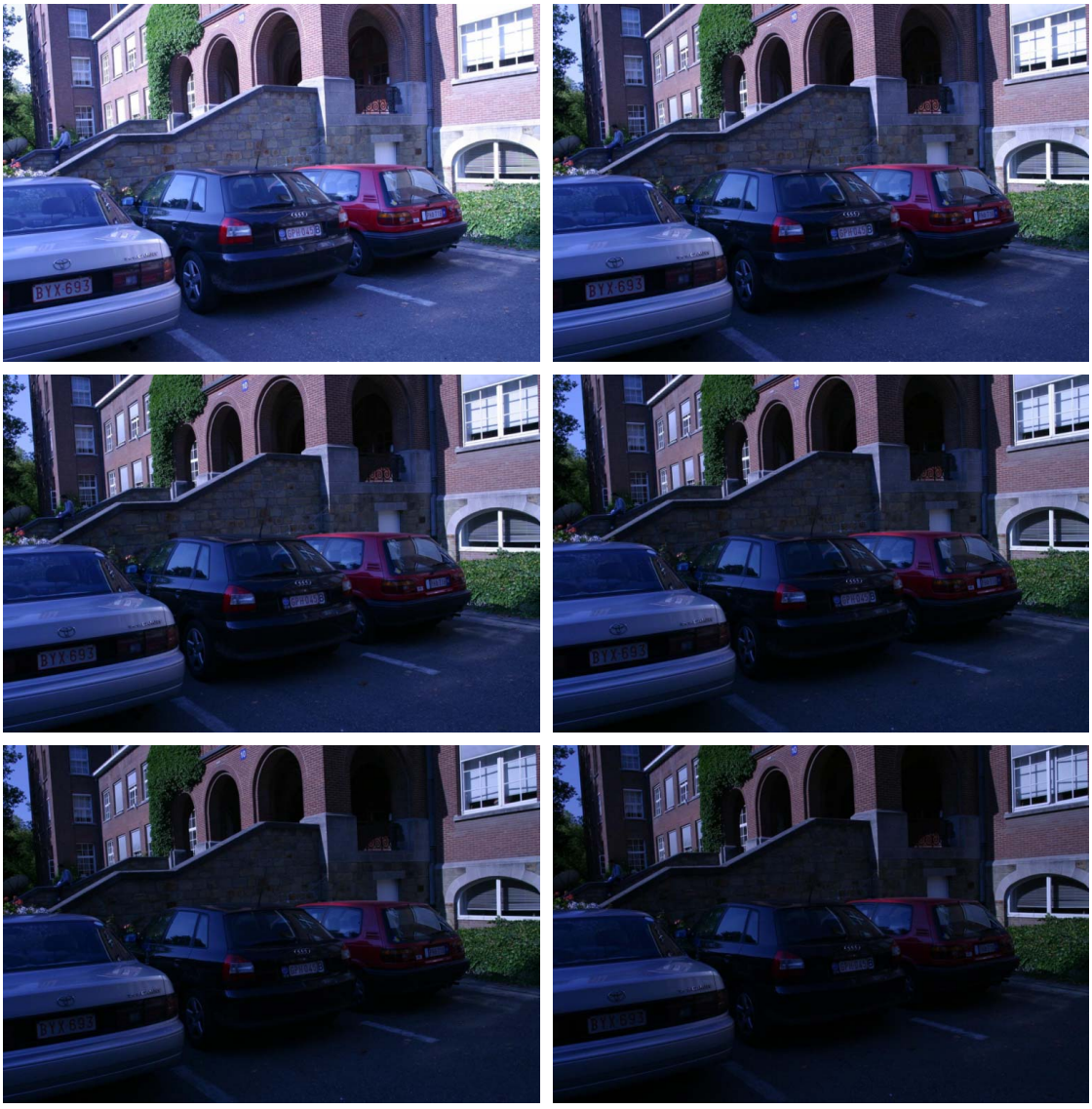

Figure C.7: Leuven sequence: Lighting change

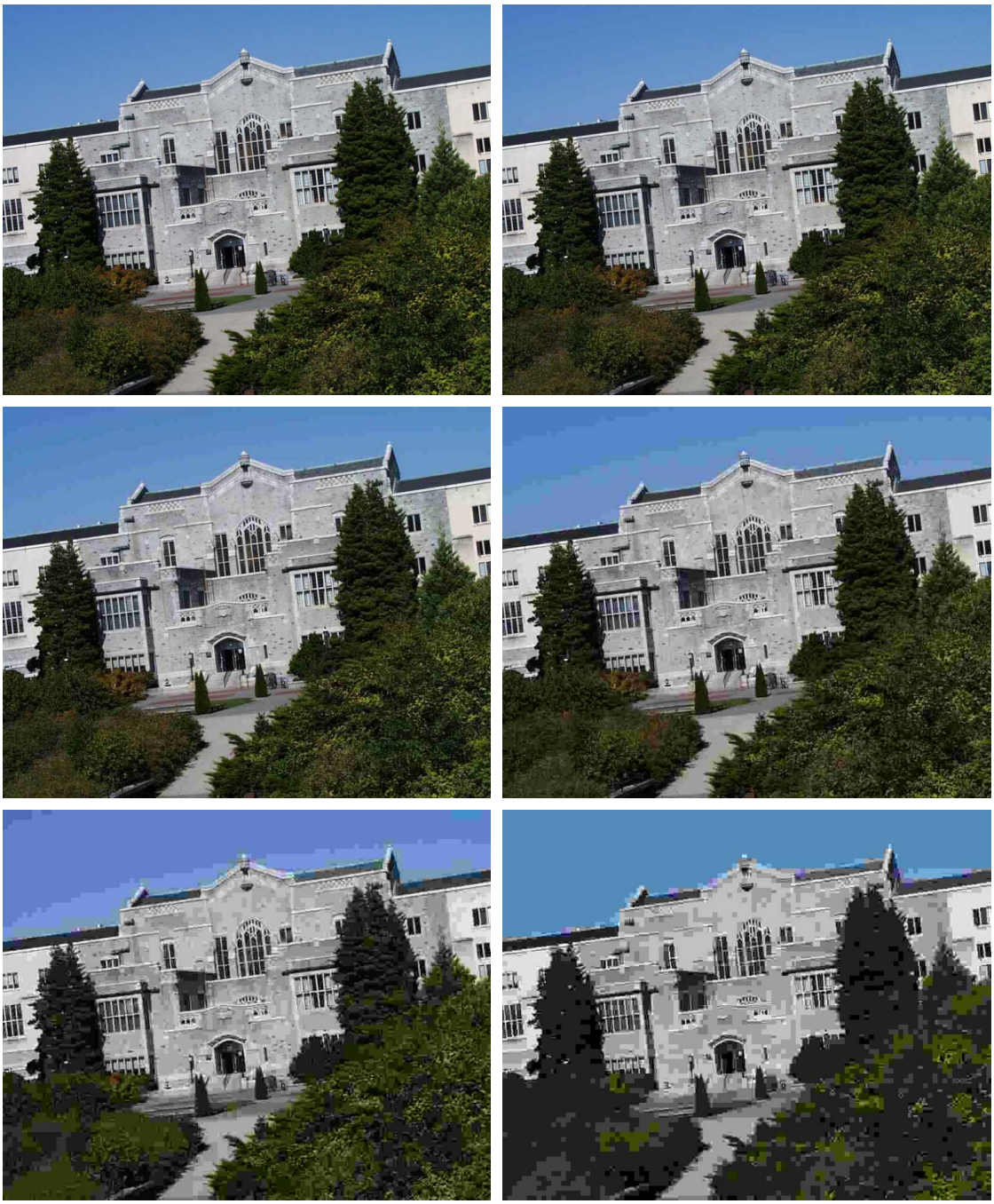

Figure C.8: Ubc sequence: JPEG compression

Bibliography

[Alberti 1977] L. B. Alberti. *On Painting*. Yale University Press, New Haven, UK, revised edition, jul 1977. Translation from "Della Pittura" (1435). 1.1

[Baillard and Zisserman 2000] C. Baillard and A. Zisserman. A plane-sweep strategy for the 3d reconstruction of buildings from multiple images. In *19th ISPRS Congress and Exhibition*, 2000. 7.1

[Bartoli and Sturm 2003] Adrien Bartoli and Peter F. Sturm. Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. *International Journal of Computer Vision*, 52(1):45 – 64, 2003. 7.1

