Visual 3D Modeling from Images

Marc Pollefeys
University of North Carolina – Chapel Hill, USA

Tutorial Notes

Acknowledgments

iv

Notations

To enhance the readability the notations used throughout the text are summarized here.

For matrices bold face fonts are used (i.e. $\mathbf{A}$). 4-vectors are represented by $\mathtt{A}$ and 3-vectors by $\mathtt{a}$. Scalar values will be represented as a.

Unless stated differently the indices i, j and k are used for views, while l and m are used for indexing points, lines or planes. The notation $\mathbf{A}_{ij}$ indicates the entity $\mathbf{A}$ which relates view i to view j (or going from view i to view j). The indices i,j and k will also be used to indicate the entries of vectors, matrices and tensors. The subscripts P, A, M and E will refer to projective, affine, metric and Euclidean entities respectively

$\mathbf{P}$	camera projection matrix (3×4 matrix)
$\mathtt{M}$	world point (4-vector)
Π	world plane (4-vector)
$\mathtt{m}$	image point (3-vector)
$\mathtt{l}$	image line (3-vector)
$\mathbf{H}_{ij}^{\Pi}$	homography for plane Π from view i to view j (3×3 matrix)
$\mathbf{H}_{\Pi i}$	homography from plane Π to image i (3×3 matrix)
$\mathbf{F}$	fundamental matrix (3×3 rank 2 matrix)
$\mathtt{e}_{ij}$	epipole (projection of projection center of viewpoint i into image j)
$\mathbf{T}$	trifocal tensor ($3 \times 3 \times 3$ tensor)
$\mathbf{K}$	calibration matrix (3×3 upper triangular matrix)
$\mathbf{R}$	rotation matrix
Π_{∞}	plane at infinity (canonical representation: $W = 0$)
Ω	absolute conic
	(canonical representation: $X^2 + Y^2 + Z^2 = 0$ and $W = 0$)
Ω^*	absolute dual quadric (4×4 rank 3 matrix)
ω_{∞}	absolute conic embedded in the plane at infinity (3×3 matrix)
ω_{∞}^*	dual absolute conic embedded in the plane at infinity (3×3 matrix)
ω	image of the absolute conic (3×3 matrices)
ω^*	dual image of the absolute conic (3×3 matrices)
$\sim$	equivalence up to scale ($A \sim B \Leftrightarrow \exists \lambda \neq 0 : A = \lambda B$)
$\|\mathbf{A}\|_F$	indicates the Frobenius norm of $\mathbf{A}$ (i.e. $\sum_{ij} a_{ij}^2$)
$\mathbf{F}(\mathbf{A})$	indicates the matrix $\mathbf{A}$ scaled to have unit Frobenius norm
	(i.e. $\frac{\mathbf{A}}{\|\mathbf{A}\|_F}$)
$\mathbf{A}^\top$	is the transpose of $\mathbf{A}$
$\mathbf{A}^{-1}$	is the inverse of $\mathbf{A}$ (i.e. $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$)
$\mathbf{A}^\dagger$	is the Moore-Penrose pseudo inverse of $\mathbf{A}$

Contents

Chapter 1

Introduction

In recent years computer graphics has made tremendous progress in visualizing 3D models. Many techniques have reached maturity and are being ported to hardware. This explains that in the area of 3D visualization performance is increasing even faster than Moore's law[1]. What required a million dollar computer a few years ago can now be achieved by a game computer costing a few hundred dollars. It is now possible to visualize complex 3D scenes in real time.

This evolution causes an important demand for more complex and realistic models. The problem is that even though the tools that are available for three-dimensional modeling are getting more and more powerful, synthesizing realistic models is difficult and time-consuming, and thus very expensive. Many virtual objects are inspired by real objects and it would therefore be interesting to be able to acquire the models directly from the real object.

Researchers have been investigating methods to acquire 3D information from objects and scenes for many years. In the past the main applications were visual inspection and robot guidance. Nowadays however the emphasis is shifting. There is more and more demand for 3D content for computer graphics, virtual reality and communication. This results in a change in emphasis for the requirements. The visual quality becomes one of the main points of attention. Therefore not only the position of a small number of points have to be measured with high accuracy, but the geometry and appearance of all points of the surface have to be measured.

The acquisition conditions and the technical expertise of the users in these new application domains can often not be matched with the requirements of existing systems. These require intricate calibration procedures every time the system is used. There is an important demand for flexibility in acquisition. Calibration procedures should be absent or restricted to a minimum.

Additionally, the existing systems are often built around specialized hardware (e.g. laser range finders or stereo rigs) resulting in a high cost for these systems. Many new applications however require robust low cost acquisition systems. This stimulates the use of consumer photo- or video cameras. The recent progress in consumer digital imaging facilitates this. Moore's law also tells us that more and more can be done in software.

Due to the convergence of these different factors, many techniques have been developed over the last few years. Many of them do not require more than a camera and a computer to acquire three-dimensional models of real objects.

There are active and passive techniques. The former ones control the lighting of the scene (e.g. projection of structured light) which on the one hand simplifies the problem, but on the other hand restricts the applicability. The latter ones are often more flexible, but computationally more expensive and dependent on the structure of the scene itself.

Some examples of state-of-the-art active techniques are the simple shadow-based approach proposed by Bouguet and Perona [10] or the grid projection approach proposed by Proesmans et al. [126, 133] which is able to extract dynamic textured 3D shapes (this technique is commercially available, see [133]). For the passive techniques many approaches exist. The main differences between the approaches consist of the

---

[1]Moore's law tells us that the density of silicon integrated devices roughly doubles every 18 months.

Figure 1.1: An image of a scene

required level of calibration and the amount of interaction that is required.

For many years photogrammetry [137] has been dealing with the extraction of high accuracy measurements from images. These techniques mostly require very precise calibration and there is almost no automation. The detailed acquisition of models is therefore very time consuming. Besides the tools available for professionals, some simpler tools are commercially available (e.g. PhotoModeler [97]).

Since a few years researchers in computer vision have tried to both reduce the requirements for calibration and augment the automation of the acquisition. The goal is to automatically extract a realistic 3D model by freely moving a camera around an object.

An early approach was proposed by Tomasi and Kanade [149]. They used an affine factorization method to extract 3D from image sequences. An important restriction of this system is the assumption of orthographic projection.

Another type of system starts from an approximate 3D model and camera poses and refines the model based on images (e.g. *Facade* proposed by Debevec et al. [22, 148]). The advantage is that less images are required. On the other hand a preliminary model must be available and the geometry should not be too complex.

In this text it is explained how a 3D surface model can be obtained from a sequence of images taken with off-the-shelf consumer cameras. The user acquires the images by freely moving the camera around the object. Neither the camera motion nor the camera settings have to be known. The obtained 3D model is a scaled version of the original object (i.e. a *metric* reconstruction), and the surface albedo is obtained from the image sequence as well. This approach has been developed over the last few years [99, 102, 103, 105, 107, 111, 109, 71, 112, 113, 66, 101]. The presented system uses full perspective cameras and does not require prior models. It combines state-of-the-art algorithms to solve the different subproblems.

1.1   3D from images

In this section we will try to formulate an answer to the following questions. What do images tell us about a 3D scene? How can we get 3D information from these images? What do we need to know beforehand? A few problems and difficulties will also be presented.

An image like in Figure 1.1 tells us a lot about the observed scene. There is however not enough information to reconstruct the 3D scene (at least not without doing an important number of assumptions on the structure of the scene). This is due to the nature of the image formation process which consists of a projection from a three-dimensional scene onto a two-dimensional image. During this process the depth is lost. Figure 1.2 illustrates this. The three-dimensional point corresponding to a specific image point is constraint to be on the associated line of sight. From a single image it is not possible to determine which point of this line corresponds to the image point. If two (or more) images are available, then -as can be seen
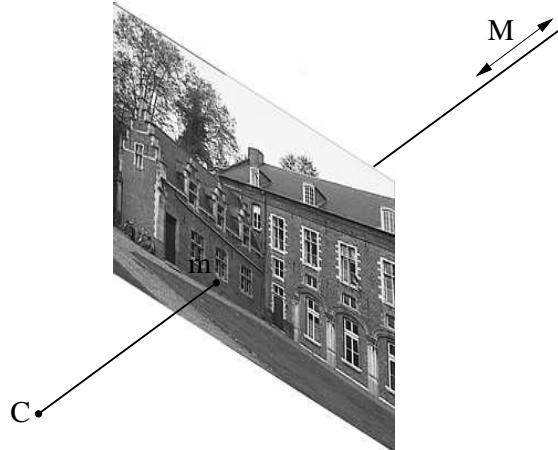
Figure 1.2: Back-projection of a point along the line of sight.

from Figure 1.3- the three-dimensional point can be obtained as the intersection of the two line of sights. This process is called triangulation. Note, however, that a number of things are needed for this:

- Corresponding image points
- Relative pose of the camera for the different views
- Relation between the image points and the corresponding line of sight

The relation between an image point and its line of sight is given by the camera model (e.g. pinhole camera) and the calibration parameters. These parameters are often called the *intrinsic* camera parameters while the position and orientation of the camera are in general called *extrinsic* parameters. In the following chapters we will learn how all these elements can be retrieved from the images. The key for this are the relations between multiple views which tell us that corresponding sets of points must contain some structure and that this structure is related to the poses and the calibration of the camera.

Note that different viewpoints are not the only depth cues that are available in images. In Figure 1.4 some other depth cues are illustrated. Although approaches have been presented that can exploit most of these, in this text we will concentrate on the use of multiple views.

In Figure 1.5 a few problems for 3D modeling from images are illustrated. Most of these problems will limit the application of the presented method. However, some of the problems can be tackled by the presented approach. Another type of problems is caused when the imaging process does not satisfy the camera model that is used. In Figure 1.6 two examples are given. In the left image quite some radial distortion is present. This means that the assumption of a pinhole camera is not satisfied. It is however possible to extend the model to take the distortion into account. The right image however is much harder to use since an important part of the scene is not in focus. There is also some blooming in that image (i.e. overflow of CCD-pixel to the whole column). Most of these problems can however be avoided under normal imaging circumstance.

1.2 Overview

The presented system gradually retrieves more information about the scene and the camera setup. Images contain a huge amount of information (e.g. 768×512 color pixels). However, a lot of it is redundant (which explains the success of image compression algorithms). The structure recovery approaches require correspondences between the different images (i.e. image points originating from the same scene point). Due to the combinatorial nature of this problem it is almost impossible to work on the raw data. The first step therefore consists of extracting features. The features of different images are then compared using similarity measures and lists of potential matches are established. Based on these the relation between the views are computed. Since wrong correspondences can be present, robust algorithms are used. Once
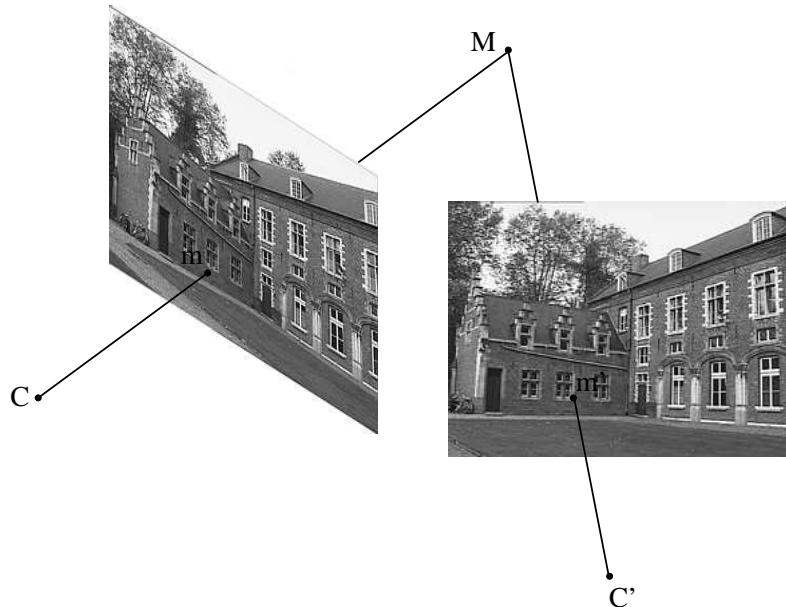
Figure 1.3: Reconstruction of three-dimensional point through triangulation.
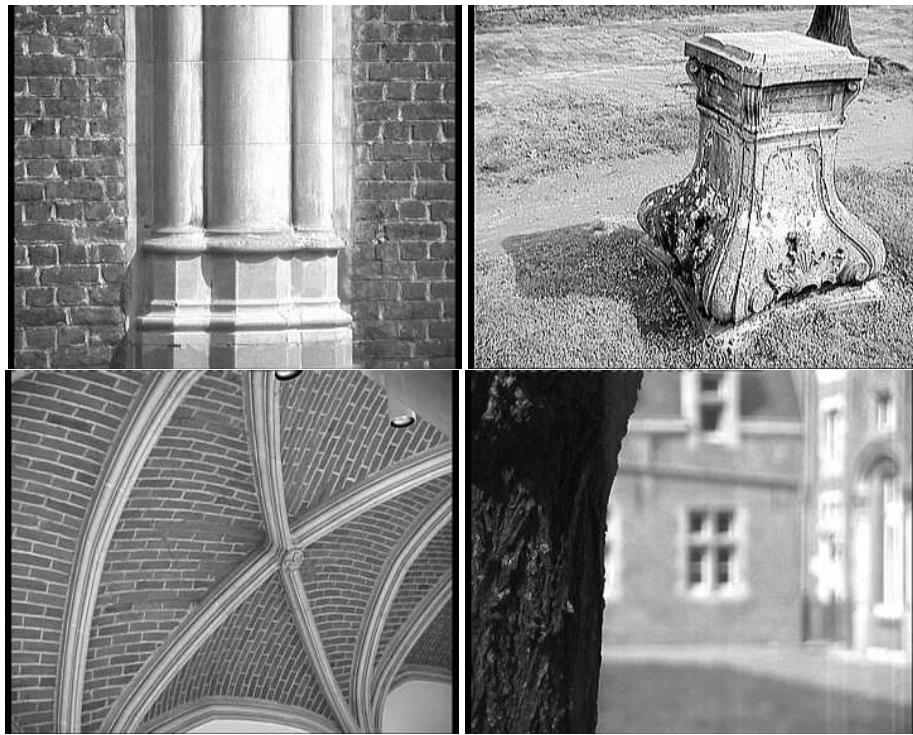
Figure 1.4: Shading (top-left), shadows/symmetry/silhouette (top-right), texture (bottom-left) and focus (bottom-right) also give some hints about depth or local geometry.

Figure 1.5: Some difficult scenes: moving objects (top-left), complex scene with many discontinuities (top-right), reflections (bottom-left) and another hard scene (bottom-right).

Figure 1.6: Some problems with image acquisition: radial distortion (left), un-focussed and blooming (right).

consecutive views have been related to each other, the structure of the features and the motion of the camera is computed. An initial reconstruction is then made for the first two images of the sequence. For the subsequent images the camera pose is estimated in the frame defined by the first two cameras. For every additional image that is processed at this stage, the features corresponding to points in previous images are reconstructed, refined or corrected. Therefore it is not necessary that the initial points stay visible throughout the entire sequence. The result of this step is a reconstruction of typically a few hundred feature points. When uncalibrated cameras are used the structure of the scene and the motion of the camera is only determined up to an arbitrary projective transformation. The next step consists of restricting this ambiguity to metric (i.e. Euclidean up to an arbitrary scale factor) through self-calibration. In a projective reconstruction not only the scene, but also the camera is distorted. Since the algorithm deals with unknown scenes, it has no way of identifying this distortion in the reconstruction. Although the camera is also assumed to be unknown, some constraints on the intrinsic camera parameters (e.g. rectangular or square pixels, constant aspect ratio, principal point in the middle of the image, ...) can often still be assumed. A distortion on the camera mostly results in the violation of one or more of these constraints. A metric reconstruction/calibration is obtained by transforming the projective reconstruction until all the constraints on the cameras intrinsic parameters are satisfied. At this point enough information is available to go back to the images and look for correspondences for all the other image points. This search is facilitated since the line of sight corresponding to an image point can be projected to other images, restricting the search range to one dimension. By pre-warping the image -this process is called rectification- standard stereo matching algorithms can be used. This step allows to find correspondences for most of the pixels in the images. From these correspondences the distance from the points to the camera center can be obtained through triangulation. These results are refined and completed by combining the correspondences from multiple images. Finally all results are integrated in a textured 3D surface reconstruction of the scene under consideration. The model is obtained by approximating the depth map with a triangular wire frame. The texture is obtained from the images and mapped onto the surface. An overview of the systems is given in Figure 1.7.

Throughout the rest of the text the different steps of the method will be explained in more detail. An image sequence of the Arenberg castle in Leuven will be used for illustration. Some of the images of this sequence can be seen in Figure 1.8. The full sequence consists of 24 images recorded with a video camera.
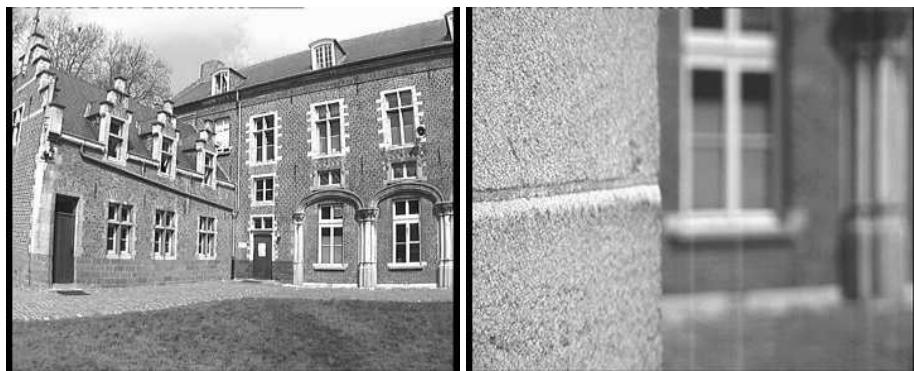
Structure of the notes   Chapter 2 and 3 give the geometric foundation to understand the principles behind the presented approaches. The former introduces projective geometry and the stratification of geometric structure. The latter describes the perspective camera model and derives the relation between multiple views. These are at the basis of the possibility to achieve structure and motion recovery. This allows the interested reader to understand what is behind the techniques presented in the other chapters, but can also be skipped.

Chapter 4 deals with the extraction and matching of features and the recovery of multiple view relations. A robust technique is presented to automatically relate two views to each other.

Chapter 5 describes how starting from the relation between consecutive images the structure and motion of the whole sequence can be built up. Chapter 6 briefly describes some self-calibration approaches and proposes a practical method to reduce the ambiguity on the structure and motion to metric.

Chapter 7 is concerned with computing correspondences for all the image points. First an algorithm for stereo matching is presented. Then rectification is explained. A general method is proposed which can transform every image pair to standard stereo configuration. Finally, a multi-view approach is presented which allows to obtain denser depth maps and better accuracy.

In Chapter 8 it is explained how the results obtained in the previous chapters can be combined to obtain realistic models of the acquired scenes. At this point a lot of information is available and different types of models can be computed. The chapter describes how to obtain surface models and other visual models. The possibility to augment a video sequence is also presented.
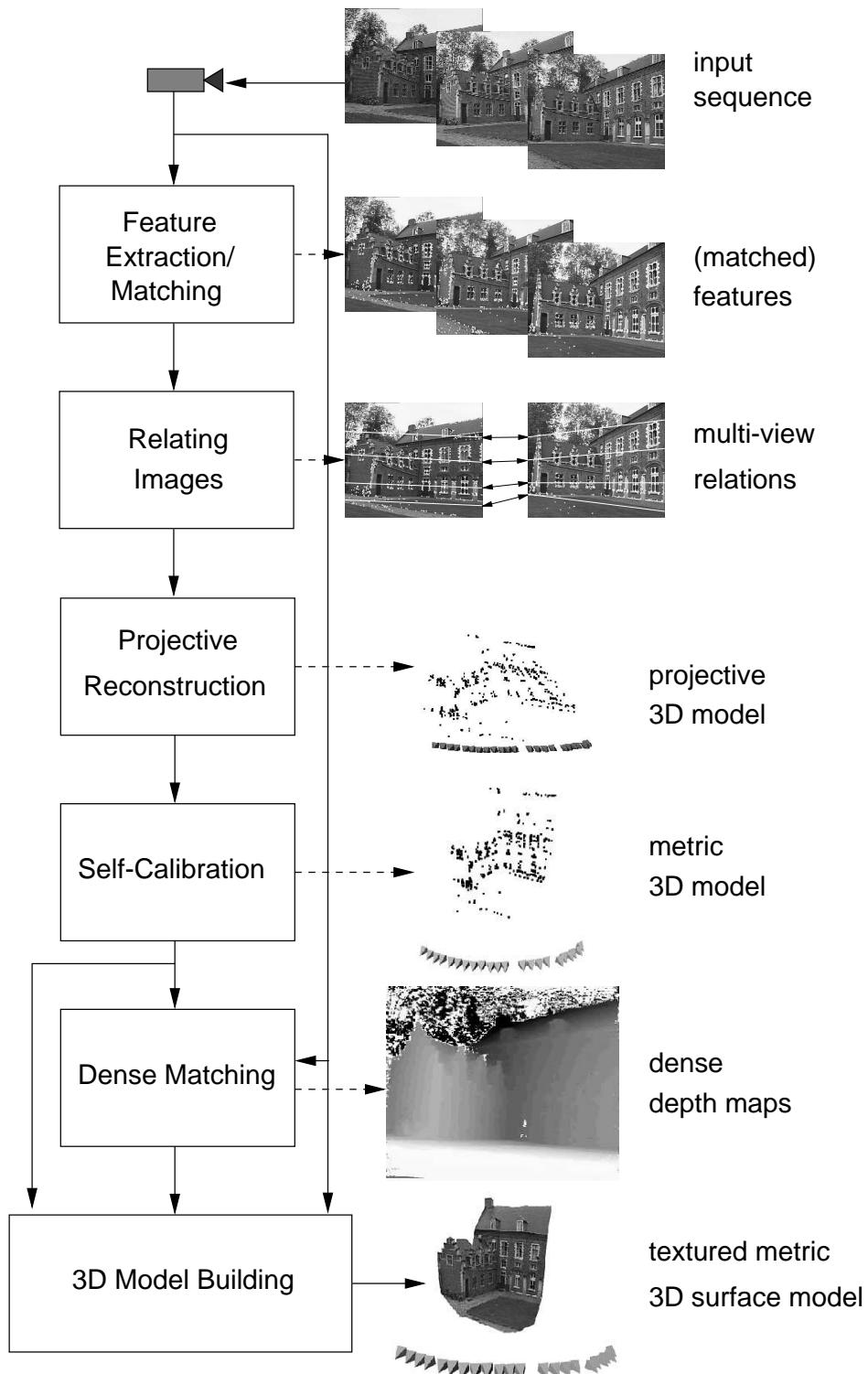
Figure 1.7: Overview of the presented approach for 3D modeling from images

Figure 1.8: *Castle sequence*: this sequence is used throughout the text to illustrate the different steps of the reconstruction system.

Chapter 2

Projective geometry

... ὤστε καλλιον ἀποδεξεσθαι, ἰσμεν πον ὀτι τω ὁλω και παντι διοισει ἠμμενος τε γεωμετραις
και μη

"... experience proves that anyone who has studied geometry is infinitely quicker to grasp difficult subjects
than one who has not."
Plato - The Republic, Book 7, 375 B.C.

The work presented in this text draws a lot on concepts of projective geometry. This chapter and the
next one introduce most of the geometric concepts used in the rest of the text. This chapter concentrates
on projective geometry and introduces concepts as points, lines, planes, conics and quadrics in two or
three dimensions. A lot of attention goes to the stratification of geometry in projective, affine, metric
and Euclidean layers. Projective geometry is used for its simplicity in formalism, additional structure and
properties can then be introduced were needed through this hierarchy of geometric strata. This section
was inspired by the introductions on projective geometry found in Faugeras' book [29], in the book by
Mundy and Zisserman (in [91]) and by the book on projective geometry by Semple and Kneebone [132].
A detailed account on the subject can be found in the recent book by Hartley and Zisserman [55].

2.1 Projective geometry

A point in projective n-space, $\mathcal{P}^n$, is given by a $(n+1)$-vector of coordinates $\mathtt{x} = [x_1 \ldots x_{n+1}]^\top$. At least
one of these coordinates should differ from zero. These coordinates are called *homogeneous* coordinates.
In the text the coordinate vector and the point itself will be indicated with the same symbol. Two points
represented by $(n+1)$-vectors $\mathtt{x}$ and $\mathtt{y}$ are equal if and only if there exists a nonzero scalar λ such that
$x_i = \lambda y_i$, for every i $(1 \leq i \leq n+1)$. This will be indicated by $\mathtt{x} \sim \mathtt{y}$.

Often the points with coordinate $x_{n+1} = 0$ are said to be *at infinity*. This is related to the affine space
$\mathcal{A}^n$. This concept is explained more in detail in section 2.2.

A *collineation* is a mapping between projective spaces, which preserves collinearity (i.e. collinear
points are mapped to collinear points). A collineation from $\mathcal{P}^m$ to $\mathcal{P}^n$ is mathematically represented
by a $(m+1) \times (n+1)$-matrix $\mathbf{H}$. Points are transformed linearly: $\mathtt{x} \mapsto \mathtt{x}' \sim \mathbf{H}\mathtt{x}$. Observe that matrices
$\mathbf{H}$ and $\lambda \mathbf{H}$ with λ a nonzero scalar represent the same collineation.

A *projective basis* is the extension of a coordinate system to projective geometry. A projective basis is
a set of $n+2$ points such that no $n+1$ of them are linearly dependent. The set $\mathtt{e}_l = [0 \ldots 1 \ldots 0]^\top$ for
every l $(1 \leq l \leq n+1)$, where 1 is in the lth position and $\mathtt{e}_{n+2} = [1\,1 \ldots 1]^\top$ is the standard projective
basis. A projective point of $\mathcal{P}^n$ can be described as a linear combination of any $n+1$ points of the standard
basis. For example:

$$\mathtt{m} = \sum_{l=1}^{n+1} \lambda_l \mathtt{e}_l$$

It can be shown [31] that any projective basis can be transformed via a uniquely determined collineation

into the standard projective basis. Similarly, if two set of points $m_1, \ldots, m_{n+2}$ and $m'_1, \ldots, m'_{n+2}$ both form a projective basis, then there exists a uniquely determined collineation $\mathbf{T}$ such that $m'_l \sim \mathbf{T}m_l$ for every l $(1 \leq l \leq n+2)$. This collineation $\mathbf{T}$ describes the change of projective basis. In particular, $\mathbf{T}$ is invertible.

2.1.1 The projective plane

The projective plane is the projective space $\mathcal{P}^2$. A point of $\mathcal{P}^2$ is represented by a 3-vector $m = [x\, y\, w]^\top$. A line l is also represented by a 3-vector. A point m is located on a line l if and only if

$$l^\top m = 0 \ . \tag{2.1}$$

This equation can however also be interpreted as expressing that the line l passes through the point m. This symmetry in the equation shows that there is no formal difference between points and lines in the projective plane. This is known as the principle of *duality*. A line l passing through two points m_1 and m_2 is given by their vector product $m_1 \times m_2$. This can also be written as

$$l \sim [m_1]_\times m_2 \text{ with } [m_1]_\times = \begin{bmatrix} 0 & w_1 & -y_1 \\ -w_1 & 0 & x_1 \\ y_1 & -x_1 & 0 \end{bmatrix} \ . \tag{2.2}$$

The dual formulation gives the intersection of two lines. All the lines passing through a specific point form a *pencil of lines*. If two lines l_1 and l_2 are distinct elements of the pencil, all the other lines can be obtained through the following equation:

$$l \sim \lambda_1 l_1 + \lambda_2 l_2 \tag{2.3}$$

for some scalars λ_1 and λ_2. Note that only the ratio $\frac{\lambda_1}{\lambda_2}$ is important.

2.1.2 Projective 3-space

Projective 3D space is the projective space $\mathcal{P}^3$. A point of $\mathcal{P}^3$ is represented by a 4-vector $M = [X\, Y\, Z\, W]^\top$. In $\mathcal{P}^3$ the dual entity of a point is a plane, which is also represented by a 4-vector. A point M is located on a plane Π if and only if

$$\Pi^\top M = 0 \ . \tag{2.4}$$

A line can be given by the linear combination of two points $\lambda_1 M_1 + \lambda_2 M_2$ or by the intersection of two planes $\Pi_1 \cap \Pi_2$.

2.1.3 Transformations

Transformations in the images are represented by *homographies* of $\mathcal{P}^2 \to \mathcal{P}^2$. A homography of $\mathcal{P}^2 \to \mathcal{P}^2$ is represented by a 3×3-matrix $\mathbf{H}$. Again $\mathbf{H}$ and $\lambda\mathbf{H}$ represent the same homography for all nonzero scalars λ. A point is transformed as follows:

$$m \mapsto m' \sim \mathbf{H}m \ . \tag{2.5}$$

The corresponding transformation of a line can be obtained by transforming the points which are on the line and then finding the line defined by these points:

$$l'^\top m' = l^\top \mathbf{H}^{-1} \mathbf{H}m = l^\top m = 0 \ . \tag{2.6}$$

From the previous equation the transformation equation for a line is easily obtained (with $\mathbf{H}^{-\top} = (\mathbf{H}^{-1})^\top = (\mathbf{H}^\top)^{-1}$):

$$l \mapsto l' \sim \mathbf{H}^{-\top} l \tag{2.7}$$

Similar reasoning in $\mathcal{P}^3$ gives the following equations for transformations of points and planes in 3D space:

$$M \quad \mapsto \quad M' \sim \mathbf{T}M, \tag{2.8}$$
$$\Pi \quad \mapsto \quad \Pi' \sim \mathbf{T}^{-\top}\Pi \tag{2.9}$$

where $\mathbf{T}$ is a 4×4-matrix.

2.1.4 Conics and quadrics

Conic A *conic* in $\mathcal{P}^2$ is the locus of all points $\mathtt{m}$ satisfying a homogeneous quadratic equation:

$$S(\mathtt{m}) = \mathtt{m}^\top \mathbf{C} \mathtt{m} = 0 \,, \tag{2.10}$$

where $\mathbf{C}$ is a 3×3 symmetric matrix only defined up to scale. A conic thus depends on five independent parameters.

Dual conic Similarly, the dual concept exists for lines. A *conic envelope* or *dual conic* is the locus of all lines $\mathtt{l}$ satisfying a homogeneous quadratic equation:

$$\mathtt{l}^\top \mathbf{C}^* \mathtt{l} = 0 \,, \tag{2.11}$$

where $\mathbf{C}^*$ is a 3×3 symmetric matrix only defined up to scale. A dual conic thus also depends on five independent parameters.

Line-conic intersection Let $\mathtt{m}$ and $\mathtt{m}'$ be two points defining a line. A point on this line can then be represented by $\mathtt{m} + \lambda \mathtt{m}'$. This point lies on a conic S if and only if

$$S(\mathtt{m} + \lambda \mathtt{m}') = 0 \,,$$

which can also be written as

$$S(\mathtt{m}) + 2\lambda S(\mathtt{m}, \mathtt{m}') + \lambda^2 S(\mathtt{m}') = 0 \,, \tag{2.12}$$

where

$$S(\mathtt{m}, \mathtt{m}') = \mathtt{m}^\top \mathbf{C} \mathtt{m}' = S(\mathtt{m}', \mathtt{m})$$

This means that a line has in general two intersection points with a conic. These intersection points can be real or complex and can be obtained by solving equation (2.12).

Tangent to a conic The two intersection points of a line with a conic coincide if the discriminant of equation (2.12) is zero. This can be written as

$$S(\mathtt{m}, \mathtt{m}') - S(\mathtt{m}) S(\mathtt{m}') = 0 \ .$$

If the point $\mathtt{m}$ is considered fixed, this forms a quadratic equation in the coordinates of $\mathtt{m}'$ which represents the two tangents from $\mathtt{m}$ to the conic. If $\mathtt{m}$ belongs to the conic, $S(\mathtt{m}) = 0$ and the equation of the tangents becomes

$$S(\mathtt{m}, \mathtt{m}') = \mathtt{m}^\top \mathbf{C} \mathtt{m}' = 0 \ ,$$

which is linear in the coefficients of $\mathtt{m}'$. This means that there is only one tangent to the conic at a point of the conic. This tangent $\mathtt{l}$ is thus represented by :

$$\mathtt{l} \sim \mathbf{C}^\top \mathtt{m} = \mathbf{C}\mathtt{m} \tag{2.13}$$

Relation between conic and dual conic When $\mathtt{m}$ varies along the conic, it satisfies $\mathtt{m}^\top \mathbf{C} \mathtt{m}$ and thus the tangent line $\mathtt{l}$ to the conic at $\mathtt{m}$ satisfies $\mathtt{l}^\top \mathbf{C}^{-1} \mathtt{l} = 0$. This shows that the tangents to a conic $\mathbf{C}$ are belonging to a dual conic $\mathbf{C}^* \sim \mathbf{C}^{-1}$ (assuming $\mathbf{C}$ is of full rank).

Transformation of a conic/dual conic The transformation equations for conics and dual conics under a homography $\mathbf{H}$ can be obtained in a similar way to Section 2.1.3. Using equations (2.5) and (2.7) the following is obtained:

$$\begin{aligned}
\mathtt{m}'^\top \mathbf{C}' \mathtt{m}' &\sim \mathtt{m}^\top \mathbf{H}^\top \mathbf{H}^{-\top} \mathbf{C} \mathbf{H}^{-1} \mathbf{H} \mathtt{m} = 0 \,, \\
\mathtt{l}'^\top \mathbf{C}^{*'} \mathtt{l}' &\sim \mathtt{l}^\top \mathbf{H}^{-1} \mathbf{H} \mathbf{C}^* \mathbf{H}^\top \mathbf{H}^{-\top} \mathtt{l} = 0 \,,
\end{aligned}$$

and thus

$$\mathbf{C} \mapsto \mathbf{C}' \sim \mathbf{H}^{-\top} \mathbf{C} \mathbf{H}^{-1} \tag{2.14}$$

$$\mathbf{C}^* \mapsto \mathbf{C}^{*'} \sim \mathbf{H} \mathbf{C}^* \mathbf{H}^\top \tag{2.15}$$

Observe that (2.14) and (2.15) also imply that $(\mathbf{C}')^* = (\mathbf{C}^*)'$.

Quadric   In projective 3-space $\mathcal{P}^3$ similar concepts exist. These are quadrics. A *quadric* is the locus of all points M satisfying a homogeneous quadratic equation:

$$\mathtt{M}^\top \mathbf{Q} \mathtt{M} = 0 \,, \tag{2.16}$$

where $\mathbf{Q}$ is a 4×4 symmetric matrix only defined up to scale. A quadric thus depends on nine independent parameters.

Dual quadric   Similarly, the dual concept exists for planes. A *dual quadric* is the locus of all planes Π satisfying a homogeneous quadratic equation:

$$\Pi^\top \mathbf{Q}^* \Pi = 0 \tag{2.17}$$

where $\mathbf{Q}^*$ is a 3×3 symmetric matrix only defined up to scale and thus also depends on nine independent parameters.

Tangent to a quadric   Similar to equation (2.13), the tangent plane Π to a quadric $\mathbf{Q}$ through a point M of the quadric is obtained as

$$\Pi = \mathbf{Q} \mathtt{M} \ . \tag{2.18}$$

Relation between quadric and dual quadric   When M varies along the quadric, it satisfies $\mathtt{M}^\top \mathbf{Q} \mathtt{M}$ and thus the tangent plane Π to $\mathbf{Q}$ at M satisfies $\Pi^\top \mathbf{Q}^{-1} \Pi = 0$. This shows that the tangent planes to a quadric $\mathbf{Q}$ are belonging to a dual quadric $\mathbf{Q}^* \sim \mathbf{Q}^{-1}$ (assuming $\mathbf{Q}$ is of full rank).

Transformation of a quadric/dual quadric   The transformation equations for quadrics and dual quadrics under a homography $\mathbf{T}$ can be obtained in a similar way to Section 2.1.3. Using equations (2.8) and (2.9) the following is obtained

$$\mathtt{M}'^\top \mathbf{Q}' \mathtt{M}' \quad \sim \quad \mathtt{M}^\top \mathbf{T}^\top \mathbf{T}^{-\top} \mathbf{Q} \mathbf{T}^{-1} \mathbf{T} \mathtt{M} = 0$$
$$\Pi'^\top \mathbf{Q}^{*'} \Pi' \quad \sim \quad \Pi^\top \mathbf{T}^{-1} \mathbf{T} \mathbf{Q}^* \mathbf{T}^\top \mathbf{T}^{-\top} \Pi = 0$$

and thus

$$\mathbf{Q} \quad \mapsto \quad \mathbf{Q}' \sim \mathbf{T}^{-\top} \mathbf{Q} \mathbf{T}^{-1} \tag{2.19}$$
$$\mathbf{Q}^* \quad \mapsto \quad \mathbf{Q}^{*'} \sim \mathbf{T} \mathbf{Q}^* \mathbf{T}^\top \tag{2.20}$$

Observe again that $(\mathbf{Q}')^* = (\mathbf{Q}^*)'$.

2.2   The stratification of 3D geometry

Usually the world is perceived as a Euclidean 3D space. In some cases (e.g. starting from images) it is not possible or desirable to use the full Euclidean structure of 3D space. It can be interesting to only deal with the less structured and thus simpler projective geometry. Intermediate layers are formed by the affine and metric geometry. These structures can be thought of as different geometric strata which can be overlaid on the world. The simplest being projective, then affine, next metric and finally Euclidean structure.

This concept of stratification is closely related to the groups of transformations acting on geometric entities and leaving invariant some properties of configurations of these elements. Attached to the projective stratum is the group of projective transformations, attached to the affine stratum is the group of affine transformations, attached to the metric stratum is the group of similarities and attached to the Euclidean stratum is the group of Euclidean transformations. It is important to notice that these groups are subgroups of each other, e.g. the metric group is a subgroup of the affine group and both are subgroups of the projective group.

An important aspect related to these groups are their invariants. An *invariant* is a property of a con-figuration of geometric entities that is not altered by any transformation belonging to a specific group.

Invariants therefore correspond to the measurements that one can do considering a specific stratum of geometry. These invariants are often related to geometric entities which stay unchanged – at least as a whole – under the transformations of a specific group. These entities play an important role in part of this text. Recovering them allows to upgrade the structure of the geometry to a higher level of the stratification.

In the following paragraphs the different strata of geometry are discussed. The associated groups of transformations, their invariants and the corresponding invariant structures are presented. This idea of stratification can be found back in [132] and [30].

2.2.1 Projective stratum

The first stratum is the projective one. It is the less structured one and has therefore the least number of invariants and the largest group of transformations associated with it. The group of projective transformations or collineations is the most general group of linear transformations.

As seen in the previous chapter a projective transformation of 3D space can be represented by a 4×4 invertible matrix

$$\mathbf{T}_P \sim \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} \tag{2.21}$$

This transformation matrix is only defined up to a nonzero scale factor and has therefore 15 degrees of freedom.

Relations of incidence, collinearity and tangency are projectively invariant. The cross-ratio is an invariant property under projective transformations as well. It is defined as follows: Assume that the four points $\mathtt{M}_1, \mathtt{M}_2, \mathtt{M}_3$ and $\mathtt{M}_4$ are collinear. Then they can be expressed as $\mathtt{M}_i = \mathtt{M} + \lambda_i \mathtt{M}'$ (assume none is coincident with $\mathtt{M}'$). The cross-ratio is defined as

$$\{\mathtt{M}_1, \mathtt{M}_2; \mathtt{M}_3, \mathtt{M}_4\} = \frac{\lambda_1 - \lambda_3}{\lambda_1 - \lambda_4} : \frac{\lambda_2 - \lambda_3}{\lambda_2 - \lambda_4} \quad . \tag{2.22}$$

The cross-ratio is not depending on the choice of the reference points $\mathtt{M}$ and $\mathtt{M}'$ and is invariant under the group of projective transformations of $\mathcal{P}^3$. A similar cross-ratio invariant can be derived for four lines intersecting in a point or four planes intersecting in a common line.

The cross-ratio can in fact be seen as the coordinate of a fourth point in the basis of the first three, since three points form a basis for the projective line $\mathcal{P}^1$. Similarly, two invariants could be obtained for five coplanar points; and, three invariants for six points, all in general position.

2.2.2 Affine stratum

The next stratum is the affine one. In the hierarchy of groups it is located in between the projective and the metric group. This stratum contains more structure than the projective one, but less than the metric or the Euclidean strata. Affine geometry differs from projective geometry by identifying a special plane, called the *plane at infinity*.

This plane is usually defined by $W = 0$ and thus $\Pi_\infty = [0\,0\,0\,1]^\top$. The projective space can be seen as containing the affine space under the mapping $\mathcal{A}^3 \to \mathcal{P}^3 : [X\,Y\,Z]^\top \mapsto [X\,Y\,Z\,1]^\top$. This is a one-to-one mapping. The plane $W = 0$ in $\mathcal{P}^3$ can be seen as containing the limit points for $\|\mathtt{M}\| \to \infty$, since these points are $[\frac{X}{\|\mathtt{M}\|} \ \frac{Y}{\|\mathtt{M}\|} \ \frac{Z}{\|\mathtt{M}\|} \ \frac{1}{\|\mathtt{M}\|}]^\top \sim [X_\infty\,Y_\infty\,Z_\infty\,0]$. This plane is therefore called the plane at infinity Π_∞. Strictly speaking, this plane is not part of the affine space, the points contained in it can't be expressed through the usual non-homogeneous 3-vector coordinate notation used for affine, metric and Euclidean 3D space.