[Bartoli *et al.* 2001] A. Bartoli, P. Sturm, and R. Horaud. Structure and motion from two uncalibrated views using points on planes. In *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling*, pages 83 – 90, June 2001. 7.1

[Bartoli *et al.* 2004] Adrien Bartoli, Navneet Dalal, and Radu Horaud. Motion panoramas. *Computer Animation and Virtual Worlds*, 15:501–517, 2004. 4.2

[Baumberg 2000] Adam Baumberg. Reliable feature matching across widely separated views. In *CVPR*, pages 774 – 781, 2000. 3.1, 4.1.2

[Bay *et al.* 2006] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006. A

[Bellutta *et al.* 1989] P. Bellutta, G. Collini, A. Verri, and V. Torre. 3d visual information from vanishing points. In *Interpretation of 3D Scenes, 1989. Proceedings.,Workshop on*, pages 41 – 49, 1989. 5.4.4

[Blake and Isard 1998] Andrew Blake and M. Isard. *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998. 5.1, 5.3

[Brown and Lowe 2002] M. Brown and D. Lowe. Invariant features from interest point groups. In *BMVC*, 2002. 2.5

[Brown *et al.* 2005] Matthew Brown, Richard Szeliski, and Simon Winder. Multi-image matching using multi-scale oriented patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517, june 2005. 4.2, A

[BTnode] `h`ttp://www.btnode.ethz.ch. 4.1.2

[Burgard *et al.* 1998] W. Burgard, A.B. Cremers, Dieter Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and Sebastian Thrun. The interactive museum tour-guide robot. In *Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 1998. 4.1

[Burns *et al.* 1986] J.B. Burns, A.R. Hanson, and E.M. Riseman. Extracting straight lines. *PAMI*, 8(4):425 – 455, 1986. 5.1

[Burt and Adelson 1983] Peter J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–36, Oct 1983. 4.2.1, 4.2.4

[Can *et al.* 2002] Ali Can, Charles V. Stewart, Badrinath Roysam, and Howard L. Tanenbaum. A feature-based, robust, hierarchical algorithm for registering pairs of images of the curved human retina. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):347–364, 2002. 4.2

[Canny 1986] J Canny. A computational approach to edge detection. *PAMI*, 8(6):679 – 698, 1986. 5.1

[Carneiro and Jepson 2003] G. Carneiro and A.D. Jepson. Multi-scale phase-based local features. In *CVPR (1)*, pages 736 – 743, 2003. 3.1

[Chum *et al.* 2005] O. Chum, T. Werner, and Jiri Matas. Two-view geometry estimation unaffected by a dominant plane. In *CVPR (1)*, pages 772 – 779, 2005. 7.4

[Cipolla *et al.* 1999] R. Cipolla, T. Drummond, and D.P. Robertson. Camera calibration from vanishing points in image of architectural scenes. In *BMVC*, pages 382 – 391, 1999. 5.4.4, 7.1

[Cormen *et al.* 1990] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, Mass., 1990. 7.3, 7.5.1

[Cornelis and Van Gool 2005] Nico Cornelis and Luc Van Gool. Real-time connectivity constrained depth map computation using programmable graphics hardware. In *CVPR (1)*, pages 1099 – 1104, 2005. 6

[Debevec 1996] Paul E. Debevec. *Modeling and Rendering Architecture from Photographs*. PhD thesis, University of California at Berkeley, Computer Science Division, Berkeley CA, 1996. 7.1

[Dick *et al.* 2000] Anthony R. Dick, Philip H. S. Torr, and Roberto Cipolla. Automatic 3d modelling of architecture. In *BMVC*, 2000. 6.3, 7.1

[Faugeras 1993] Olivier D. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993. 6.1

[Ferrari *et al.* 2003] V. Ferrari, T. Tuytelaars, and L. Van Gool. Wide-baseline muliple-view correspondences. In *CVPR*, volume I, pages 718 – 728, June 2003. 5.4.2, 5.4.2, 5.4.2, 5.4.3

[Fischler and Bolles 1981] M.A. Fischler and R.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, June 1981. 4.2.2, 2

[Florack *et al.* 1994] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. General intensity transformations and differential invariants. *JMIV*, 4(2):171–187, 1994. 3.1