An *affine transformation* is usually presented as follows:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix} \text{ with } \det(a_{ij}) \neq 0$$

Using homogeneous coordinates, this can be rewritten as follows $M' \sim \mathbf{T}_A M$ with

$$\mathbf{T}_A \sim \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} . \tag{2.23}$$

An affine transformation counts 12 independent degrees of freedom. It can easily be verified that this transformation leaves the plane at infinity Π_∞ unchanged (i.e. $\Pi_\infty \sim \mathbf{T}_A^{-\top} \Pi_\infty$ or $\mathbf{T}_A^\top \Pi_\infty \sim \Pi_\infty$). Note, however, that the position of points in the plane at infinity can change under an affine transformation, but that all these points stay within the plane Π_∞.

All projective properties are a fortiori affine properties. For the (more restrictive) affine group parallelism is added as a new invariant property. Lines or planes having their intersection in the plane at infinity are called *parallel*. A new invariant property for this group is the *ratio of lengths along a certain direction*. Note that this is equivalent to a cross-ratio with one of the points at infinity.

From projective to affine   Up to now it was assumed that these different strata could simply be overlaid onto each other, assuming that the plane at infinity is at its canonical position (i.e. $\Pi_\infty = [0\,0\,0\,1]^\top$). This is easy to achieve when starting from a Euclidean representation. Starting from a projective representation, however, the structure is only determined up to an arbitrary projective transformation. As was seen, these transformations do – in general – not leave the plane at infinity unchanged.

Therefore, in a specific projective representation, the plane at infinity can be anywhere. In this case upgrading the geometric structure from projective to affine implies that one first has to find the position of the plane at infinity in the particular projective representation under consideration.

This can be done when some affine properties of the scene are known. Since parallel lines or planes are intersecting in the plane at infinity, this gives constraints on the position of this plane. In Figure 2.1 a projective representation of a cube is given. Knowing this is a cube, three vanishing points can be identified. The plane at infinity is the plane containing these 3 vanishing points.

Ratios of lengths along a line define the point at infinity of that line. In this case the points M_0, M_1, M_2 and the cross-ratio $\{M_1, M_2; M_0, M_\infty\}$ are known, therefore the point M_∞ can be computed.

Once the plane at infinity Π_∞ is known, one can upgrade the projective representation to an affine one by applying a transformation which brings the plane at infinity to its canonical position. Based on (2.9) this equation should therefore satisfy

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \sim \mathbf{T}^{-\top} \Pi_\infty \text{ or } \mathbf{T}^\top \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \sim \Pi_\infty \tag{2.24}$$

This determines the fourth row of $\mathbf{T}$. Since, at this level, the other elements are not constrained, the obvious choice for the transformation is the following

$$\mathbf{T}_{PA} \sim \begin{bmatrix} \mathbf{I}_{3\times3} & 0_3 \\ \pi_\infty^\top & 1 \end{bmatrix} \tag{2.25}$$

with π_∞ the first 3 elements of Π_∞ when the last element is scaled to 1. It is important to note, however, that every transformation of the form

$$\begin{bmatrix} \mathbf{A} & 0_3 \\ \pi_\infty^\top & 1 \end{bmatrix} \text{ with } \det \mathbf{A} \neq 0 \tag{2.26}$$

maps Π_∞ to $[0\,0\,0\,1]^\top$.

2.2.3   Metric stratum

The metric stratum corresponds to the group of similarities. These transformations correspond to Euclidean transformations (i.e. orthonormal transformation + translation) complemented with a scaling. When no
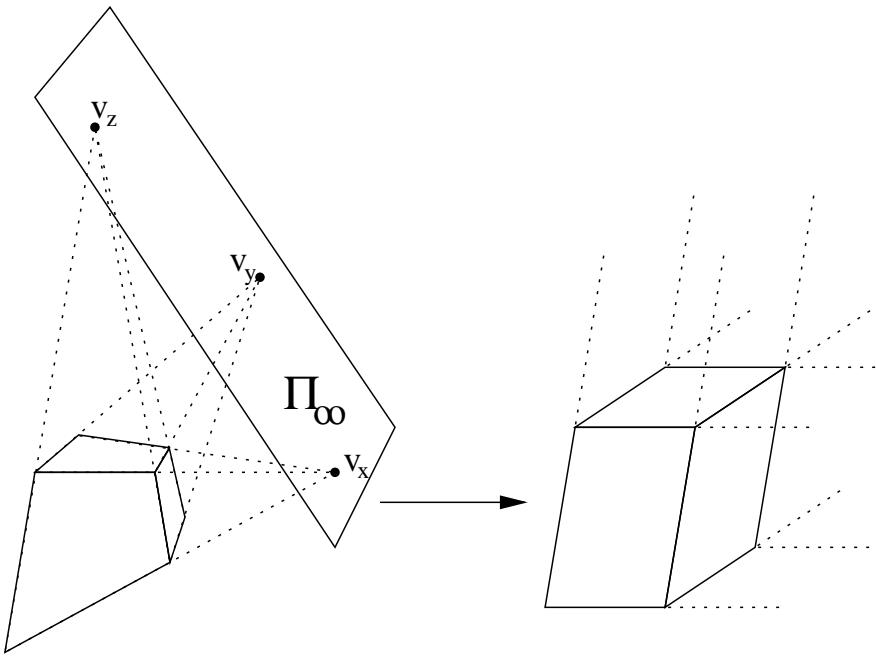
Figure 2.1: *Projective (left) and affine (right) structures which are equivalent to a cube under their respective ambiguities. The vanishing points obtained from lines which are parallel in the affine stratum constrain the position of the plane at infinity in the projective representation. This can be used to upgrade the geometric structure from projective to affine.*

absolute yardstick is available, this is the highest level of geometric structure that can be retrieved from images. Inversely, this property is crucial for special effects since it enables the possibility to use scale models in movies.

A metric transformation can be represented as follows:

$$
\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \sigma \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_{14} \\ t_{24} \\ t_{34} \end{bmatrix} \tag{2.27}
$$

with r_{ij} the coefficients of an orthonormal matrix. The coefficients r_{ij} are related by 6 independent constraints $\sum_{k=1}^{3} r_{ik} r_{jk} = \delta_{ij}, (1 \leq i \leq j; 1 \leq j \leq 3)$ with δ_{ij} the Kronecker delta[1]. This corresponds to the matrix relation that $\mathbf{R}^{\top}\mathbf{R} = \mathbf{R}\mathbf{R}^{\top} = \mathbf{I}$ and thus $\mathbf{R}^{-1} = \mathbf{R}^{\top}$. Recall that $\mathbf{R}$ is a rotation matrix if and only if $\mathbf{R}\mathbf{R}^{\top} = \mathbf{I}$ and $\det \mathbf{R} = 1$. In particular, an orthonormal matrix only has 3 degrees of freedom. Using homogeneous coordinates, (2.27) can be rewritten as $\mathtt{M}' \sim \mathbf{T}_M \mathtt{M}$, with

$$
\mathbf{T}_M \sim \begin{bmatrix} \sigma r_{11} & \sigma r_{12} & \sigma r_{13} & t_X \\ \sigma r_{21} & \sigma r_{22} & \sigma r_{23} & t_Y \\ \sigma r_{31} & \sigma r_{32} & \sigma r_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \sim \begin{bmatrix} r_{11} & r_{12} & r_{13} & \sigma^{-1}t_X \\ r_{21} & r_{22} & r_{23} & \sigma^{-1}t_Y \\ r_{31} & r_{32} & r_{33} & \sigma^{-1}t_Z \\ 0 & 0 & 0 & \sigma^{-1} \end{bmatrix} \tag{2.28}
$$

A metric transformation therefore counts 7 independent degrees of freedom, 3 for orientation, 3 for translation and 1 for scale.

In this case there are two important new invariant properties: *relative lengths* and *angles*. Similar to the affine case, these new invariant properties are related to an invariant geometric entity. Besides leaving the plane at infinity unchanged similarity transformations also transform a specific conic into itself, i.e. the

---

[1]The Kronecker delta is defined as follows $\begin{cases} \delta_{ij} = 1 \text{ for } i = j \\ \delta_{ij} = 0 \text{ for } i \neq j \end{cases}$.
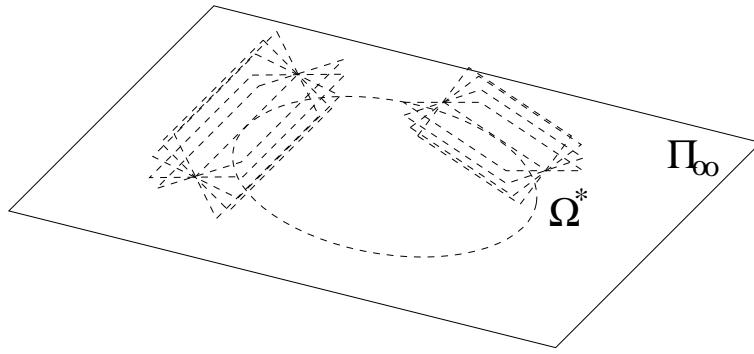
Figure 2.2: *The absolute conic Ω and the absolute dual quadric Ω^* in 3D space.*
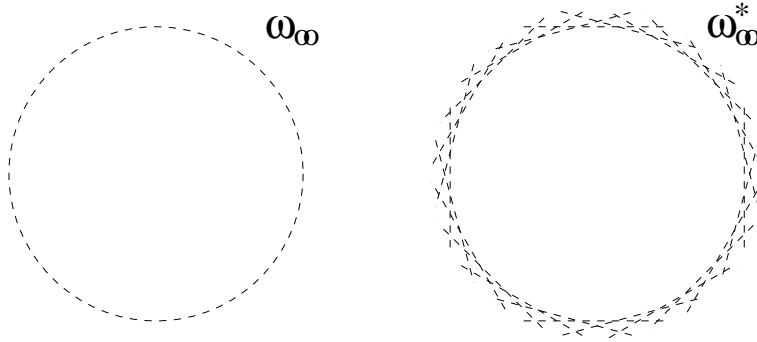
Figure 2.3: *The absolute conic ω_∞ and dual absolute conic ω_∞^* represented in the purely imaginary part of the plane at infinity Π_∞*

absolute conic. This geometric concept is more abstract than the plane at infinity. It could be seen as an imaginary circle located in the plane at infinity. In this text the absolute conic is denoted by Ω. It is often more practical to represent this entity in 3D space by its dual entity Ω^*. When only the plane at infinity is under consideration, ω_∞ and ω_∞^* are used to represent the absolute conic and the dual absolute conic (these are 2D entities). Figure 2.2 and Figure 2.3 illustrate these concepts.   The canonical form for the absolute conic Ω is:

$$\Omega : X^2 + Y^2 + Z^2 = 0 \text{ and } W = 0 \tag{2.29}$$

Note that two equations are needed to represent this entity. The associated dual entity, the absolute dual quadric Ω^*, however, can be represented as a single quadric. The canonical form is:

$$\Omega^* \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} . \tag{2.30}$$

Note that $\Pi_\infty = [0\,0\,0\,1]^\top$ is the null space of Ω^*. Let $M_\infty \sim [X\,Y\,Z\,0]^\top$ be a point of the plane at infinity, then that point in the plane at infinity is easily parameterized as $m_\infty \sim [X\,Y\,Z]^\top$. In this case the absolute conic can be represented as a 2D conic:

$$\omega_\infty \sim \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \omega_\infty^* \sim \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{2.31}$$

According to (2.28), applying a similarity transformation to M_∞ results in $m_\infty \mapsto m'_\infty \sim \sigma \mathbf{R} m_\infty$. Using equations (2.14),(2.15) and (2.20), it can now be verified that a similarity transformation leaves the absolute

Figure 2.4: *Affine (left) and metric (right) representation of a cube. The right angles and the identical lengths in the different directions of a cube give enough information to upgrade the structure from affine to metric.*

conic and its associated entities unchanged:

$$
\begin{bmatrix} \mathbf{I}_{3\times3} & 0_3 \\ 0_3^\top & 0 \end{bmatrix} \sim \begin{bmatrix} \sigma\mathbf{R} & \mathtt{t} \\ 0_3^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3\times3} & 0_3 \\ 0_3^\top & 0 \end{bmatrix} \begin{bmatrix} \sigma\mathbf{R} & \mathtt{t} \\ 0_3^\top & 1 \end{bmatrix}^\top \tag{2.32}
$$

and

$$
\mathbf{I}_{3\times3} \sim \sigma^{-1}\mathbf{R}^{-\top}\mathbf{I}_{3\times3}\mathbf{R}^{-1}\sigma^{-1} \qquad\qquad \mathbf{I}_{3\times3} \sim \sigma\mathbf{R}\mathbf{I}_{3\times3}\mathbf{R}^\top\sigma \tag{2.33}
$$

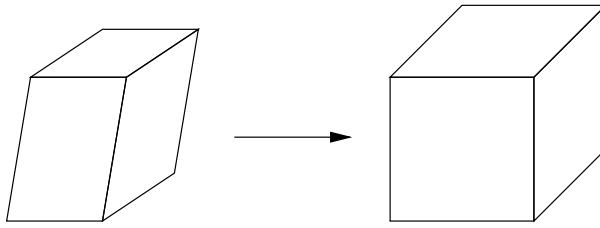Inversely, it is easy to prove that the projective transformations which leave the absolute quadric unchanged form the group of similarity transformations (the same could be done for the absolute conic and the plane at infinity):

$$
\begin{bmatrix} \mathbf{I}_{3\times3} & 0_3 \\ 0_3^\top & 0 \end{bmatrix} \sim \begin{bmatrix} \mathbf{A} & \mathtt{b} \\ \mathtt{c}^\top & d \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3\times3} & 0_3 \\ 0_3^\top & 0 \end{bmatrix} \begin{bmatrix} \mathbf{A}^\top & \mathtt{c} \\ \mathtt{b}^\top & d \end{bmatrix} \sim \begin{bmatrix} \mathbf{A}\mathbf{A}^\top & \mathbf{A}\mathtt{c} \\ \mathtt{c}^\top\mathbf{A}^\top & \mathtt{c}^\top\mathtt{c} \end{bmatrix}
$$

Therefore $\mathbf{A}\mathbf{A}^\top \sim \mathbf{I}_{3\times3}$ and $\mathtt{c} = 0_3$ which are exactly the constraints for a similarity transformation.

Angles can be measured using Laguerre's formula (see for example [132]). Assume two directions are characterized by their vanishing points $\mathtt{v}$ and $\mathtt{v}'$ in the plane at infinity (i.e. the intersection of a line with the plane at infinity indicating the direction). Compute the intersection points $\mathtt{j}$ and $\mathtt{j}'$ between the absolute conic and the line through the two vanishing points. The following formula based on the cross-ratio then gives the angle (with $i = \sqrt{-1}$):

$$
\alpha = \frac{1}{2i}\log\{\mathtt{v}_1,\mathtt{v}_2;\mathtt{j},\mathtt{j}'\} \tag{2.34}
$$

For two orthogonal planes Π and Π' the following equation must be satisfied:

$$
\Pi^\top\Omega^*\Pi' = 0 \tag{2.35}
$$

From projective or affine to metric In some cases it is needed to upgrade the projective or affine representation to metric. This can be done by retrieving the absolute conic or one of its associated entities. Since the conic is located in the plane at infinity, it is easier to retrieve it once this plane has been identified (i.e. the affine structure has been recovered). It is, however, possible to retrieve both entities at the same time. The absolute quadric Ω^* is especially suited for this purpose, since it encodes both entities at once.

Every known angle or ratio of lengths imposes a constraint on the absolute conic. If enough constraints are at hand, the conic can uniquely be determined. In Figure 2.4 the cube of Figure 2.1 is further upgraded to metric (i.e. the cube is transformed so that obtained angles are orthogonal and the sides all have equal length).

Once the absolute conic has been identified, the geometry can be upgraded from projective or affine to metric by bringing it to its canonical (metric) position. In Section 2.2.2 the procedure to go from projective to affine was explained. Therefore, we can restrict ourselves here to the upgrade from affine to metric. In this case, there must be an affine transformation which brings the absolute conic to its canonical position; or, inversely, from its canonical position to its actual position in the affine representation. Combining (2.23) and (2.20) yields

$$
\Omega^* \sim \begin{bmatrix} \mathbf{A} & \mathtt{a} \\ 0_3^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3\times3} & 0_3 \\ 0_3^\top & 0 \end{bmatrix} \begin{bmatrix} \mathbf{A}^\top & 0_3 \\ \mathtt{a}^\top & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}\mathbf{A}^\top & 0_3 \\ 0_3^\top & 0 \end{bmatrix} \tag{2.36}
$$

Under these circumstances the absolute conic and its dual have the following form (assuming the standard parameterization of the plane at infinity, i.e. $W = 0$):

$$\omega_\infty = \mathbf{A}^{-\top}\mathbf{A}^{-1} \text{ and } \omega_\infty^* = \mathbf{A}\mathbf{A}^\top \tag{2.37}$$

One possible choice for the transformation to upgrade from affine to metric is

$$\mathbf{T}_{AM} = \begin{bmatrix} \mathbf{A}^{-1} & 0_3 \\ 0_3^\top & 0 \end{bmatrix} \tag{2.38}$$

where a valid $\mathbf{A}$ can be obtained from Ω^* by Cholesky factorization or by singular value decomposition. Combining (2.25) and (2.38) the following transformation is obtained to upgrade the geometry from projective to metric at once

$$\mathbf{T}_{PM} = \mathbf{T}_{AM}\mathbf{T}_{PA} = \begin{bmatrix} \mathbf{A}^{-1} & 0_3 \\ \pi_\infty & 1 \end{bmatrix} \tag{2.39}$$

2.2.4   Euclidean stratum

For the sake of completeness, Euclidean geometry is briefly discussed. It does not differ much from metric geometry as we have defined it here. The difference is that the scale is fixed and that therefore not only relative lengths, but *absolute lengths* can be measured. Euclidean transformations have 6 degrees of freedom, 3 for orientation and 3 for translation. A Euclidean transformation has the following form

$$\mathbf{T}_E \sim \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.40}$$

with r_{ij} representing the coefficients of an orthonormal matrix, as described previously. If $\mathbf{R}$ is a rotation matrix (i.e. $\det \mathbf{R} = 1$) then, this transformation represents a rigid motion in space.

2.2.5   Overview of the different strata

The properties of the different strata are briefly summarized in Table 2.1 . The different geometric strata are presented. The number of degrees of freedom, transformations and the specific invariants are given for each stratum. Figure 2.5 gives an example of an object which is equivalent to a cube under the different geometric ambiguities. Note from the figure that for purposes of visualization at least a metric level should be reached (i.e. is perceived as a cube).

2.3   Conclusion

In this chapter some concepts of projective geometry were presented. These will allow us, in the next chapter, to described the projection from a scene into an image and to understand the intricate relationships which relate multiple views of a scene. Based on these concepts methods can be conceived that inverse this process and obtain 3D reconstructions of the observed scenes. This is the main subject of this text.

ambiguity	DOF	transformation	invariants
projective	15	$\mathbf{T}_P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$	cross-ratio
affine	12	$\mathbf{T}_A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$	relative distances along direction parallelism *plane at infinity*
metric	7	$\mathbf{T}_M = \begin{bmatrix} \sigma r_{11} & \sigma r_{12} & \sigma r_{13} & t_x \\ \sigma r_{21} & \sigma r_{22} & \sigma r_{23} & t_y \\ \sigma r_{31} & \sigma r_{32} & \sigma r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	relative distances angles *absolute conic*
Euclidean	6	$\mathbf{T}_E = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	absolute distances

Table 2.1: *Number of degrees of freedom, transformations and invariants corresponding to the different geometric strata (the coefficients r_{ij} form orthonormal matrices)*
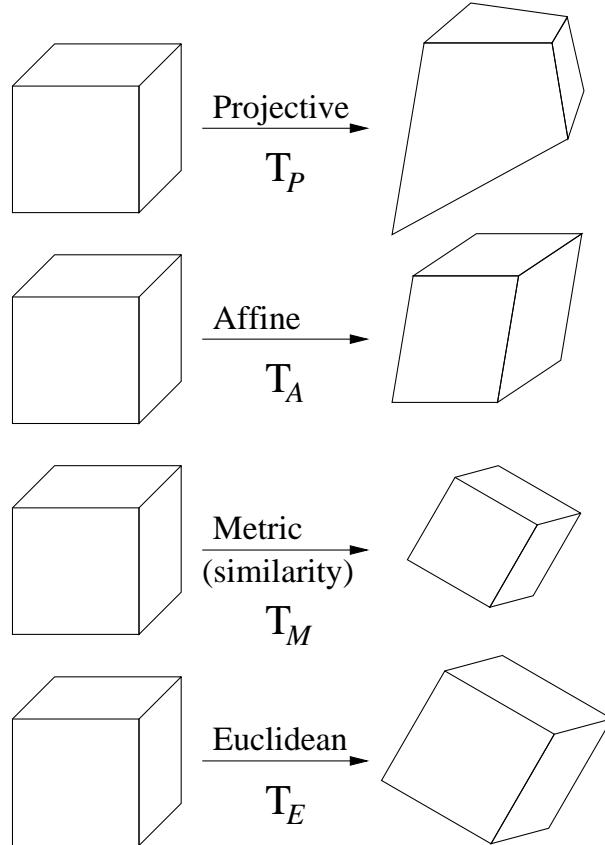
Figure 2.5: *Shapes which are equivalent to a cube for the different geometric ambiguities*

Chapter 3

Camera model and multiple view geometry

Before discussing how 3D information can be obtained from images it is important to know how images are formed. First, the camera model is introduced; and then some important relationships between multiple views of a scene are presented.

3.1 The camera model

In this work the perspective camera model is used. This corresponds to an ideal pinhole camera. The geometric process for image formation in a pinhole camera has been nicely illustrated by Dürer (see Figure 3.1). The process is completely determined by choosing a perspective projection center and a retinal plane. The projection of a scene point is then obtained as the intersection of a line passing through this point and the center of projection C with the retinal plane $\mathcal{R}$.

Most cameras are described relatively well by this model. In some cases additional effects (e.g. radial distortion) have to be taken into account (see Section 3.1.5).

3.1.1 A simple model

In the simplest case where the projection center is placed at the origin of the world frame and the image plane is the plane $Z = 1$, the projection process can be modeled as follows:

$$x = \tfrac{X}{Z} \quad y = \tfrac{Y}{Z} \tag{3.1}$$

For a world point (X, Y, Z) and the corresponding image point (x, y). Using the homogeneous representation of the points a linear projection equation is obtained:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.2}$$

This projection is illustrated in Figure 3.2. The optical axis passes through the center of projection C and is orthogonal to the retinal plane $\mathcal{R}$. It's intersection with the retinal plane is defined as the principal point c.

3.1.2 Intrinsic calibration

With an actual camera the focal length f (i.e. the distance between the center of projection and the retinal plane) will be different from 1, the coordinates of equation (3.2) should therefore be scaled with f to take this into account.

Figure 3.1: *Man Drawing a Lute (The Draughtsman of the Lute), woodcut 1525, Albrecht Dürer.*
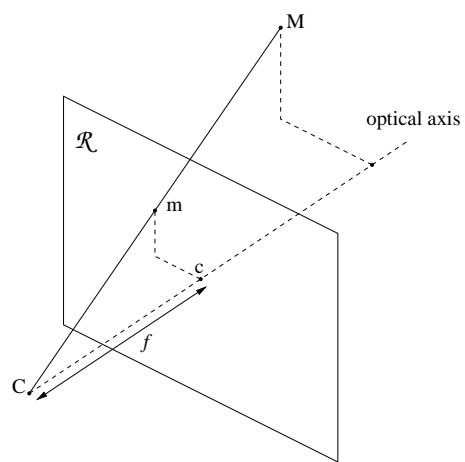
Figure 3.2: *Perspective projection*
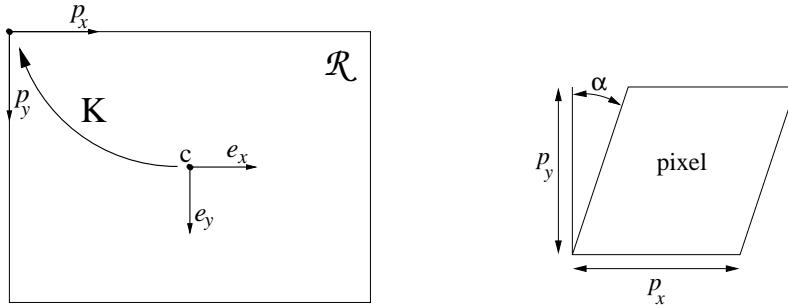
Figure 3.3: *From retinal coordinates to image coordinates*

In addition the coordinates in the image do not correspond to the physical coordinates in the retinal plane. With a CCD camera the relation between both depends on the size and shape of the pixels and of the position of the CCD chip in the camera. With a standard photo camera it depends on the scanning process through which the images are digitized.
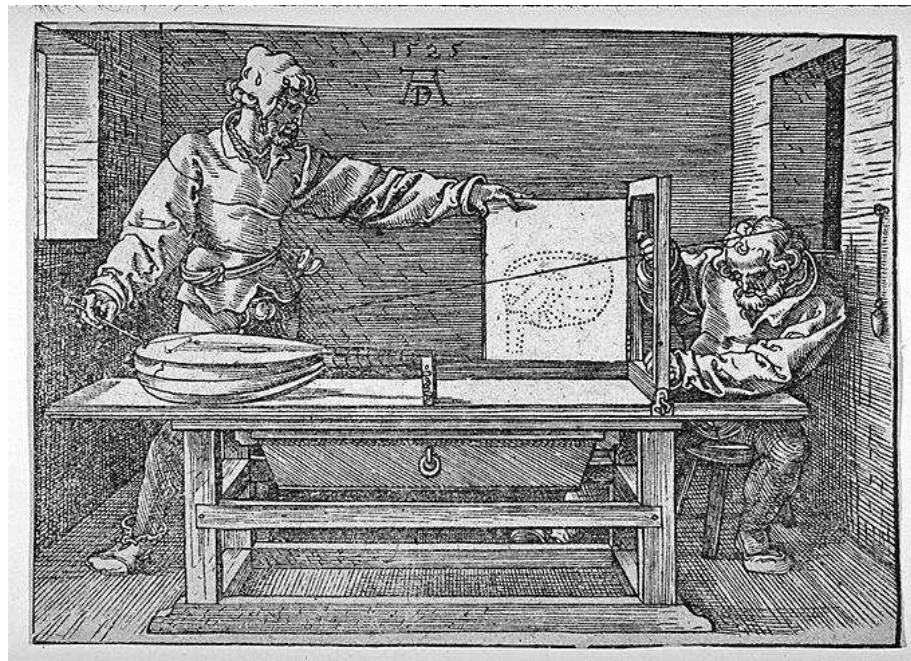
The transformation is illustrated in Figure 3.3. The image coordinates are obtained through the following equations:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{p_x} & (\tan\alpha)\frac{f}{p_y} & c_x \\ & \frac{f}{p_y} & c_y \\ & & 1 \end{bmatrix} \begin{bmatrix} x_{\mathcal{R}} \\ y_{\mathcal{R}} \\ 1 \end{bmatrix}$$

where p_x and p_y are the width and the height of the pixels, $\mathtt{c} = \begin{bmatrix} c_x\, c_y\, 1 \end{bmatrix}^\top$ is the principal point and α the skew angle as indicated in Figure 3.3. Since only the ratios $\frac{f}{p_x}$ and $\frac{f}{p_y}$ are of importance the simplified notations of the following equation will be used in the remainder of this text:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix} \begin{bmatrix} x_{\mathcal{R}} \\ y_{\mathcal{R}} \\ 1 \end{bmatrix} \tag{3.3}$$

with f_x and f_y being the focal length measured in width and height of the pixels, and s a factor accounting for the skew due to non-rectangular pixels. The above upper triangular matrix is called the *calibration matrix* of the camera; and the notation $\mathbf{K}$ will be used for it. So, the following equation describes the transformation from retinal coordinates to image coordinates.

$$\mathtt{m} = \mathbf{K}\mathtt{m}_{\mathcal{R}} \ . \tag{3.4}$$

For most cameras the pixels are almost perfectly rectangular and thus s is very close to zero. Furthermore, the principal point is often close to the center of the image. These assumptions can often be used, certainly to get a suitable initialization for more complex iterative estimation procedures.

For a camera with fixed optics these parameters are identical for all the images taken with the camera. For cameras which have zooming and focusing capabilities the focal length can obviously change, but also the principal point can vary. An extensive discussion of this subject can for example be found in the work of Willson [173, 171, 172, 174].

3.1.3 Camera motion

Motion of scene points can be modeled as follows

$$\mathtt{M}' = \begin{bmatrix} \mathbf{R} & \mathtt{t} \\ 0_3^\top & 1 \end{bmatrix} \mathtt{M} \tag{3.5}$$

with $\mathbf{R}$ a rotation matrix and $\mathtt{t} = \begin{bmatrix} t_x\, t_y\, t_z \end{bmatrix}^\top$ a translation vector.

The motion of the camera is equivalent to an inverse motion of the scene and can therefore be modeled as

$$\mathtt{M}' = \begin{bmatrix} \mathbf{R}^\top & -\mathbf{R}^\top \mathtt{t} \\ 0_3^\top & 1 \end{bmatrix} \mathtt{M}, \tag{3.6}$$

with $\mathbf{R}$ and $\mathtt{t}$ indicating the motion of the camera.

3.1.4 The projection matrix

Combining equations (3.2), (3.3) and (3.6) the following expression is obtained for a camera with some specific intrinsic calibration and with a specific position and orientation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}^\top & -\mathbf{R}^\top \mathtt{t} \\ 0_3^\top & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$

which can be simplified to

$$\mathtt{m} \sim \mathbf{K}[\mathbf{R}^\top \ \text{-}\mathbf{R}^\top \mathtt{t}]\mathtt{M} \tag{3.7}$$

or even

$$\mathtt{m} \sim \mathbf{P}\mathtt{M} \ . \tag{3.8}$$

The 3×4 matrix $\mathbf{P}$ is called the *camera projection matrix*.

Using (3.8) the plane corresponding to a back-projected image line $\mathtt{l}$ can also be obtained: Since $\mathtt{l}^\top \mathtt{m} \sim \mathtt{l}^\top \mathbf{P}\mathtt{M} \sim \Pi^\top \mathtt{M}$,

$$\Pi \sim \mathbf{P}^\top \mathtt{l} \tag{3.9}$$

The transformation equation for projection matrices can be obtained as described in paragraph 2.1.3. If the points of a calibration grid are transformed by the same transformation as the camera, their image points should stay the same:

$$\mathtt{m} \sim \mathbf{P}'\mathtt{M}' \sim \mathbf{P}\mathbf{T}^{-1}\mathbf{T}\mathtt{M} \sim \mathbf{P}\mathtt{M} \tag{3.10}$$

and thus

$$\mathbf{P} \mapsto \mathbf{P}' \sim \mathbf{P}\mathbf{T}^{-1} \tag{3.11}$$

The projection of the outline of a quadric can also be obtained. For a line in an image to be tangent to the projection of the outline of a quadric, the corresponding plane should be on the dual quadric. Substituting equation (3.9) in (2.17) the following constraint $\mathtt{l}^\top \mathbf{P}\mathbf{Q}^*\mathbf{P}^\top \mathtt{l} = 0$ is obtained for $\mathtt{l}$ to be tangent to the outline. Comparing this result with the definition of a conic (2.10), the following projection equation is obtained for quadrics (this results can also be found in [65]). :

$$\mathbf{C}^* \sim \mathbf{P}\mathbf{Q}^*\mathbf{P}^\top \ . \tag{3.12}$$

Relation between projection matrices and image homographies

The homographies that will be discussed here are collineations from $\mathcal{P}^2 \rightarrow \mathcal{P}^2$. A homography $\mathbf{H}$ describes the transformation from one plane to another. A number of special cases are of interest, since the image is also a plane. The projection of points of a plane into an image i can be described through a homography $\mathbf{H}_{\Pi i}$. The matrix representation of this homography is dependent on the choice of the projective basis in the plane.

As an image is obtained by perspective projection, the relation between points $\mathtt{M}_\Pi$ belonging to a plane Π in 3D space and their projections $\mathtt{m}_{\Pi i}$ in the image is mathematically expressed by a homography $\mathbf{H}_{\Pi i}$. The matrix of this homography is found as follows. If the plane Π is given by $\Pi \sim [\pi^\top \ 1]^\top$ and the point $\mathtt{M}_\Pi$ of Π is represented as $\mathtt{M}_\Pi \sim [\mathtt{m}_\Pi^\top \ 1]^\top$, then $\mathtt{M}_\Pi$ belongs to Π if and only if $0 = \Pi^\top \mathtt{M}_\Pi = \pi^\top \mathtt{m}_\Pi + 1$. Hence,

$$\mathtt{M}_\Pi \sim \begin{bmatrix} \mathtt{m}_\Pi \\ 1 \end{bmatrix} = \begin{bmatrix} \mathtt{m}_\Pi \\ -\pi^\top \mathtt{m}_\Pi \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3\times 3} \\ -\pi^\top \end{bmatrix} \mathtt{m}_\Pi \ . \tag{3.13}$$

Now, if the camera projection matrix is $\mathbf{P}_i = [\mathbf{A}_i|\mathbf{a}_i]$, then the projection $\mathbf{m}_{\Pi i}$ of $\mathbf{M}_\Pi$ onto the image is

$$
\begin{aligned}
\mathbf{m}_{\Pi i} \sim \mathbf{P}_i \mathbf{M}_\Pi &= [\mathbf{A}_i|\mathbf{a}_i]\begin{bmatrix} \mathbf{I}_{3\times 3} \\ -\pi^\top \end{bmatrix}\mathbf{m}_\Pi \\
&= [\mathbf{A}_i - \mathbf{a}_i\pi^\top]\mathbf{m}_\Pi \ .
\end{aligned}
\tag{3.14}
$$

Consequently, $\mathbf{H}_{\Pi i} \sim \mathbf{A}_i - \mathbf{a}_i\pi^\top$.

Note that for the specific plane $\Pi_{\texttt{REF}} = [0\,0\,0\,1]^\top$ the homographies are simply given by $\mathbf{H}_{\texttt{REF}i} \sim \mathbf{A}_i$.

It is also possible to define homographies which describe the transfer from one image to the other for points and other geometric entities located on a specific plane. The notation $\mathbf{H}_{ij}^\Pi$ will be used to describe such a homography from view i to j for a plane Π. These homographies can be obtained through the following relation $\mathbf{H}_{ij}^\Pi = \mathbf{H}_{\Pi j}\mathbf{H}_{\Pi i}^{-1}$ and are independent to reparameterizations of the plane (and thus also to a change of basis in $\mathcal{P}^3$).

In the metric and Euclidean case, $\mathbf{A}_i = \mathbf{K}_i\mathbf{R}_i^\top$ and the plane at infinity is $\Pi_\infty = [0001]^\top$. In this case, the homographies for the plane at infinity can thus be written as:

$$
\mathbf{H}_{ij}^\infty = \mathbf{K}_i\mathbf{R}_{ij}^\top\mathbf{K}_i^{-1} ,
\tag{3.15}
$$

where $\mathbf{R}_{ij} = \mathbf{R}_i^\top\mathbf{R}_j$ is the rotation matrix that describes the relative orientation from the jth camera with respect top the ith one.

In the projective and affine case, one can assume that $\mathbf{P}_1 = [\mathbf{I}_{3\times 3}|\mathbf{0}_3]$ (since in this case $\mathbf{K}_i$ is unknown). In that case, the homographies $\mathbf{H}_{\Pi 1} \sim \mathbf{I}_{3\times 3}$ for all planes; and thus, $\mathbf{H}_{1i}^{\texttt{REF}} = \mathbf{H}_{\texttt{REF}i}$. Therefore $\mathbf{P}_i$ can be factorized as

$$
\mathbf{P}_i = [\mathbf{H}_{1i}^{\texttt{REF}}|\mathbf{e}_{1i}]
\tag{3.16}
$$

where $\mathbf{e}_{1i}$ is the projection of the center of projection of the first camera (in this case, $[0\,0\,0\,1]^\top$) in image i. This point $\mathbf{e}_{1i}$ is called the *epipole*, for reasons which will become clear in Section 3.2.1.

Note that this equation can be used to obtain $\mathbf{H}_{1i}^{\texttt{REF}}$ and $\mathbf{e}_{1i}$ from $\mathbf{P}_i$, but that due to the unknown relative scale factors $\mathbf{P}_i$ can, in general, not be obtained from $\mathbf{H}_{1i}^{\texttt{REF}}$ and $\mathbf{e}_{1i}$. Observe also that, in the affine case (where $\Pi_\infty = [0001]^\top$), this yields $\mathbf{P}_i = [\mathbf{H}_{1i}^\infty|\mathbf{e}_{1i}]$.

Combining equations (3.14) and (3.16), one obtains

$$
\mathbf{H}_{1i}^\Pi = \mathbf{H}_{1i}^{\texttt{REF}} - \mathbf{e}_{1i}\pi^\top
\tag{3.17}
$$

This equation gives an important relationship between the homographies for all possible planes. Homographies can only differ by a term $\mathbf{e}_{1i}[1 - \pi']^\top$. This means that in the projective case the homographies for the plane at infinity are known up to 3 common parameters (i.e. the coefficients of π_∞ in the projective space).

Equation (3.16) also leads to an interesting interpretation of the camera projection matrix:

$$
\mathbf{m}_1 \sim [\mathbf{I}_{3\times 3}|\mathbf{0}_3]\begin{bmatrix} \mathbf{m} \\ 1 \end{bmatrix} = \mathbf{m}
\tag{3.18}
$$

$$
\mathbf{m}_i \sim [\mathbf{H}_{1i}^{\texttt{REF}}|\mathbf{e}_{1i}]\begin{bmatrix} \mathbf{m} \\ 1 \end{bmatrix} = \mathbf{H}_{1i}^{\texttt{REF}}\mathbf{m} + \mathbf{e}_{1i}
\tag{3.19}
$$

$$
= \lambda\mathbf{H}_{1i}^{\texttt{REF}}\mathbf{m}_1 + \mathbf{e}_{1i} = \mathbf{P}_i(\lambda\begin{bmatrix} \mathbf{m}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0}_3 \\ 1 \end{bmatrix})
\tag{3.20}
$$

In other words, a point can thus be parameterized as being on the line through the optical center of the first camera (i.e. $[0001]^\top$) and a point in the reference plane $\Pi_{\texttt{REF}}$. This interpretation is illustrated in Figure 3.4.

3.1.5 Deviations from the camera model

The perspective camera model describes relatively well the image formation process for most cameras. However, when high accuracy is required or when low-end cameras are used, additional effects have to be taken into account.
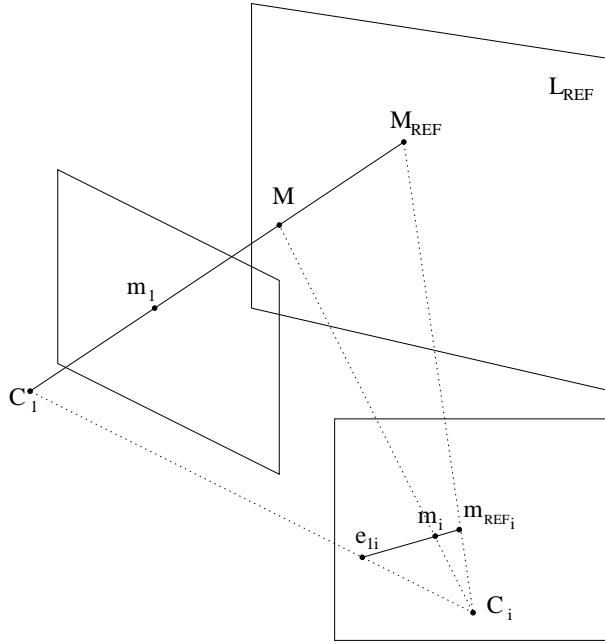
Figure 3.4: *A point* M *can be parameterized as* $C_1 + \lambda M_{REF}$. *Its projection in another image can then be obtained by transferring* m_1 *according to* Π_{REF} *(i.e. with* $\mathbf{H}_{1i}^{REF}$*) to image* i *and applying the same linear combination with the projection* e_{1i} *of* C_1 *(i.e.* $m_i \sim e_{1i} + \lambda \mathbf{H}_{1i}^{REF} m_1$*).*

The failures of the optical system to bring all light rays received from a point object to a single image point or to a prescribed geometric position should then be taken into account. These deviations are called aberrations. Many types of aberrations exist (e.g. astigmatism, chromatic aberrations, spherical aberrations, coma aberrations, curvature of field aberration and distortion aberration). It is outside the scope of this work to discuss them all. The interested reader is referred to the work of Willson [173] and to the photogrammetry literature [137].

Many of these effects are negligible under normal acquisition circumstances. Radial distortion, however, can have a noticeable effect for shorter focal lengths. Radial distortion is a linear displacement of image points radially to or from the center of the image, caused by the fact that objects at different angular distance from the lens axis undergo different magnifications.

It is possible to cancel most of this effect by warping the image. The coordinates in undistorted image plane coordinates (x, y) can be obtained from the observed image coordinates (x_o, y_o) by the following equation:

$$\begin{aligned} x &= x_o + (x_o - c_x)(K_1 r^2 + K_2 r^4 + \ldots) \\ y &= y_o + (y_o - c_y)(K_1 r^2 + K_2 r^4 + \ldots) \end{aligned} \tag{3.21}$$

where K_1 and K_2 are the first and second parameters of the radial distortion and

$$r^2 = (x_o - c_x)^2 + (y_o - c_y)^2 \quad .$$

Note that it can sometimes be necessary to allow the center of radial distortion to be different from the principal point [174].

When the focal length of the camera changes (through zoom or focus) the parameters K_1 and K_2 will also vary. In a first approximation this can be modeled as follows:

$$\begin{aligned} x &= x_o + (x_o - c_x)(K_{f1} \tfrac{r^2}{f^2} + K_{f2} \tfrac{r^4}{f^4} + \ldots) \\ y &= y_o + (y_o - c_y)(K_{f1} \tfrac{r^2}{f^2} + K_{f2} \tfrac{r^4}{f^4} + \ldots) \end{aligned} \tag{3.22}$$

Due to the changes in the lens system this is only an approximation, except for digital zooms where (3.22) should be exactly satisfied.