[Föckler *et al.* 2005] P. Föckler, T. Zeidler, and O. Bimber. Phoneguide: Museum guidance supported by on-device object recognition on mobile phones. Research Report 54.74 54.72, Bauhaus-University Weimar, Media Faculty, Dept. Augmented Reality, 2005. 4.1, 4.1.6

[Freeman and Adelson 1991] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *PAMI*, 13(9):891 – 906, 1991. 3.1

[Garey and Johnson 1979] M.R. Garey and D.S. Johnson. *Computers and Intractability, a Guide to the Theory of NP-completeness*. Freeman, 1979. 4.2.3

[Harris and Stephens 1988] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, pages 147 – 151, 1988. 2.1, 1

[Hartley and Sturm 1995] R.I. Hartley and P. Sturm. Triangulation. In *6th International Conference on Computer Analysis of Images and Patterns, Prague, Czech Republic*, pages 190–197, Sep 1995. 6.1, 6.3

[Hartley and Zisserman 2004] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 2.6, 4.2.2, 5.4.4, 6, 6.2, 6.3, 6.3, 6.3, 6.4, 7.4, 7.7

[Hartley 1993] R.I. Hartley. Cheirality invariants. In *DARPA93*, pages 745–753, 1993. 6.3

[Hartley 1994] R.I. Hartley. Projective reconstruction from line correspondences. In *CVPR*, pages 903 – 907, June 1994. 7.1

[Horn 1987] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629 – 642, April 1987. 7.5.2

[Irani and Anandan 1996] Michal Irani and P. Anandan. Parallax geometry of pairs of points for 3d scene analysis. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume I*, pages 17–30, London, UK, 1996. Springer-Verlag. 7.4

[Jonk and Smeulders 1995] A. Jonk and A.W.M. Smeulders. An axiomatic approach to clustering line-segments. *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, 1:386 – 389, 1995. 5.3, 5.3

[Jurie and Schmid 2004] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *CVPR*, volume II, pages 90 – 96, 2004. 2.1

[Kadir and Brady 2001] T Kadir and M Brady. Scale, saliency and image description. *IJCV*, 45(2):83 – 105, 2001. 2.1

[Kass *et al.* 1987] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *ICCV*, pages 259–268, 1987. 5.1, 5.3

[Ke and Sukthankar 2004] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR (2)*, pages 506 – 513, 2004. 3.1, 3.5, 4.1.5

[Koenderink 1984] J.J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363 – 370, 1984. 2.3, 2.4

[Kullback 1959] S. Kullback. *Information theory and statistics*. John Wiley and Sons., New York, 1959. 5.2.4

[Kusunoki *et al.* 2002] F. Kusunoki, M. Sugimoto, and H. Hashizume. Toward an interactive museum guide with sensing and wireless network technologies. In *WMTE2002, Vaxjo, Sweden*, pages 99 – 102, 2002. 4.1

[Laveau 1996] S. Laveau. *Géométrie d'un système de N caméras : théorie, estimation et applications*. Thèse de : Ecole polytechnique, may 1996. 6.4

[Lindeberg and Bretzner 2003] Tony Lindeberg and Lars Bretzner. Real-time scale selection in hybrid multi-scale representations. In *Scale-Space*, pages 148–163, 2003. 2.4

[Lindeberg 1990] T. Lindeberg. Scale-space for discrete signals. *PAMI*, 12(3):234–254, 1990. 2.3

[Lindeberg 1991] T. Lindeberg. *Discrete Scale-Space Theory and the Scale-Space Primal Sketch, PhD, KTH Stockholm,*. KTH, may 1991. 2.4

[Lindeberg 1998] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2):79 – 116, 1998. I, 2, 2.1, 2.3

[Longuet-Higgins 1981] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133 – 135, 1981. 6.2, 6.3

[Lourakis *et al.* 1998] M. Lourakis, S. Halkidis, and S. Orphanoudakis. Matching disparate views of planar surfaces using projective invariants. In *BMVC*, pages 94 – 104, 1998. 5

[Lowe 1987] D G Lowe. Three-dimensional object recognition from single two-dimensional images. *Artif. Intell.*, 31(3):355–395, 1987. 5.1