Figure 3.5: *Correspondence between two views. Even when the exact position of the 3D point* M *corresponding to the image point* 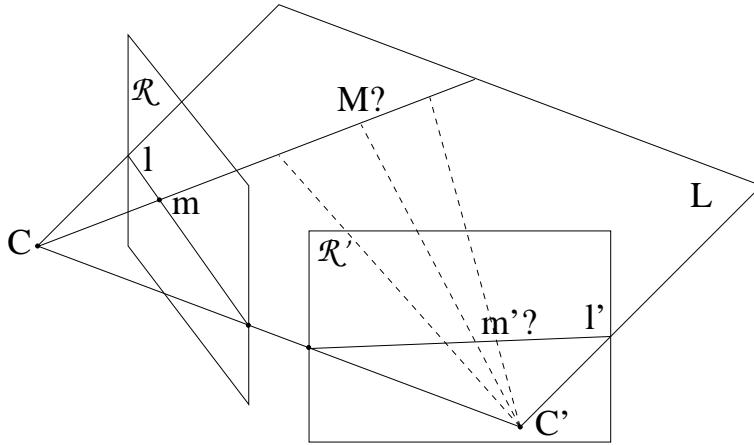m *is not known, it has to be on the line through* C *which intersects the image plane in* m. *Since this line projects to the line* l' *in the other image, the corresponding point* m' *should be located on this line. More generally, all the points located on the plane defined by* C, C' *and* M *have their projection on* l *and* l'.

3.2 Multi view geometry

Different views of a scene are not unrelated. Several relationships exist between two, three or more images. These are very important for the calibration and reconstruction from images. Many insights in these relationships have been obtained in recent years.

3.2.1 Two view geometry

In this section the following question will be addressed: *Given an image point in one image, does this restrict the position of the corresponding image point in another image?* It turns out that it does and that this relationship can be obtained from the calibration or even from a set of prior point correspondences.

Although the exact position of the scene point M is not known, it is bound to be on the line of sight of the corresponding image point m. This line can be projected in another image and the corresponding point m' is bound to be on this projected line l'. This is illustrated in Figure 3.5. In fact all the points on the plane Π defined by the two projection centers and M have their image on l'. Similarly, all these points are projected on a line l in the first image. l and l' are said to be in *epipolar correspondence* (i.e. the corresponding point of every point on l is located on l', and vice versa).

Every plane passing through both centers of projection C and C' results in such a set of corresponding epipolar lines, as can be seen in Figure 3.6. All these lines pass through two specific points e and e'. These points are called the *epipoles*, and they are the projection of the center of projection in the opposite image.

This epipolar geometry can also be expressed mathematically. The fact that a point m is on a line l can be expressed as $l^\top m = 0$. The line passing trough m and the epipole e is

$$l \sim [e]_\times m\,, \tag{3.23}$$

with $[e]_\times$ the antisymmetric 3×3 matrix representing the vectorial product with e.

From (3.9) the plane Π corresponding to l is easily obtained as $\Pi \sim P^\top l$ and similarly $\Pi \sim {P'}^\top l'$. Combining these equations gives:

$$l' \sim \left({P'}^\top\right)^\dagger P^\top l \equiv H^{-\top} l \tag{3.24}$$

with † indicating the Moore-Penrose pseudo-inverse. The notation $H^{-\top}$ is inspired by equation (2.7). Substituting (3.23) in (3.24) results in

$$l' \sim H^{-\top}[e]_\times m \ .$$
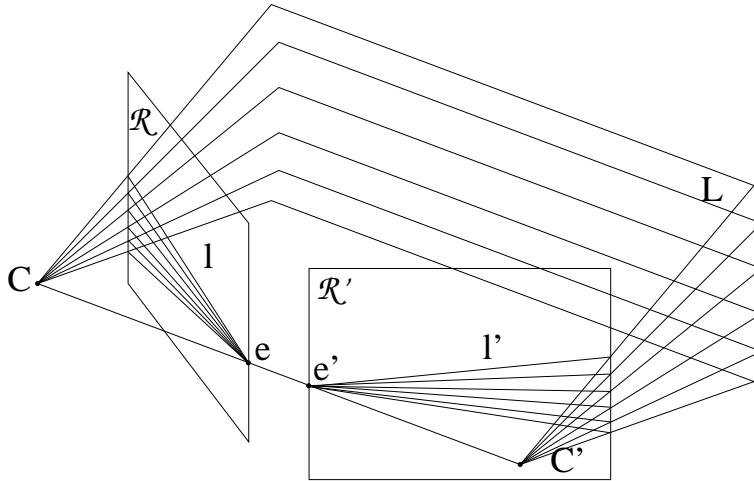
Figure 3.6: *Epipolar geometry. The line connecting* C *and* C′ *defines a bundle of planes. For every one of these planes a corresponding line can be found in each image, e.g. for* Π *these are* l *and* l′. *All 3D points located in* Π *project on* l *and* l′ *and thus all points on* l *have their corresponding point on* l′ *and vice versa. These lines are said to be in epipolar correspondence. All these epipolar lines must pass through* e *or* e′, *which are the intersection points of the line* CC′ *with the retinal planes* R *and* R′ *respectively. These points are called the epipoles.*

Defining $\mathbf{F} = \mathbf{H}^{-\top}[\mathbf{e}]_\times$, we obtain

$$\mathbf{l}' \sim \mathbf{F}\mathbf{m}\,, \tag{3.25}$$

and thus,

$$\mathbf{m}'^\top \mathbf{F}\mathbf{m} = 0 \ . \tag{3.26}$$

This matrix $\mathbf{F}$ is called the *fundamental matrix*. These concepts were introduced by Faugeras [31] and Hartley [46]. Since then many people have studied the properties of this matrix (e.g. [82, 83]) and a lot of effort has been put in robustly obtaining this matrix from a pair of uncalibrated images [153, 154, 177].

Having the calibration, $\mathbf{F}$ can be computed and a constraint is obtained for corresponding points. When the calibration is not known equation (3.26) can be used to compute the fundamental matrix $\mathbf{F}$. Every pair of corresponding points gives one constraint on $\mathbf{F}$. Since $\mathbf{F}$ is a 3×3 matrix which is only determined up to scale, it has $3 \times 3 - 1$ unknowns. Therefore 8 pairs of corresponding points are sufficient to compute $\mathbf{F}$ with a linear algorithm.

Note from (3.25) that $\mathbf{F}\mathbf{e} = 0$, because $[\mathbf{e}]_\times \mathbf{e} = 0$. Thus, rank $\mathbf{F} = 2$. This is an additional constraint on $\mathbf{F}$ and therefore 7 point correspondences are sufficient to compute $\mathbf{F}$ through a nonlinear algorithm. In Section 4.2 the robust computation of the fundamental matrix from images will be discussed in more detail.

Relation between the fundamental matrix and image homographies

There also exists an important relationship between the homographies $\mathbf{H}_{ij}^{\Pi}$ and the fundamental matrices $\mathbf{F}_{ij}$. Let $\mathbf{m}_i$ be a point in image i. Then $\mathbf{m}_j \sim \mathbf{H}_{ij}^{\Pi}\mathbf{m}_i$ is the corresponding point for the plane Π in image j. Therefore, $\mathbf{m}_j$ is located on the corresponding epipolar line; and,

$$(\mathbf{H}_{ij}^{\Pi}\mathbf{m}_i)^\top \mathbf{F}_{ij}\mathbf{m}_i = 0 \tag{3.27}$$

should be verified. Moreover, equation (3.27) holds for every image point $\mathbf{m}_i$. Since the fundamental matrix maps points to corresponding epipolar lines, $\mathbf{F}_{ij}\mathbf{m}_i \sim \mathbf{e}_{ij} \times \mathbf{m}_j$ and equation (3.27) is equivalent to $\mathbf{m}_j^\top [\mathbf{e}_{ij}]_\times \mathbf{H}_{ij}^{\Pi}\mathbf{m}_i = 0$. Comparing this equation with $\mathbf{m}_j^\top \mathbf{F}_{ij}\mathbf{m}_i = 0$, and using that these equations must hold for all image points $\mathbf{m}_i$ and $\mathbf{m}_j$ lying on corresponding epipolar lines, it follows that:

$$\mathbf{F}_{ij} \sim [\mathbf{e}_{ij}]_\times \mathbf{H}_{ij}^{\Pi} \ . \tag{3.28}$$

Let $\mathtt{l}_j$ be a line in image j and let Π be the plane obtained by back-projecting $\mathtt{l}_j$ into space. If $\mathtt{m}_{\Pi i}$ is the image of a point of this plane projected in image i, then the corresponding point in image j must be located on the corresponding epipolar line (i.e. $\mathbf{F}_{ij}\mathtt{m}_{\Pi i}$). Since this point is also located on the line $\mathtt{l}_j$ it can be uniquely determined as the intersection of both (if these lines are not coinciding): $\mathtt{l}_j \times \mathbf{F}_{ij}\mathtt{m}_{\Pi i}$. Therefore, the homography $\mathbf{H}_{ij}^{\Pi}$ is given by $[\mathtt{l}_j]_\times \mathbf{F}_{ij}$. Note that, since the image of the plane Π is a line in image j, this homography is not of full rank. An obvious choice to avoid coincidence of $\mathtt{l}_j$ with the epipolar lines, is $\mathtt{l}_j \sim \mathtt{e}_{ij}$ since this line does certainly not contain the epipole (i.e. $\mathtt{e}_{ij}^\top \mathtt{e}_{ij} \neq 0$). Consequently,

$$[\mathtt{e}_{ij}]_\times \mathbf{F}_{ij} \tag{3.29}$$

corresponds to the homography of a plane. By combining this result with equations (3.16) and (3.17) one can conclude that it is always possible to write the projection matrices for two views as

$$\begin{array}{rcl} \mathbf{P}_1 & = & [\mathbf{I}_{3\times 3} \mid \mathbf{0}_3] \\ \mathbf{P}_2 & = & [[\mathtt{e}_{12}]_\times \mathbf{F}_{12} - \mathtt{e}_{12}\pi^\top \mid \mathtt{e}_{12}] \end{array} \tag{3.30}$$

Note that this is an important result, since it means that a projective camera setup can be obtained from the fundamental matrix which can be computed from 7 or more matches between two views. Note also that this equation has 4 degrees of freedom (i.e. the 3 coefficients of π and the arbitrary relative scale between $\mathbf{F}_{12}$ and $\mathtt{e}_{12}$). Therefore, this equation can only be used to instantiate a new frame (i.e. an arbitrary projective representation of the scene) and not to obtain the projection matrices for all the views of a sequence (i.e. compute $\mathbf{P}_3, \mathbf{P}_4, \ldots$). How this can be done is explained in Section 5.2.

3.2.2   Three view geometry

Considering three views it is, of course, possible to group them in pairs and to get the two view relationships introduced in the last section. Using these pairwise epipolar relations, the projection of a point in the third image can be predicted from the coordinates in the first two images. This is illustrated in Figure 3.7. The point in the third image is determined as the intersection of the two epipolar lines. This computation, however, is not always very well conditioned. When the point is located in the trifocal plane (i.e. the plane going through the three centers of projection), it is completely undetermined.

Fortunately, there are additional constraints between the images of a point in three views. When the centers of projection are not coinciding, a point can always be reconstructed from two views. This point then projects to a unique point in the third image, as can be seen in Figure 3.7, even when this point is located in the trifocal plane. For two views, no constraint is available to restrict the position of corresponding lines. Indeed, back-projecting a line forms a plane, the intersection of two planes always results in a line. Therefore, no constraint can be obtained from this. But, having three views, the image of the line in the third view can be predicted from its location in the first two images, as can be seen in Figure 3.8. Similar to what was derived for two views, there are multi linear relationships relating the positions of points and/or lines in three images [140]. The coefficients of these multi linear relationships can be organized in a tensor which describes the relationships between points [135] and lines [49] or any combination thereof [51]. Several researchers have worked on methods to compute the trifocal tensor (e.g. see [151, 152]).

The trifocal tensor $\mathbf{T}$ is a $3 \times 3 \times 3$ tensor. It contains 27 parameters, only 18 of which are independent due to additional nonlinear constraints. The trilinear relationship for a point is given by the following equation[1]:

$$m_i(m_j' m_k'' T_{i33} - m_k'' T_{ij3} - m_j' T_{i3k} + T_{ijk}) = 0 \tag{3.31}$$

Any triplet of corresponding points should satisfy this constraint.

A similar constraint applies for lines. Any triplet of corresponding lines should satisfy:

$$l_i \sim l_j' l_k'' T_{ijk}$$

---

[1]The Einstein convention is used (i.e. indices that are repeated should be summed over).
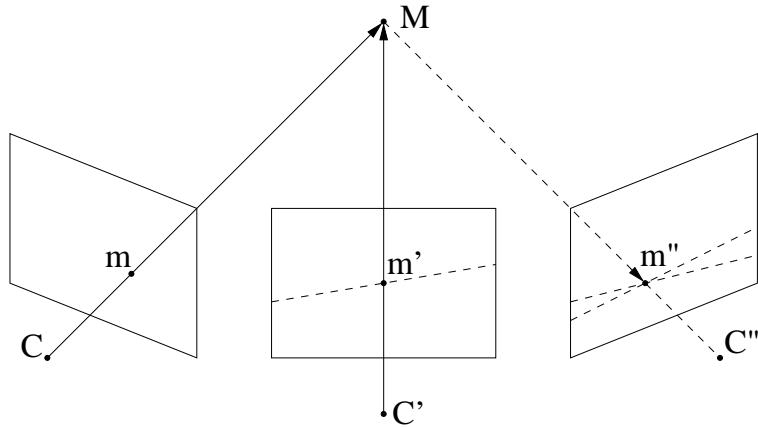
Figure 3.7: *Relation between the image of a point in three views. The epipolar lines of points* m *and* m′ *could be used to obtain* m″. *This does, however, not exhaust all the relations between the three images. For a point located in the trifocal plane (i.e. the plane defined by* C, C′ *and* C″*) this would not give a unique solution, although the 3D point could still be obtained from its image in the first two views and then be projected to* m″. *Therefore, one can conclude that in the three view case not all the information is described by the epipolar geometry. These additional relationships are described by the trifocal tensor.*

Figure 3.8: *Relation between the image of a line in three images. While in the two view case no constraints are available for lines, in the three view case it is also possible to predict the position of a line in a third image from its projection in the other two. This transfer is also described by the trifocal tensor.*

3.2.3 Multi view geometry

Many people have been studying multi view relationships [59, 156, 34]. Without going into detail we would like to give some intuitive insights to the reader. For a more in depth discussion the reader is referred to [88].

An image point has 2 degrees of freedom. But n images of a 3D point do not have $2n$ degrees of freedom, but only 3. So, there must be $2n - 3$ independent constraints between them. For lines, which also have 2 degrees of freedom in the image, but 4 in 3D space, n images of a line must satisfy $2n - 4$ constraints.

Some more properties of these constraints are explained here. A line can be back-projected into space linearly (3.9). A point can be seen as the intersection of two lines. To correspond to a real point or line the planes resulting from the backprojection must all intersect in a single point or line. This is easily expressed in terms of determinants, i.e. $|\Pi_1 \Pi_2 \Pi_3 \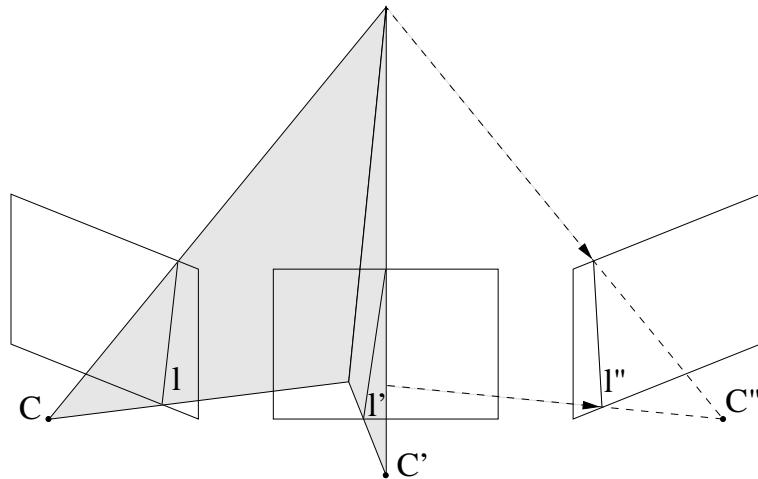Pi_4| = 0$ for points and that all the 3×3 subdeterminants of $[\Pi_1 \Pi_2 \Pi_3]$ should be zero for lines. This explains why the constraints are multi linear, since this is a property of columns of a determinant. In addition no constraints combining more than 4 images exist, since with 4-vectors (i.e. the representation of the planes) maximum 4×4 determinants can be obtained. The twofocal (i.e. the fundamental matrix) and the trifocal tensors have been discussed in the previous paragraphs, recently Hartley [54] proposed an algorithm for the practical computation of the quadrifocal tensor.

3.3 Conclusion

In this chapter some important concepts were introduced. A geometric description of the image formation process was given and the camera projection matrix was introduced. Some important relationships between multiple views were also derived. The insights obtained by carefully studying these properties have shown that it is possible to retrieve a relative calibration of a two view camera setup from point matches only. This is an important result which will be exploited further on to obtain a 3D reconstruction starting from the images.

Chapter 4

Relating images

Starting from a collection of images or a video sequence the first step consists in relating the different images to each other. This is not a easy problem. A restricted number of corresponding points is sufficient to determine the geometric relationship or *multi-view constraints* between the images. Since not all points are equally suited for matching or tracking (e.g. a pixel in a homogeneous region), the first step consist of selecting a number of interesting points or *feature points*. Some approaches also use other features, such as lines or curves, but these will not be discussed here. Depending on the type of image data (i.e. video or still pictures) the feature points are tracked or matched and a number of potential correspondences are obtained. From these the multi-view constraints can be computed. However, since the correspondence problem is an ill-posed problem, the set of corresponding points can be contaminated with an important number of wrong matches or *outliers*. In this case, a traditional least-squares approach will fail and therefore a robust method is needed. Once the multi-view constraints have been obtained they can be used to guide the search for additional correspondences. These can then be used to further refine the results for the multi-view constraints.

4.1   Feature extraction and matching

Before discussing the extraction of feature points it is necessary to have a measure to compare parts of images. The extraction and matching of features is based on these measures. Besides the simple point feature a more advanced type of feature is also presented.

4.1.1   Comparing image regions

Image regions are typically compared using sum-of-square-differences (SSD) or normalized cross-correlation (NCC). Consider a window W in image I and a corresponding region $\mathbf{T}(W)$ in image J. The *dissimilarity* between two image regions based on SSD is given by

$$D = \int \int_W \left[J(\mathbf{T}(x,y)) - I(x,y) \right]^2 w(x,y) dx dy \qquad (4.1)$$

where $w(x,y)$ is a weighting function that is defined over W. Typically, $w(x,y) = 1$ or it is a Gaussian. The *similarity* measure between two image regions based on NCC is given by

$$S = \frac{\int \int_W (J(\mathbf{T}(x,y)) - \bar{J}).(I(x,y) - \bar{I})w(x,y)dxdy}{\sqrt{\int \int_W (J(\mathbf{T}(x,y)) - \bar{J})w(x,y)dxdy}.\sqrt{\int \int_W (I(x,y) - \bar{I})w(x,y)dxdy}} \qquad (4.2)$$

with $\bar{J} = \int \int_W J(T(x,y))dxdy$ and $\bar{I} = \int \int_W I(x,y)dxdy$ the mean image intensity in the considered region. Note that this last measure is invariant to global intensity and contrast changes over the considered regions.
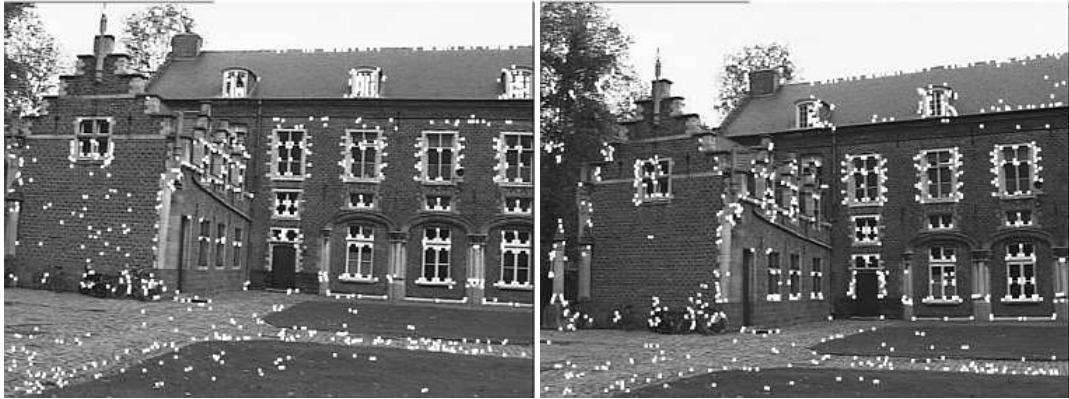
Figure 4.1: Two images with extracted corners

4.1.2   Feature point extraction

One of the most important requirements for a feature point is that it can be differentiated from its neighboring image points. If this were not the case, it wouldn't be possible to match it uniquely with a corresponding point in another image. Therefore, the neighborhood of a feature should be sufficiently different from the neighborhoods obtained after a small displacement.