[Lowe 1999] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 2.1

[Lowe 2004] D. Lowe. Distinctive image features from scale-invariant keypoints, cascade filtering approach. *IJCV*, 60(2):91 – 110, January 2004. 1.2, I, 2, 2.4, 2.6, 3, 3.1, 4.1.2, 4.1.2, 4.1.5, 6.5, A

[Matas *et al.* 2002] J. Matas, O. Chum, Urban M., and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, pages 384 – 393, 2002. 1.2, I, 3.6, 5.4.2

[McLauchlan *et al.* 2000] P. McLauchlan, X. Shen, A. Manessis, P. Palmer, and A. Hilton. Surface-based structurefrom -motion using feature groupings. In *ACCV*, 2000. 7.1

[Mendonca and Cipolla 1999] P. Mendonca and R. Cipolla. A simple technique for self-calibration. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition.*, pages 500 – 505, 1999. 6.3

[Michalewicz and Fogel 2002] Zbigniew Michalewicz and David B. Fogel. *How to Solve It: Modern Heuristics*. Springer, 2002. 7.3.2

[Mikolajczyk and Schmid 2001] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *ICCV*, volume 1, pages 525 – 531, 2001. 2, 2.1, 2.3

[Mikolajczyk and Schmid 2002] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV*, pages 128 – 142, 2002. I, 3.6, 5.4.2, A

[Mikolajczyk and Schmid 2003] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *CVPR*, volume 2, pages 257 – 263, June 2003. I, 3.1, 4.1.2

[Mikolajczyk and Schmid 2004] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63 – 86, 2004. 1.2, 2.1, 2.6, 6.5

[Mikolajczyk and Schmid 2005] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005. I, 2.1, 3, 3.1, 3.6, 3.3, 3.5, 3.6, 4.1.5, 1

[Mikolajczyk *et al.* 2005] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1/2):43–72, 2005. I, 2.6, 2.6, 2.7

[Mindru *et al.* 2004] F. Mindru, T. Tuytelaars, L. Van Gool, and T. Moons. Moment invariants for recognition under changing viewpoint and illumination. *CVIU*, 94(1-3):3–27, 2004. 3.1

[Montiel *et al.* 2000] J. M. M. Montiel, J. D. Tardós, and L. Montano. Structure and motion from straight line segments. *Pattern Recognition*, 33(8):1295 – 1307, August 2000. 7.1

[Nacken 1993] Peter F. M. Nacken. A metric for line segments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(12):1312 – 1318, 1993. 5.3, 5.3

[Nelder and Mead 1965] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965. 4.2.3

[Niblack *et al.* 1993] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The qbic project:

querying images by content using color, texture and shape. In *SPIE International Symposium on Electronic Imaging: Science and Technology, Conf. 1908, Storage and Retrieval for Image and Video Databases*, 1993. 5.2.4

[Nistér 2004] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004. 6.3

[Palmer *et al.* 1997] P.L. Palmer, J.V. Kittler, and M. Petrou. An optimizing line finder using a hough transform algorithm. *CVIU*, 67(1):1–23, July 1997. 5.1

[Pellejero *et al.* 2003] O.A. Pellejero, C. Sagüés, and J. J. Guerrero. Automatic computation of the fundamental matrix from matched lines. In *10th Conference of the Spanish Association for Artificial Intelligence CAEPIA*, pages 197 – 206, 2003. 5

[Pollefeys *et al.* 2004] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *Int. J. Comput. Vision*, 59(3):207 – 232, 2004. 6, 6.4

[Rother and Carlsson 2002] C. Rother and S. Carlsson. Linear multi view reconstruction and camera recovery using a reference plane. *IJCV*, 49(2-3):117 – 141, September 2002. 5.4.4

[Rother 2000] C. Rother. A new approach for vanishing point detection in architectural environments. In *BMVC*, 2000. 5.4.4, 5.4.4

[Rubner *et al.* 2000] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40(2):99–121, 2000. 5.2.4

[Schaffalitzky and Zisserman 2002] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *ECCV*, volume 1, pages 414 – 431, 2002. 3.1

[Schmid and Zisserman 1997] C. Schmid and A. Zisserman. Automatic line matching across views. In *CVPR*, pages 666 – 671, 1997. 5, 5.2, 5.6