A second order approximation of the dissimilarity, as defined in Eq. (4.1), between a image window W and a slightly translated image window is given by

$$D(\Delta x, \Delta y) = \left[\begin{array}{c} \Delta x \\ \Delta y \end{array} \right] \mathbf{M} \left[\begin{array}{cc} \Delta x & \Delta y \end{array} \right] \text{ with } \mathbf{M} = \int \int_W \left[\begin{array}{c} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{array} \right] \left[\begin{array}{cc} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{array} \right] w(x,y) dx dy \quad (4.3)$$

To ensure that no displacement exists for which D is small, the eigenvalues of $\mathbf{M}$ should both be large. This can be achieved by enforcing a minimal value for the smallest eigenvalue [136] or alternatively for the following corner response function $R = \det \mathbf{M} - k (\text{trace } \mathbf{M})^2$ [45] where k is a parameter set to 0.04 (a suggestion of Harris). In the case of tracking this is sufficient to ensure that features can be tracked from one video frame to the next. In this case it is natural to use the tracking neighborhood to evaluate the quality of a feature (e.g. a 7×7 window with $w(x,y) = 1$). Tracking itself is done by minimizing Eq. 4.1 over the parameters of $\mathbf{T}$. For small steps a translation is sufficient for $\mathbf{T}$. To evaluate the accumulated difference from the start of the track it is advised to use an affine motion model.

In the case of separate frames as obtained with a still camera, there is the additional requirement that as much image points originating from the same 3D points as possible should be extracted. Therefore, only local maxima of the corner response function are considered as features. Sub-pixel precision can be achieved through quadratic approximation of the neighborhood of the local maxima. A typical choice for $w(x)$ in this case is a Gaussian with $\sigma = 0.7$. Matching is typically done by comparing small, e.g. 7×7, windows centered around the feature through SSD or NCC. This measure is only invariant to image translations and can therefore not cope with too large variations in camera pose.

To match images that are more widely separated, it is required to cope with a larger set of image variations. Exhaustive search over all possible variations is computationally untractable. A more interesting approach consists of extracting a more complex feature that not only determines the position, but also the other unknowns of a local affine transformation [164] (see Section 4.1.3).

In practice often far too much corners are extracted. In this case it is often interesting to first restrict the numbers of corners before trying to match them. One possibility consists of only selecting the corners with a value R above a certain threshold. This threshold can be tuned to yield the desired number of features. Since for some scenes most of the strongest corners are located in the same area, it can be interesting to refine this scheme further to ensure that in every part of the image a sufficient number of corners are found.

In figure 4.1 two images are shown with the extracted corners. Note that it is not possible to find the corresponding corner for each corner, but that for many of them it is.
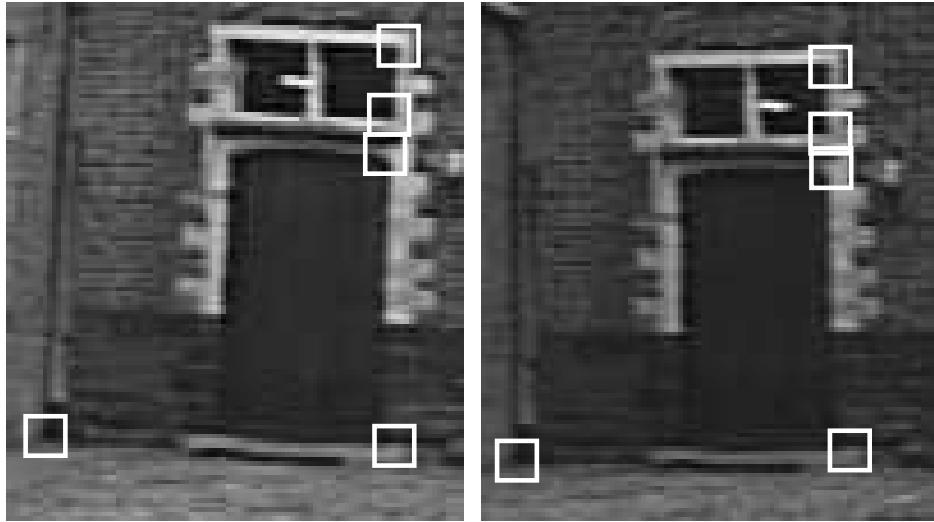
Figure 4.2: Detail of the images of figure 4.1 with 5 corresponding corners.
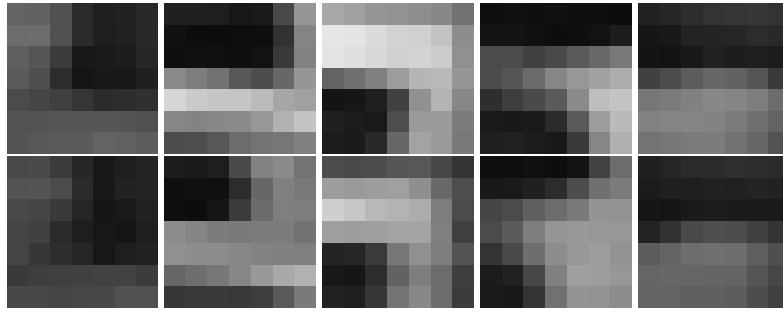
Figure 4.3: Local neighborhoods of the 5 corners of figure 4.2.

In figure 4.2 corresponding parts of two images are shown. In each the position of 5 corners is indicated. In figure 4.3 the neighborhood of each of these corners is shown. The intensity cross-correlation was computed for every possible combination. This is shown in Table 4.1. It can be seen that in this case the correct pair matches all yield the highest cross-correlation values (i.e. highest values on diagonal). However, the combination 2-5, for example, comes very close to 2-2. In practice, one can certainly not rely on the fact that all matches will be correct and automatic matching procedures should therefore be able to deal with important fraction of outliers. Therefore, further on robust matching procedures will be introduced.

If one can assume that the motion between two images is small (which is needed anyway for the intensity cross-correlation measure to yield good results), the location of the feature can not change widely between two consecutive views. This can therefore be used to reduce the combinatorial complexity of the

0.9639	-0.3994	-0.1627	-0.3868	0.1914
-0.0533	**0.7503**	-0.4677	0.5115	0.7193
-0.1826	-0.3905	**0.7730**	0.1475	-0.7457
-0.2724	0.4878	0.1640	**0.7862**	0.2077
0.0835	0.5044	-0.4541	0.2802	**0.9876**

Table 4.1: Intensity cross-correlation values for all possible combinations of the 5 corners indicated figure 4.2.

Figure 4.4: Based on the edges in the neighborhood of a corner point p an affinely invariant region is determined up to two parameters l_1 and l_2.

matching. Only features with similar coordinates in both images will be compared. For a corner located at (x, y), only the corners of the other image with coordinates located in the interval $[x - w_i, x + w_i] \times [y - h_i, y + h_i]$. w_i and h_i are typically 10% or 20% of the image.

4.1.3   Matching using affinely invariant regions

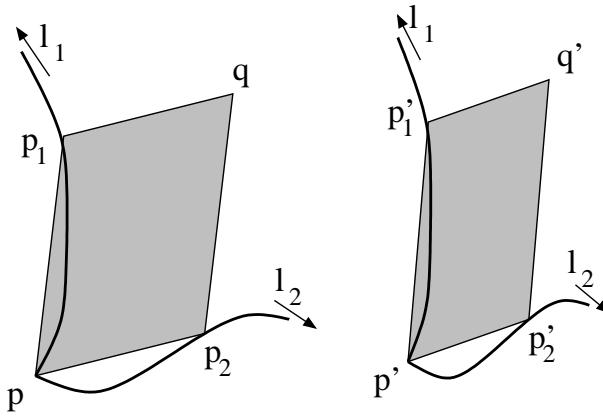One can note that the similarity measure presented in the previous section is only invariant to translation and offsets in the intensity values. If important rotation or scaling takes place the similarity measure is not appropriate. The same is true when the lighting conditions differ too much. Therefore the cross-correlation based approach can only be used between images for which the camera poses are not too far apart.

In this section a more advanced matching procedure is presented that can deal with much larger changes in viewpoint and illumination [164, 163]. As should be clear from the previous discussion, it is important that pixels corresponding to the same part of the surface are used for comparison during matching. By assuming that the surface is locally planar and that there is no perspective distortion, local transformations of pixels from one image to the other are described by 2D affine transformations. Such a transformation is defined by three points. At this level we only have one, i.e. the corner under consideration, and therefore need two more. The idea is to go look for them along edges which pass through the point of interest. It is proposed to only use corners having two edges connected to them, as in figure 4.4. For curved edges it is possible to uniquely relate a point on one edge with a point on the other edge (using an affine invariant parameterization l_1 can be linked to l_2), yielding only one degree of freedom. For straight edges two degrees of freedom are left. Over the parallelogram-shaped region (see figure 4.4) functions that reach their extrema in an invariant way for both geometric and photometric changes, are evaluated. Two possible functions are:

$$\frac{\int I(x,y)dxdy}{\int dxdy} \quad \text{and} \quad \frac{|\mathbf{p} - \mathbf{q} \ \ \mathbf{p} - \mathbf{g}|}{|\mathbf{p} - \mathbf{p}_1 \ \ \mathbf{p} - \mathbf{p}_2|} \tag{4.4}$$

with $I(x, y)$ the image intensity, $\mathbf{g}$ the center of gravity of the region, weighted with image intensity and the other points defined as in figure 4.4.

$$\mathbf{g} = \left(\frac{\int I(x,y) \, x \, dxdy}{\int I(x,y) \, dxdy}, \frac{\int I(x,y) \, y \, dxdy}{\int I(x,y) \, dxdy} \right) \tag{4.5}$$

The regions for which such an extremum is reached will thus also be determined invariantly. In practice it turns out that the extrema are often not very well defined when two degrees of freedom are left (i.e. for straight edges), but occur in shallow "valleys". In these cases more than one function is used at the same time and intersections of these "valleys" are used to determine invariant regions.

Now that we have a method at hand for the automatic extraction of local, affinely invariant image regions, these can easily be described in an affinely invariant way using moment invariants [166]. As in the

region finding steps, invariance both under affine geometric changes and linear photometric changes, with different offsets and different scale factors for each of the three color bands, is considered.

For each region, a feature vector of moment invariance is composed. These can be compared quite efficiently with the invariant vectors computed for other regions, using a hashing-technique. It can be interesting to take the region type (curved or straight edges? Extrema of which function?) into account as well. Once the corresponding regions have been identified, the cross-correlation between them (after normalization to a square reference region) is computed as a final check to reject false matches.

4.2 Two view geometry computation

As was seen in Section 3.2.1, even for an arbitrary geometric structure, the projections of points in two views contains some structure. Finding back this structure is not only very interesting since it is equivalent to the projective calibration of the camera for the two views, but also allows to simplify the search for more matches since these have to satisfy the epipolar constraint. As will be seen further it also allows to eliminate most of the outliers from the matches.

4.2.1 Eight-point algorithm

The two view structure is equivalent to the fundamental matrix. Since the fundamental matrix $\mathbf{F}$ is a 3×3 matrix determined up to an arbitrary scale factor, 8 equations are required to obtain a unique solution. The simplest way to compute the fundamental matrix consists of using Equation (3.26). This equation can be rewritten under the following form:

$$\begin{bmatrix} xx' & yx' & x' & xy' & yy' & y' & x & y & 1 \end{bmatrix} \mathbf{f} = 0 \tag{4.6}$$

with $\mathbf{m} = [x\,y\,1]^\top, \mathbf{m}' = [x'\,y'\,1]^\top$ and $\mathbf{f} = [F_{11}\,F_{12}\,F_{13}\,F_{21}\,F_{22}\,F_{23}\,F_{31}\,F_{32}\,F_{33}]^\top$ a vector containing the elements of the fundamental matrix $\mathbf{F}$. By stacking eight of these equations in a matrix $\mathbf{A}$ the following equation is obtained:

$$\mathbf{A}\mathbf{f} = 0 \tag{4.7}$$

This system of equation is easily solved by Singular Value Decomposition (SVD) [43]. Applying SVD to $\mathbf{A}$ yields the decomposition $\mathbf{U}\mathbf{S}\mathbf{V}^\top$ with $\mathbf{U}$ and $\mathbf{V}$ orthonormal matrices and $\mathbf{S}$ a diagonal matrix containing the singular values. These singular values σ_i are positive and in decreasing order. Therefore in our case σ_9 is guaranteed to be identically zero (8 equations for 9 unknowns) and thus the last column of $\mathbf{V}$ is the correct solution (at least as long as the eight equations are linearly independent, which is equivalent to all other singular values being non-zero).

It is trivial to reconstruct the fundamental matrix $\mathbf{F}$ from the solution vector $\mathbf{f}$. However, in the presence of noise, this matrix will not satisfy the rank-2 constraint. This means that there will not be real epipoles through which all epipolar lines pass, but that these will be "smeared out" to a small region. A solution to this problem is to obtain $\mathbf{F}$ as the closest rank-2 approximation of the solution coming out of the linear equations.

4.2.2 Seven-point algorithm

In fact the two view structure (or the fundamental matrix) only has seven degrees of freedom. If one is prepared to solve non-linear equations, seven points must thus be sufficient to solve for it. In this case the rank-2 constraint must be enforced during the computations.

A similar approach as in the previous section can be followed to characterize the right null-space of the system of linear equations originating from the seven point correspondences. This space can be parameterized as follows $\mathbf{v}_1 + \lambda\mathbf{v}_2$ or $\mathbf{F}_1 + \lambda\mathbf{F}_2$ with $\mathbf{v}_1$ and $\mathbf{v}_2$ being the two last columns of $\mathbf{V}$ (obtained through SVD) and $\mathbf{F}_1$ respectively $\mathbf{F}_2$ the corresponding matrices. The rank-2 constraint is then written as

$$\det(\mathbf{F}_1 + \lambda\mathbf{F}_2) = a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0 \tag{4.8}$$

which is a polynomial of degree 3 in λ. This can simply be solved analytically. There are always 1 or 3 real solutions. The special case $\mathbf{F}_1$ (which is not covered by this parameterization) is easily checked separately, i.e. it should have rank-2. If more than one solution is obtained then more points are needed to obtain the true fundamental matrix.

4.2.3   More points...

It is clear that when more point matches are available the redundancy should be used to minimize the effect of the noise. The eight-point algorithm can easily be extended to be used with more points. In this case the matrix $\mathbf{A}$ of equation 4.7 will be much bigger, it will have one row per point match. The solution can be obtained in the same way, but in this case the last singular value will not be perfectly equal to zero. It has been pointed out [52] that in practice it is very important to normalize the equations. This is for example achieved by transforming the image to the interval $[-1, 1] \times [-1, 1]$ so that all elements of the matrix $\mathbf{A}$ are of the same order of magnitude.

Even then the error that is minimized is an algebraic error which has nor real "physical" meaning. It is always better to minimize a geometrically meaningful criterion. The error measure that immediately comes to mind is the distance between the points and the epipolar lines. Assuming that the noise on every feature point is independent zero-mean Gaussian with the same sigma for all points, the minimization of the following criterion yields a maximum likelihood solution:

$$\mathcal{C}(F) = \sum \left(D(\mathbf{m}', \mathbf{F}\mathbf{m})^2 + D(\mathbf{m}, \mathbf{F}^\top \mathbf{m}')^2 \right) \tag{4.9}$$

with $D(\mathbf{m}, \mathbf{l})$ the orthogonal distance between the point $\mathbf{m}$ and the line $\mathbf{l}$. This criterion can be minimized through a Levenberg-Marquard algorithm [122]. The results obtained through linear least-squares can be used for initialization.

4.2.4   Robust algorithm

The most important problem with the previous approaches is that they can not cope with outliers. If the set of matches is contaminated with even a small set of outliers, the result will probably be unusable. This is typical for all types of least-squares approaches (even non-linear ones). The problem is that the quadratic penalty (which is optimal for Gaussian noise) allows a single outlier being very far apart from the true solution to completely bias the final result.

The problem is that it is very hard to segment the set of matches in inliers and outliers before having the correct solution. The outliers could have such a disastrous effect on the least-square computations that almost no points would be classified as inliers (see Torr [154] for a more in depth discussion of this problem).

A solution to this problem was proposed by Fischler and Bolles [35] (see also [127] for more details on robust statistics). Their algorithm is called RANSAC (RANdom SAmpling Consensus) and it can be applied to all kinds of problems.

Let us take a subset of the data and compute a solution from it. If were are lucky and there are no outliers in our set, the solution will be correct and we will be able to correctly segment the data in inliers and outliers. Of course, we can not rely on being lucky. However, by repeating this procedure with randomly selected subsets, in the end we should end up with the correct solution. The correct solution is identified as the solution with the largest support (i.e. having the largest number of inliers).

Matches are considered inliers if they are not more than 1.96σ pixels away from their epipolar lines, with σ characterizing the amount of noise on the position of the features. In practice σ is hard to estimate and one could just set it to 0.5 or 1 pixel, for example.

The remaining question is of course 'how many samples should be taken?'. Ideally one could try out every possible subset, but this is usually computationally infeasible, so one takes the number of samples m sufficiently high to give a probability Γ in excess of 95% that a good subsample was selected. The expression for this probability is [127]

$$\Gamma = 1 - (1 - (1 - \epsilon)^p)^m \, , \tag{4.10}$$

5%	10%	20%	30%	40%	50%	60%	70%	80%
3	5	13	35	106	382	1827	13692	233963

Table 4.2: The number of 7-point samples required to ensure $\Gamma \geq 0.95$ for a given fraction of outliers.

Step 1. Extract features

Step 2. Compute a set of potential matches

Step 3. While $\Gamma(\#inliers, \#samples) < 95\%$ do

      step 3.1 select minimal sample (7 matches)

      step 3.2 compute solutions for F

      step 3.3 determine inliers

step 4. Refine F based on all inliers

step 5. Look for additional matches

step 6. Refine F based on all correct matches

Table 4.3: Overview of the two-view geometry computation algorithm.

where ϵ is the fraction of outliers, and p the number of features in each sample. In the case of the fundamental matrix $p = 7$. Table 4.2 gives the required number of samples for a few values of ϵ. The algorithm can easily deal with up to 50% outliers, above this the required number of samples becomes very high.

One approach is to decide a priori which level of outlier contamination the algorithm should deal with and set the number of samples accordingly (e.g. coping with up to 50% outliers implies 382 samples).

Often a lower percentage of outliers is present in the data and the correct solution will already have been found after much fewer samples. Assume that sample 57 yields a solution with 60% of consistent matches, in this case one could decide to stop at sample 106, being sure -at least for 95%- not to have missed any bigger set of inliers.

Once the set of matches has been correctly segmented in inliers and outliers, the solution can be refined using all the inliers. The procedure of Section 4.2.3 can be used for this. Table 4.3 summarizes the robust approach to the determination of the two-view geometry. Once the epipolar geometry has been computed it can be used to guide the matching towards additional matches. At this point only features being in epipolar correspondence should be considered for matching. For a corner in one image, only the corners of the other image that are within a small region (1 or 2 pixels) around the corresponding epipolar line, are considered for matching. At this point the initial coordinate interval that was used for matching can be relaxed. By reducing the number of potential matches, the ambiguity is reduced and a number of additional matches are obtained. These can not only be used to refine the solution even further, but will be very useful further on in solving the structure from motion problem where it is important that tracked features survive as long as possible in the sequence.

4.2.5 Degenerate case

The computation of the two-view geometry requires that the matches originate from a 3D scene and that the motion is more than a pure rotation. If the observed scene is planar, the fundamental matrix is only determined up to three degrees of freedom. The same is true when the camera motion is a pure rotation.

In this last case -only having one center of projection- depth can not be observed. In the absence of noise the detection of these degenerate cases would not be too hard. Starting from real -and thus noisy- data, the problem is much harder since the remaining degrees of freedom in the equations are then determined by noise.

A solution to this problem has been proposed by Torr et al. [155]. The methods will try to fit different models to the data and the one explaining the data best will be selected. The approach is based on an extension of Akaike's information criterion [1] proposed by Kanatani [64]. It is outside the scope of this text to describe this method into details. Therefore only the key idea will briefly be sketched here.

Different models are evaluated. In this case the fundamental matrix (corresponding to a 3D scene and more than a pure rotation), a general homography (corresponding to a planar scene) and a rotation-induced homography are computed. Selecting the model with the smallest residual would always yield the most general model. Akaike's principle consist of taking into account the effect of the additional degrees of freedom (which when not needed by the structure of the data end up fitting the noise) on the *expected residual*. This boils down to adding a penalty to the observed residuals in function of the number of degrees of freedom of the model. This makes a fair comparison between the different models feasible.

4.3   Three and four view geometry computation

It is possible to determine the three or four view geometry in a similar way to the two view geometry computation explained in the previous section. More details on these concepts can be found in Section 3.2. Since the points satisfying the three or four view geometry certainly must satisfy the two view geometry, it is often interesting to have a hierarchical approach. In this case the two view geometry is estimated first from consecutive views. Then triplet matches are inferred by comparing two consecutive sets of pair-matches. These triplets are then used in a robust approach similar to the method presented in Section 4.2.4. In this case only 6 triplets of points are needed. A similar approach is possible for the four view geometry.

The method to recover structure and motion presented in the next chapter only relies on the two view geometry. Therefore the interested reader is referred to the literature for more details on the direct computation of three and four view geometric relations. Many authors studied different approaches to compute multi view relations (e.g. [135, 51]). Torr and Zisserman [151] have proposed a robust approach to the computation of the three view geometry. Hartley [54] proposed a method to compute the four view geometry.

Chapter 5

Structure and motion

In the previous section it was seen how different views could be related to each other. In this section the relation between the views and the correspondences between the features will be used to retrieve the structure of the scene and the motion of the camera. This problem is called *Structure and Motion*.

The approach that is proposed here extends [7, 67] by being fully projective and therefore not dependent on the quasi-euclidean initialization. This was achieved by carrying out all measurements in the images. This approach provides an alternative for the triplet-based approach proposed in [36]. An image-based measure that is able to obtain a qualitative distance between viewpoints is also proposed to support initialization and determination of close views (independently of the actual projective frame).

At first two images are selected and an initial reconstruction frame is set-up. Then the pose of the camera for the other views is determined in this frame and each time the initial reconstruction is refined and extended. In this way the pose estimation of views that have no common features with the reference views also becomes possible. Typically, a view is only matched with its predecessor in the sequence. In most cases this works fine, but in some cases (e.g. when the camera moves back and forth) it can be interesting to also relate a new view to a number of additional views. Once the structure and motion has been determined for the whole sequence, the results can be refined through a projective bundle adjustment. Then the ambiguity will be restricted to metric through self-calibration. Finally, a metric bundle adjustment is carried out to obtain an optimal estimation of the structure and motion.

5.1   Initial structure and motion

The first step consists of selecting two views that are suited for initializing the sequential structure and motion computation. On the one hand it is important that sufficient features are matched between these views, on the other hand the views should not be too close to each other so that the initial structure is well-conditioned. The first of these criterions is easy to verify, the second one is harder in the uncalibrated case. The image-based distance that we propose is the median distance between points transferred through an average planar-homography and the corresponding points in the target image:

$$\text{median}\{D(\mathbf{H}\mathbf{m}_i, \mathbf{m}'_i)\} \tag{5.1}$$

This planar-homography $\mathbf{H}$ is determined as follows from the matches between the two views:

$$\mathbf{H} = [\mathbf{e}]_\times \mathbf{F} + \mathbf{e}\mathbf{a}_{min}^\top \text{ with } \mathbf{a}_{min} = \text{argmin}_\mathbf{a} \sum_i D(([\mathbf{e}]_\times \mathbf{F} + \mathbf{e}\mathbf{a}^\top)\mathbf{m}_i, \mathbf{m}'_i)^2 \tag{5.2}$$

In practice the selection of the initial frame can be done by maximizing the product of the number of matches and the image-based distance defined above. When features are matched between sparse views, the evaluation can be restricted to consecutive frames. However, when features are tracked over a video sequence, it is important to consider views that are further apart in the sequence.

5.1.1   Initial frame

Two images of the sequence are used to determine a reference frame. The world frame is aligned with the first camera. The second camera is chosen so that the epipolar geometry corresponds to the retrieved $\mathbf{F}_{12}$:

$$\begin{aligned}
\mathbf{P}_1 &= \begin{bmatrix} \mathbf{I}_{3\times3} & | & \mathbf{0}_3 \end{bmatrix} \\
\mathbf{P}_2 &= \begin{bmatrix} [\mathbf{e}_{12}]_\times \mathbf{F}_{12} + \mathbf{e}_{12}\mathbf{a}^\top & | & \sigma\mathbf{e}_{12} \end{bmatrix}
\end{aligned} \tag{5.3}$$

Equation 5.3 is not completely determined by the epipolar geometry (i.e. $\mathbf{F}_{12}$ and $\mathbf{e}_{12}$), but has 4 more degrees of freedom (i.e. $\mathbf{a}$ and σ). $\mathbf{a}$ determines the position of the reference plane (i.e. the plane at infinity in an affine or metric frame) and σ determines the global scale of the reconstruction. The parameter σ can simply be put to one or alternatively the baseline between the two initial views can be scaled to one. In [7] it was proposed to set the coefficient of $\mathbf{a}$ to ensure a quasi-Euclidean frame, to avoid too large projective distortions. This was needed because not all parts of the algorithms where strictly projective. For the structure and motion approach proposed in this paper $\mathbf{a}$ can be arbitrarily set, e.g. $\mathbf{a} = \begin{bmatrix} 0\ 0\ 0 \end{bmatrix}^\top$.

5.1.2   Initializing structure

Once two projection matrices have been fully determined the matches can be reconstructed through triangulation. Due to noise the lines of sight will not intersect perfectly. In the uncalibrated case the minimizations should be carried out in the images and not in projective 3D space. Therefore, the distance between the reprojected 3D point and the image points should be minimized:

$$D(\mathbf{m}_1, \mathbf{P}_1\mathbf{M})^2 + D(\mathbf{m}_2, \mathbf{P}_2\mathbf{M})^2 \tag{5.4}$$

It was noted by Hartley and Sturm [53] that the only important choice is to select in which epipolar plane the point is reconstructed. Once this choice is made it is trivial to select the optimal point from the plane. A bundle of epipolar planes has only one parameter. In this case the dimension of the problem is reduced from 3-dimensions to 1-dimension. Minimizing the following equation is thus equivalent to minimizing equation (5.4).

$$D(\mathbf{m}_1, \mathbf{l}_1(\alpha))^2 + D(\mathbf{m}_2, \mathbf{l}_2(\alpha))^2 \tag{5.5}$$

with $\mathbf{l}_1(\alpha)$ and $\mathbf{l}_2(\alpha)$ the epipolar lines obtained in function of the parameter α describing the bundle of epipolar planes. It turns out (see [53]) that this equation is a polynomial of degree 6 in α. The global minimum of equation (5.5) can thus easily be computed. In both images the point on the epipolar line $\mathbf{l}_1(\alpha)$ and $\mathbf{l}_2(\alpha)$ closest to the points $\mathbf{m}_1$ resp. $\mathbf{m}_2$ is selected. Since these points are in epipolar correspondence their lines of sight meet in a 3D point.

5.2   Updating the structure and motion

The previous section dealt with obtaining an initial reconstruction from two views. This section discusses how to add a view to an existing reconstruction. First the pose of the camera is determined, then the structure is updated based on the added view and finally new points are initialized.

5.2.1   projective pose estimation

For every additional view the pose towards the pre-existing reconstruction is determined, then the reconstruction is updated. This is illustrated in Figure 5.1. The first step consists of finding the epipolar geometry as described in Section 4.2. Then the matches which correspond to already reconstructed points are used to infer correspondences between 2D and 3D. Based on these the projection matrix $\mathbf{P}_k$ is computed using a robust procedure similar to the one laid out in Table 4.3. In this case a minimal sample of 6 matches is needed to compute $\mathbf{P}_k$. A point is considered an inlier if it is possible to reconstruct a 3D point for which the maximal reprojection error for all views (including the new view) is below a preset threshold. Once $\mathbf{P}_k$ has been determined the projection of already reconstructed points can be predicted. This allows to find some additional matches to refine the estimation of $\mathbf{P}_k$. This means that the search space is gradually reduced from the full image to the epipolar line to the predicted projection of the point.
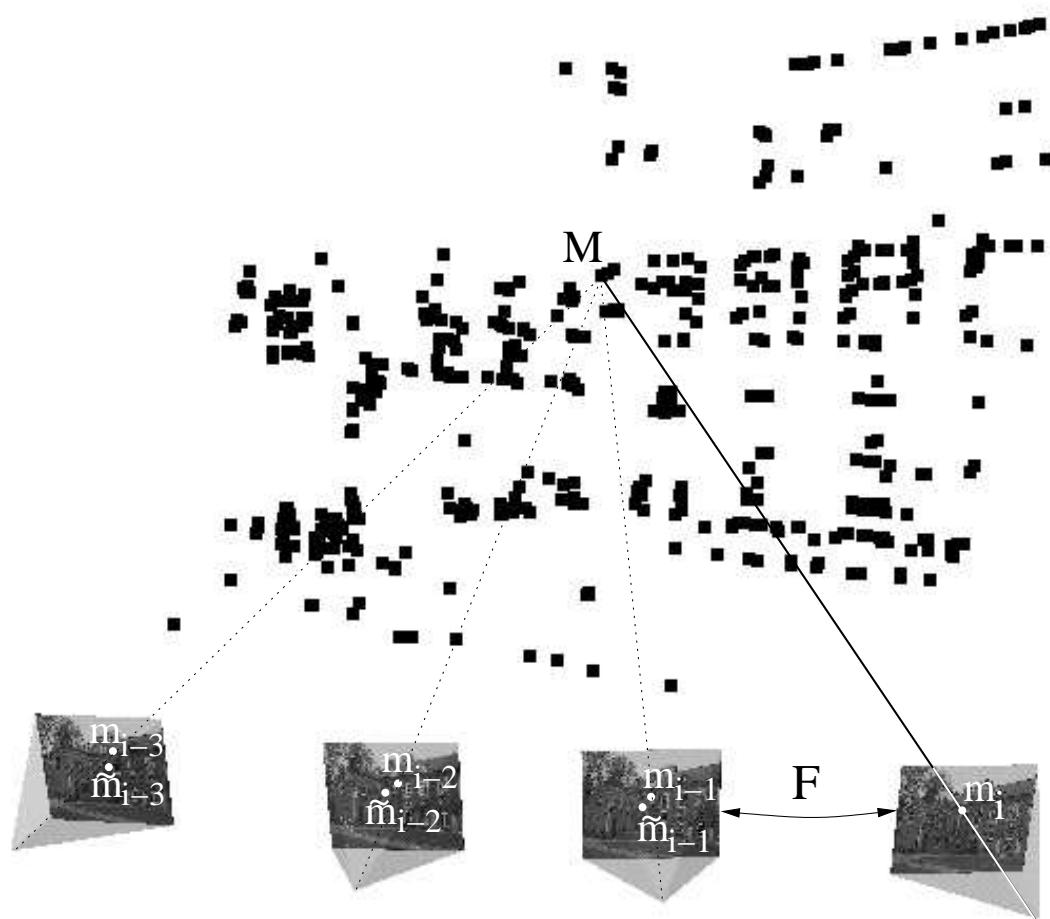
Figure 5.1: Image matches ($\mathtt{m}_{i-1}$, $\mathtt{m}_i$) are found as described before. Since the image points, $\mathtt{m}_{i-1}$, relate to object points, $\mathtt{M}_i$, the pose for view i can be computed from the inferred matches ($\mathtt{M}$, $\mathtt{m}_i$). A point is accepted as an inlier if its line of sight projects sufficiently close to all corresponding points.
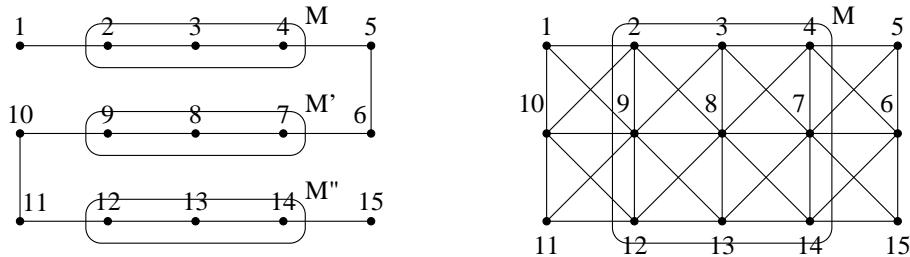
Figure 5.2: *Sequential approach (left) and extended approach (right). In the traditional scheme view 8 would be matched with view 7 and 9 only. A point* M *which would be visible in views 2,3,4,7,8,9,12,13 and 14 would therefore result in 3 independently reconstructed points. With the extended approach only one point will be instantiated. It is clear that this results in a higher accuracy for the reconstructed point while it also dramatically reduces the accumulation of calibration errors.*

5.2.2   Relating to other views

The procedure proposed in th eprevious section only relates the image to the previous image. In fact it is implicitly assumed that once a point gets out of sight, it will not come back. Although this is true for many sequences, this assumptions does not always hold. Assume that a specific 3D point got out of sight, but that it is visible again in the last two views. In this case a new 3D point will be instantiated. This will not immediately cause problems, but since these two 3D points are unrelated for the system, nothing enforces their position to correspond. For longer sequences where the camera is moved back and forth over the scene, this can lead to poor results due to accumulated errors. The problem is illustrated in Figure 5.2

The solution that we propose is to match all the views that are close with the actual view (as described in Section 4.2). For every close view a set of potential 2D-3D correspondences is obtained. These sets are merged and the camera projection matrix is estimated using the same robust procedure as described above, but on the merged set of 2D-3D correspondences.

Close views are determined as follows. First a planar-homography that explains best the image-motion of feature points between the actual and the previous view is determined (using Equation 5.2). Then, the median residual for the transfer of these features to other views using homographies corresponding to the same plane are computed (see Equation 5.1). Since the direction of the camera motion is given through the epipoles, it is possible to limit the selection to the closest views in each direction. In this case it is better to take orientation into account [47, 76] to differentiate between opposite directions.

Example

Figure 5.3 shows one of the images of the *sphere* sequence and the recovered camera calibration together with the tracked points. This calibration can then be used to generate a plenoptic representation from the recorded images (see Section 8.2). Figure 5.4 shows all the images in which each 3D point is tracked. The points are in the order that they were instantiated. This explains the upper triangular structure. It is clear that for the sequential approach, even if some points can be tracked as far as 30 images, most are only seen in a few consecutive images. From the results for the extended approach several things can be noticed. The proposed method is clearly effective in the recovery of points which were not seen in the last images, thereby avoiding unnecessary instantiations of new points (the system only instantiated 2170 points instead of 3792 points). The band structure of the appearance matrix for the sequential approach has been replaced by a dense upper diagonal structure. Some points which were seen in the first images are still seen in the last one (more than 60 images further down the sequence). The mesh structure in the upper triangular part reflects the periodicity in the motion during acquisition. On the average, a point is tracked over 9.1 images instead of 4.8 images with the standard approach. Comparison with ground-truth data shows that the calibration accuracy was improved from 2.31% of the mean object distance to 1.41% by extending the standard structure and motion technique by scanning the viewpoint surface as described in this section.
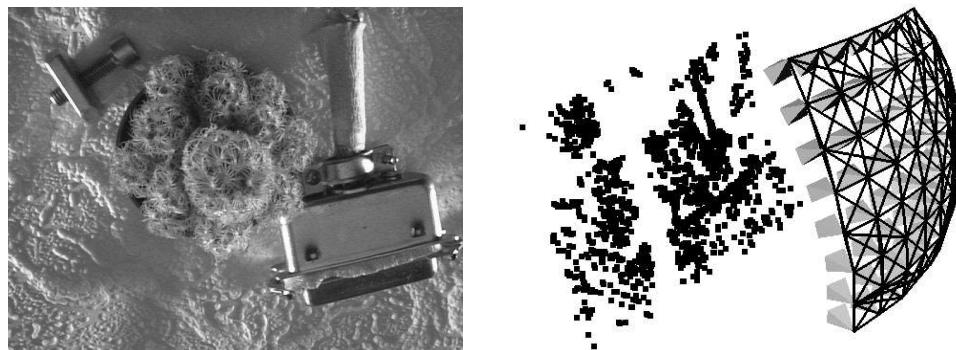
Figure 5.3: Image of the *sphere* sequence (left) and result of calibration step (right). The cameras are represented by little pyramids. Images which were matched together are connected with a black line.

Figure 5.4: Statistics of the *sphere* sequence. This figure indicates in which images a 3D point is seen. Points (vertical) versus images (horizontal). The results are illustrated for both the sequential approach (left) as the extended approach (right) are illustrated.

5.2.3   Refining and extending structure

The structure is refined using an iterated linear reconstruction algorithm on each point. Equation 3.8 can be rewritten to become linear in M:

$$\begin{array}{rcl} \mathtt{P_3 M}x - \mathtt{P_1 M} & = & 0 \\ \mathtt{P_3 M}y - \mathtt{P_2 M} & = & 0 \end{array} \tag{5.6}$$

with $\mathtt{P}_i$ the i-th row of $\mathbf{P}$ and (x, y) being the image coordinates of the point. An estimate of M is co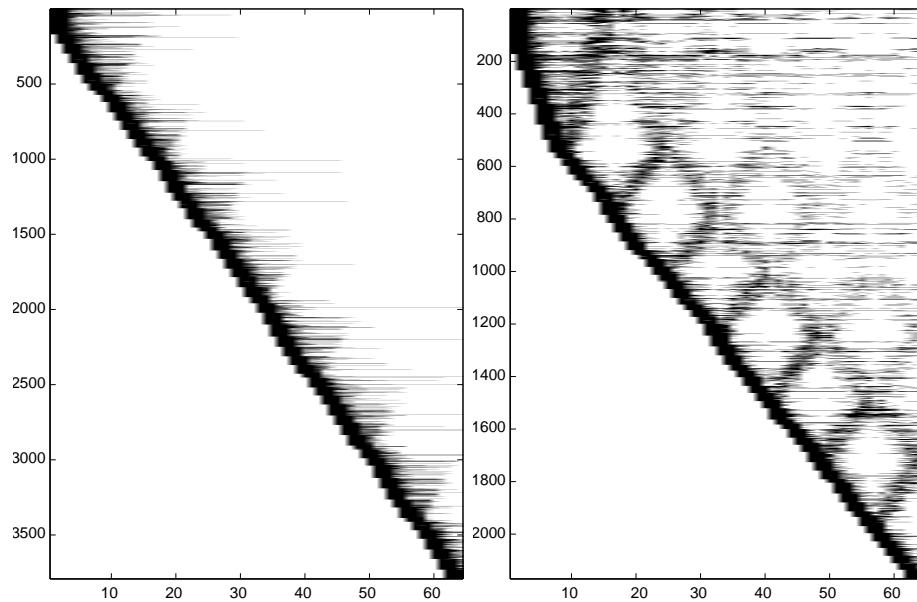mputed by solving the system of linear equations obtained from all views where a corresponding image point is available. To obtain a better solution the criterion $\sum D(\mathbf{P}\mathtt{M}, \mathtt{m})$ should be minimized. This can be approximately obtained by iteratively solving the following weighted linear equations (in matrix form):

$$\frac{1}{\mathtt{P_3}\tilde{\mathtt{M}}} \left[\begin{array}{c} \mathtt{P_3}x - \mathtt{P_1} \\ \mathtt{P_3}y - \mathtt{P_2} \end{array} \right] \mathtt{M} = 0 \tag{5.7}$$

where $\tilde{\mathtt{M}}$ is the previous solution for M. This procedure can be repeated a few times. By solving this system of equations through SVD a normalized homogeneous point is automatically obtained. If a 3D point is not observed the position is not updated. In this case one can check if the point was seen in a sufficient number of views to be kept in the final reconstruction. This minimum number of views can for example be put to three. This avoids to have an important number of outliers due to spurious matches.

Of course in an image sequence some new features will appear in every new image. If point matches are available that were not related to an existing point in the structure, then a new point can be initialized as in section 5.1.2.

After this procedure has been repeated for all the images, one disposes of camera poses for all the views and the reconstruction of the interest points. In the further modules mainly the camera calibration is used. The reconstruction itself is used to obtain an estimate of the disparity range for the dense stereo matching.

5.3   Refining structure and motion

Once the structure and motion has been obtained for the whole sequence, it is recommended to refine it through a global minimization step. A maximum likelihood estimation can be obtained through *bundle adjustment* [158, 137]. The goal is to find the parameters of the camera view $\mathbf{P}_k$ and the 3D points $\mathtt{M}_i$ for which the mean squared distances between the observed image points $\mathtt{m}_{ki}$ and the reprojected image points $\mathbf{P}_k(\mathtt{M}_i)$ is minimized. The camera projection model should also take radial distortion into account. For m views and n points the following criterion should be minimized:

$$\min_{\mathbf{P}_k, \mathtt{M}_i} \sum_{k=1}^{m} \sum_{i=1}^{n} D(\mathtt{m}_{ki}, \mathbf{P}_k(\mathtt{M}_i))^2 \tag{5.8}$$

If the image error is zero-mean Gaussian then bundle adjustment is the Maximum Likelihood Estimator. Although it can be expressed very simply, this minimization problem is huge. For a typical sequence of 20 views and 2000 points, a minimization problem in more than 6000 variables has to be solved. A straight-forward computation is obviously not feasible. However, the special structure of the problem can be exploited to solve the problem much more efficiently. More details on this approach is given in Appendix A.