[Schmid *et al.* 2000] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *IJCV*, 37(2):151 – 172, 2000. 2.6, 7.3.1

[Se *et al.* 2004] S Se, H.K. Ng, P. Jasiobedzki, and T.J. Moyung. Vision based modeling and localization for planetary exploration rovers. *Proceedings of International Astronautical Congress*, 2004. I, 3.1

[Sinclair *et al.* 1993] D. Sinclair, A. Blake, S. Smith, and C. Rothwell. Planar region detection and motion recovery. *Image and Vision Computing*, 11(4):229 – 234, 1993. 7.3

[Smith and Chang 1995] J. Smith and S. Chang. Single color extraction and image query. In *International Conference on Image Processing*, pages 528 – 531, 1995. 5.2.3

[Strecha *et al.* 2003] C. Strecha, T. Tuytelaars, and L.J. Van Gool. Dense matching of multiple wide-baseline views. In *ICCV03*, pages 1194–1200, 2003. 6

[Strecha *et al.* 2004] Christoph Strecha, Rik Fransen, and Luc J. Van Gool. Wide-baseline stereo from multiple views: A probabilistic account. In *CVPR (1)*, pages 552 – 559, 2004. 6

[Stricker and Orengo 1995] Markus A. Stricker and Markus Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995. 5.2.4

[Swain and Ballard 1991] Michael J. Swain and Dana H. Ballard. Color indexing. *Int. J. Comput. Vision*, 7(1):11–32, 1991. 5.2.4

[Taylor and Kriegman 1995] C.J. Taylor and D. Kriegman. Structure and motion from line segments in multiple images. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 17(11):1021 – 1033, November 1995. 7.1

[Tell and Carlsson 2000] D. Tell and S. Carlsson. Wide baseline point matching using affine invariants computed from intensity profiles. In *ECCV*, pages 814 – 828, 2000. 5.2.2

[Thrun *et al.* 2000] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 19(11):972–999, 2000. 4.1

[Triggs *et al.* 2000] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 298–372, London, UK, 2000. Springer-Verlag. 6.4

[Tuytelaars and Van Gool 2000] T. Tuytelaars and L. Van Gool. Wide baseline stereo based on local, affinely invariant regions. In *BMVC*, pages 412 – 422, 2000. 1.2, I, 3.6, 5.4.2

[Unser *et al.* 1991] M. Unser, A. Aldroubi, and M. Eden. Fast B-spline transforms for continuous image representation and interpolation. *PAMI*, 13(3):277 – 285, March 1991. 5.2

[Vergauwen and Van Gool to appear] M. Vergauwen and L. Van Gool. Web-based 3d reconstruction service. *Special Issue of Machine Vision and Applications*, to appear. 6.5

[Viola and Jones 2001] P.A. Viola and M.J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR (1)*, pages 511 – 518, 2001. 2

[Vitruvius 1914] Marcus Vitruvius. *The Ten Books on Architecture*. Cambridge (Mass.): Harvard University Press, 1914. translated by Morris Hicky Morgan. 1.1

[Werner and Zisserman 2002] Tomás Werner and Andrew Zisserman. New techniques for automated architectural reconstruction from photographs. In *ECCV (2)*, pages 541 – 555, 2002. 7.1

[Zhang 1995] Zhengyou Zhang. Estimating motion and structure from correspondences of line segments between two perspective images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1129 – 1139, 1995. 7.1

Curriculum Vitae

Personal Data

Name	Herbert Bay
Date of birth	13 June 1974
Place of birth	Winterthur, Switzerland
Citizenship	Swiss

Education

2002 – 2006	PhD student at ETH Zurich, Department of Information Technology and Electrical Engineering, Computer Vision Lab
1997 – 2002	Studies of Micro Enginering at ETH Lausanne and participation at an exchange program with the Ecole Polytechnique de Montreal, Canada
1996 – 1997	Preparation year in Lausanne for the entrance examination to the ETH
1995 – 1996	Fachhochschulreife at the Zeppelin Gewerbeschule Konstanz, Germany
1994 – 1995	Linguistic sojourn in the USA
1991 – 1994	General qualification for studies at a college of higher education at the Berufsmittelschule Frauenfeld
1990 – 1994	Acquisition of the certificate of capability as machine mechanic