To conclude this section an overview of the algorithm to retrieve structure and motion from a sequence of images is given. Two views are selected and a projective frame is initialized. The matched corners are reconstructed to obtain an initial structure. The other views in the sequence are related to the existing structure by matching them with their predecessor. Once this is done the structure is updated. Existing points are refined and new points are initialized. When the camera motion implies that points continuously disappear and reappear it is interesting to relate an image to other close views. Once the structure and motion has been retrieved for the whole sequence, the results can be refined through bundle adjustment. The whole procedure is resumed in Table 5.1.

Step 1. Match or track points over the whole image sequence.

Step 2. Initialize the structure and motion recovery

    step 2.1. Select two views that are suited for initialization.

    step 2.2. Relate these views by computing the two view geometry.

    step 2.3. Set up the initial frame.

    step 2.4. Reconstruct the initial structure.

Step 3. For every additional view

    step 3.1. Infer matches to the structure and compute the camera pose using a robust algorithm.

    step 3.2. Refine the existing structure.

    step 3.3. (optional) For already computed views which are "close"

        3.4.1. Relate this view with the current view by finding feature matches and computing the two view geometry.

        3.4.2. Infer new matches to the structure based on the computed matches and add these to the list used in step 3.1.

    Refine the pose from all the matches using a robust algorithm.

    step 3.5. Initialize new structure points.

Step 4. Refine the structure and motion through bundle adjustment.

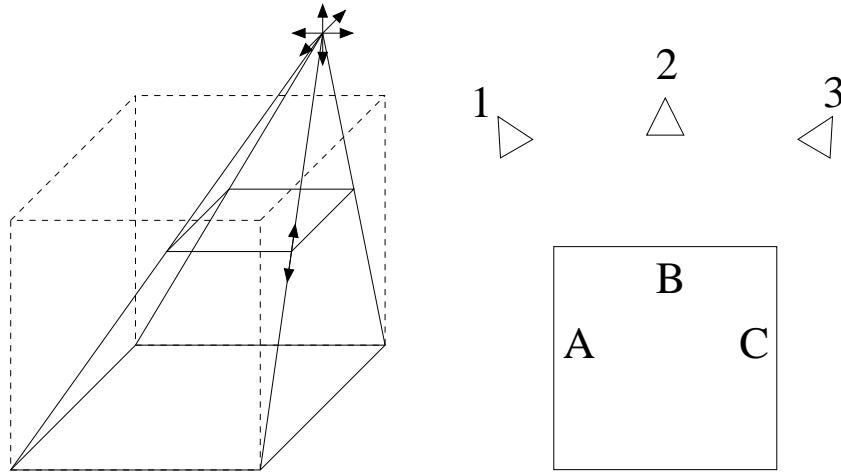Table 5.1: Overview of the projective structure and motion algorithm.

Figure 5.5: Left: Illustration of the four-parameter ambiguity between two projective reconstructions sharing a common plane. If the base of the cube is shared, a projective transformation can still affect the height of the cube and the position of the third vanishing point. Right: A fundamental problem for many (man-made) scenes is that it is not possible to see A,B and C at the same time and therefore when moving from position 1 to position 3 the planar ambiguity problem will be encountered.

5.4   Dealing with dominant planes

The projective structure and motion approach described in the previous section assumes that both motion and structure are general. When this is not the case, the approach can fail. In the case of motion this will happen when the camera is purely rotating. A solution to this problem was proposed in [155]. Here we will assume that care is taken during acquisition to not take multiple images from the same position so that this problem doesn't occur[1].

Scene related problems occur when (part of) the scene is purely planar. In this case it is not possible anymore to determine the epipolar geometry uniquely. If the scene is planar, the image motion can be fully described by a homography. Since $\mathbf{F} = [\mathbf{e}']_\times \mathbf{H}$ (with $[\mathbf{e}']_\times$ the vector product with the epipole $\mathbf{e}'$), there is a 2 parameter family of solutions for the epipolar geometry. In practice robust techniques would pick a random solution based on the inclusion of some outliers.

Assuming we would be able to detect this degeneracy, the problem is not completely solved yet. Obviously, the different subsequences containing sufficient general 3D structure could be reconstructed separately. The structure of subsequences containing only a single plane could also be reconstructed as such. These planar reconstructions could then be inserted into the neighboring 3D projective reconstructions. However, there remains an ambiguity on the transformation relating two 3D projective reconstruction only sharing a common plane. The plane shared by the two reconstructions can be uniquely parameterized by three 3D points (3×3 parameters) and a fourth point in the plane (2 free parameters) to determine the projective basis within the plane. The ambiguity therefore has 15-11=4 degrees of freedom. An illustration is given on the left side of Figure 5.5. Note also that it can be very hard to avoid this type of degeneracy as can be seen from the right side of Figure 5.5. Many scenes have a configuration similar to this one.

In [100] a practical solution was proposed for recovering the 3D structure and camera motion of these types of sequences without ambiguities. This approach is described in the following sections.

5.4.1   Detecting dominant planes

The first part of the solution consists of detecting the cases where only planar features are being matched. The Geometric Robust Information Criterion (GRIC) model selection approach proposed in [150] is briefly reviewed. The GRIC selects the model with the lowest score. The score of a model is obtained by summing

---

[1]Note that the approach would still work if the pure rotation takes place while observing a planar part.
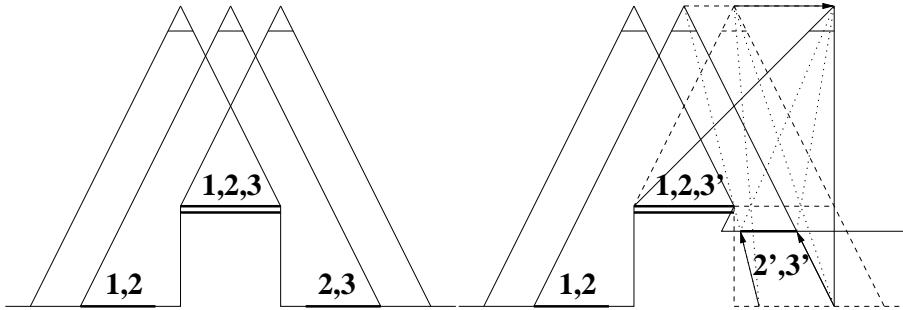
Figure 5.6: Left: Although each pair contains non-coplanar features, the three views only have coplanar points in common. Right: Illustration of the remaining ambiguity if the position of the center of projection for view 2 corresponds for structure 1–2 and 2–3.

two contributions. The first one is related to the goodness of the fit and the second one is related to the parsimony of the model. It is important that a robust Maximum Likelihood Estimator (MLE) be used for estimating the different structure and motion models being compared through GRIC. GRIC takes into account the number n of inliers plus outliers, the residuals e_i, the standard deviation of the measurement error σ, the dimension of the data r, the number k of motion model parameters and the dimension d of the structure:

$$\text{GRIC} = \sum \rho(e_i^2) + (nd\ln(r) + k\ln(rn)) \ . \tag{5.9}$$

where $\rho(e^2)$

$$\rho(e^2) = \min\left(\frac{e^2}{\sigma^2}, 2(r-d)\right) \ . \tag{5.10}$$

In the above equation $nd\ln(r)$ represents the penalty term for the structure having n times d parameters each estimated from r observations and $k\ln(rn)$ represents the penalty term for the motion model having k parameters estimated from rn observations.

For each image pair GRIC($\mathbf{F}$) and GRIC($\mathbf{H}$) can be compared. If GRIC($\mathbf{H}$) yields the lowest value it is assumed that most matched features are located on a dominant plane and that a homography model is therefore appropriate. On the contrary, when GRIC($\mathbf{F}$) yields the lowest value one could assume, as did Torr [155], that standard projective structure and motion recovery could be continued. In most cases this is correct, however, in some cases this might still fail. An illustration of the problem is given on the left side of Figure 5.6 where both $\mathbf{F}_{12}$ and $\mathbf{F}_{23}$ could be successfully computed, but where structure and motion recovery would fail because all features common to the three views are located on a plane. Estimating the pose of camera 3 from features reconstructed from views 1 and 2 or alternatively estimating the trifocal tensor from the triplets would yield a three-parameter family of solutions. However, imposing reconstruction 1–2 and reconstruction 2–3 to be aligned (including the center of projection for view 2) would reduce the ambiguity to a one-parameter family of solutions. This ambiguity is illustrated on the right side of Figure 5.6. Compared to the reference frame of cameras 1 and 2 the position of camera 3 can change arbitrarily as long as the epipole in image 2 is not modified (i.e. motion along a line connecting the center of projections of image 2 and 3). Since intersection has to be preserved and the image of the common plane also has to be invariant, the transformation of the rest of space is completely determined. Note –as seen in Figure 5.6– that this remaining ambiguity could still cause an important distortion.

For the reason described above we propose to use the GRIC criterion on triplets of views ($r = 6$). On the one hand we have GRIC($\mathbf{PPP}$) based on a model containing 3 projection matrices (up to a projective ambiguity) with $k = 3 \times 11 - 15 = 18$ and $d = 3$ (note that using a model based on the trifocal tensor would be equivalent), on the other hand we have GRIC($\mathbf{HH}$) based on a model containing 2 homographies with $k = 2 \times 8 = 16$ and $d = 2$. To efficiently compute the MLE of both $\mathbf{PPP}$ and $\mathbf{HH}$ the sparse structure of the problem is exploited (similar to bundle adjustment). We can now differentiate between two different cases: Case A: GRIC($\mathbf{PPP}$)<GRIC($\mathbf{HH}$): three views observe general 3D structure. Case B: GRIC($\mathbf{PPP}$)>GRIC($\mathbf{HH}$): common structure between three views is planar. Note that it does not make

case	AABAABBBBBAAA
3D	PPPP PPPPP
2D	HH HHHHHH
3D	PPPP
	FF<u>FFF</u>FHHHHFFFF

Table 5.2: Example on how a sequence would be partitioned based on the different cases obtained in the model selection step. Underlined <u>F</u> correspond to cases that would not be dealt with appropriately using a pairwise analysis.

sense to consider mixed cases such as **HF** or **FH** since for structure and motion recovery triplets are needed which in these cases would all be located on a plane anyway.

Note that in addition, one should verify that a sufficient number of triplets remain (say more than 50) to allow a reliable estimation. When too few points are seen in common over three views, the sequence is also split up. In a later stage it can be reassembled (using the procedure laid out in Section 5.4.3). This avoids the risk of a (slight) change of projective basis due to an unreliable estimation based on too few points. Note that it is important to avoid this, since this would mean that different transformations would be required to bring the different parts of the recovered structure and motion back to a metric reference frame. In practice this causes self-calibration to fail and should therefore be avoided.

5.4.2   Partial projective structure and motion recovery

The sequence is first traversed and separated in subsequences. For subsequences with sufficient 3D structure (case A) the approach described in Section 5 is followed so that the projective structure and motion is recovered. When a triplet corresponds to case B, only planar features are tracked and reconstructed (in 2D). A possible partitioning of an image sequence is given in Table 5.2. Note that the triplet 3-4-5 would cause an approach based on [155] to fail.

Suppose the plane Π is labeled as a dominant plane from view i based on features tracked in views $(i-1, i, i+1)$. In general, some feature points $\mathtt{M}_\Pi$ located on Π will have been reconstructed in 3D from previous views (e.g. i and $(i-1)$). Therefore, the coefficients of Π can be computed from $\mathtt{M}_\Pi^\top \Pi = 0$. Define $\mathbf{M}_\Pi$ as the right null space of $\Pi^\top$ (4×3 matrix). $\mathbf{M}_\Pi$ represents 3 supporting points for the plane Π and let $\mathbf{m}_{\Pi i} = \mathbf{P}_i \mathbf{M}_\Pi$ be the corresponding image projections. Define the homography $\mathbf{H}_{i\Pi} = \mathbf{m}_{\Pi i}^{-1}$, then the 3D reconstruction of image points located in the plane Π are obtained as follows:

$$\mathtt{M}_i = \mathbf{M}_\Pi \mathbf{H}_{i\Pi} \mathtt{m}_i \tag{5.11}$$

Similarly, a feature $\mathtt{m}_j$ seen in view $j (> i)$ can be reconstructed as:

$$\mathtt{M}_j = \mathbf{M}_\Pi \mathbf{H}_{i\Pi} (\mathbf{H}_{ij}^\Pi)^{-1} \mathtt{m}_j \tag{5.12}$$

where $\mathbf{H}_{ij}^\Pi = \mathbf{H}_{i(i+1)}^\Pi \ldots \mathbf{H}_{(j-1)j}^\Pi$.

5.4.3   Combined metric structure and motion recovery

Using the coupled self-calibration algorithm described in Section 6.3.1 it is possible to recover the metric structure of the different subsequences.

Once the metric structure of the subsequences has been recovered, the pose of the camera can also be determined for the viewpoints observing only planar points. Since the intrinsics have been computed, a standard pose estimation algorithm can be used. We use Grunert's algorithm as described in [44]. To deal with outliers a robust approach was implemented [35].

Finally, it becomes possible to align the structure and motion recovered for the separate subsequences based on common points. Note that these points are all located in a plane and therefore some precautions have to be taken to obtain results using linear equations. However, since 3 points form a basis in a metric

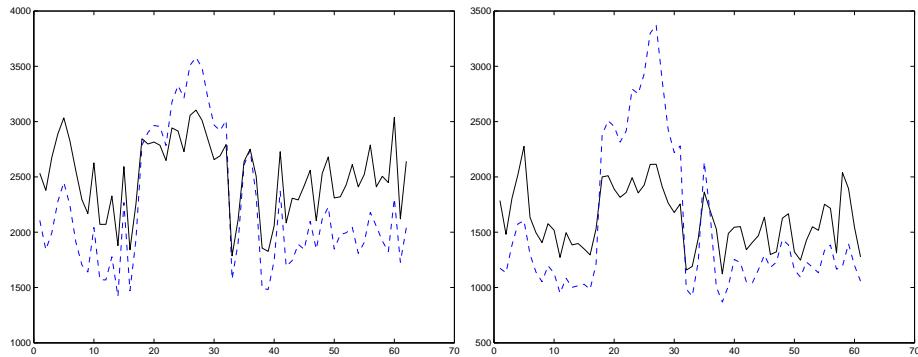Figure 5.7: Some of the 64 images of the *corner* sequence.

Figure 5.8: Left: GRIC(**F**) (solid/black line) and GRIC(**H**) (dashed/blue line). Right: GRIC(**PPP**) (solid/black line) and GRIC(**HH**) (dashed/blue line).

3D space, additional points out of the plane can easily be generated (i.e. using the vector product) and used to compute the relative transform using linear equations. Here again a robust approach is used.

Now that all structure and motion parameters have been estimated for the whole sequence. A final bundle adjustment is carried out to obtain a globally optimal solution.

Example

In this section results of our approach on two real image sequences are shown. The first image sequence was recorded from a corner of our institute. The *corner* sequence contains 64 images recorded using a Sony TRV900 digital camcorder in progressive scan mode. The images therefore have a resolution of 720×576 (PAL). Some of the images are shown in Figure 5.7. Note that the images contain quite some radial distortion.

In Figure 5.8 the GRIC values are given for **F** and **H** as well as for **PPP** and **HH**. It can clearly be seen that –besides dealing with additional ambiguities– the triplet based analysis in general provides more discriminant results. It is also interesting to note that triplet 34-35-36 is clearly indicated as containing sufficiently general structure for the triplet-based approach while the pair-based approach marginally prefers to use the plane based model. The USaM approach reconstructs the structure for this triplet (including some points seen in the background of the lower left picture of Figure 5.7) and successfully integrates them with the rest of the recovered structure and motion. Figure 5.9 shows results for different stages of our approach. At the top-left the recovered metric structure and motion for the two subsequences that contain
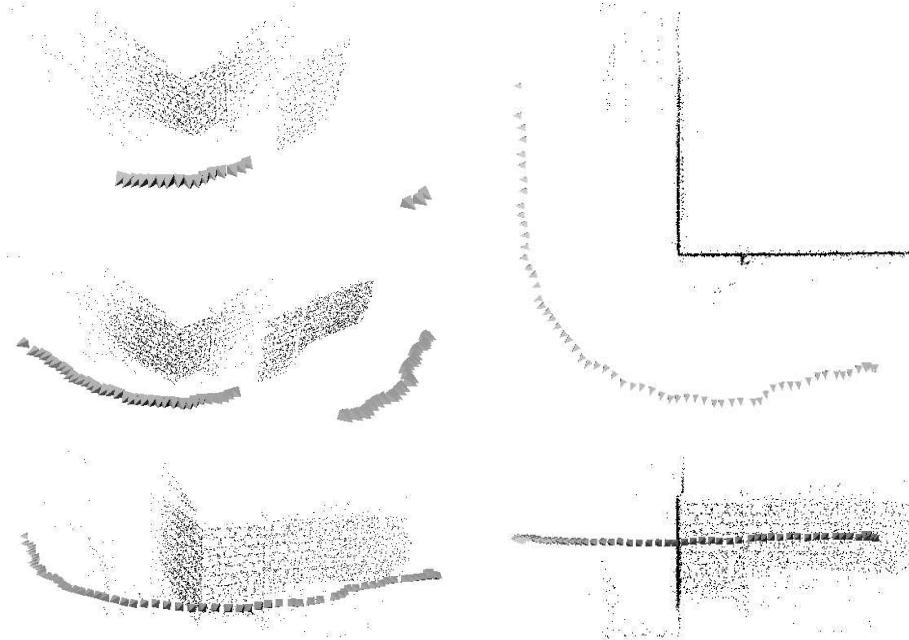
Figure 5.9: Left: different stages of the structure and motion recovery, Right: orthogonal views of the final result.

sufficiently general structure is given (after coupled self-calibration). Then, both structure and motion are extended over the planar parts. This can be seen in the middle-left part of the figure. At the bottom-left the complete structure and motion for the whole sequence is shown after bundle adjustment. On the right side of the figure orthogonal top and front views are shown.

The second sequence consists of 150 images of an old *farmhouse*. It was recorded with the same camera as the first sequence. In Figure 5.10 the GRIC values are plotted and for some of them the corresponding images are shown. As can be seen the approach successfully discriminates between the planar parts and the others. In Figure 5.11 the computed structure and motion is shown. In Figure 5.12 some views of a dense textured 3D model are shown. This model was obtained by computing some depth maps using a stereo algorithm and the obtained metric structure and motion. Note that the whole approach from image sequence to complete 3D model is fully automatic.

5.5   Conclusion

In this section an overview of the algorithm to retrieve structure and motion from a sequence of images is given. First a projective frame is initialized from the two first views. The projective camera matrices are chosen so that they satisfy the computed fundamental matrix. The matched corners are reconstructed so that an initial structure is obtained. The other views in the sequence are related to the existing structure by matching them with their predecessor. Once this is done the structure is updated. Existing points are refined and new points are initialized. When the camera motion implies that points continuously disappear and reappear it is interesting to relate an image to other "close" views. Once the structure and motion has been retrieved for the whole sequence, the results can be refined through bundle adjustment. A solution was also presented for the degenerate case where at some point during the acquisition a dominant plane is observed (i.e. all features common to three consecutive views are located on a plane).
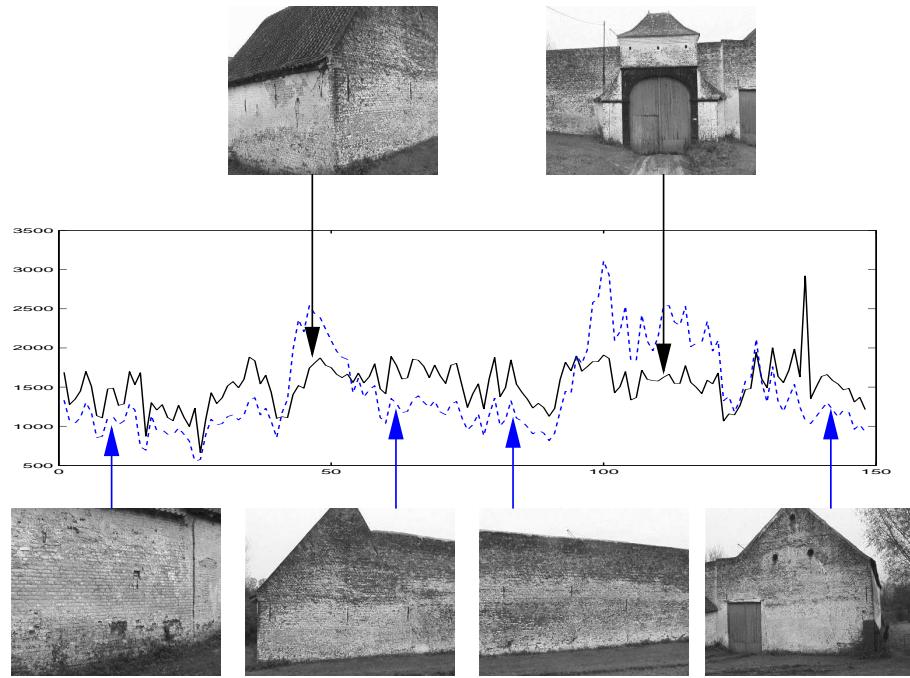
Figure 5.10: Some images of the *farmhouse* sequence together with GRIC(**PPP**) (solid/black line) and GRIC(**HH**) (dashed/blue line).
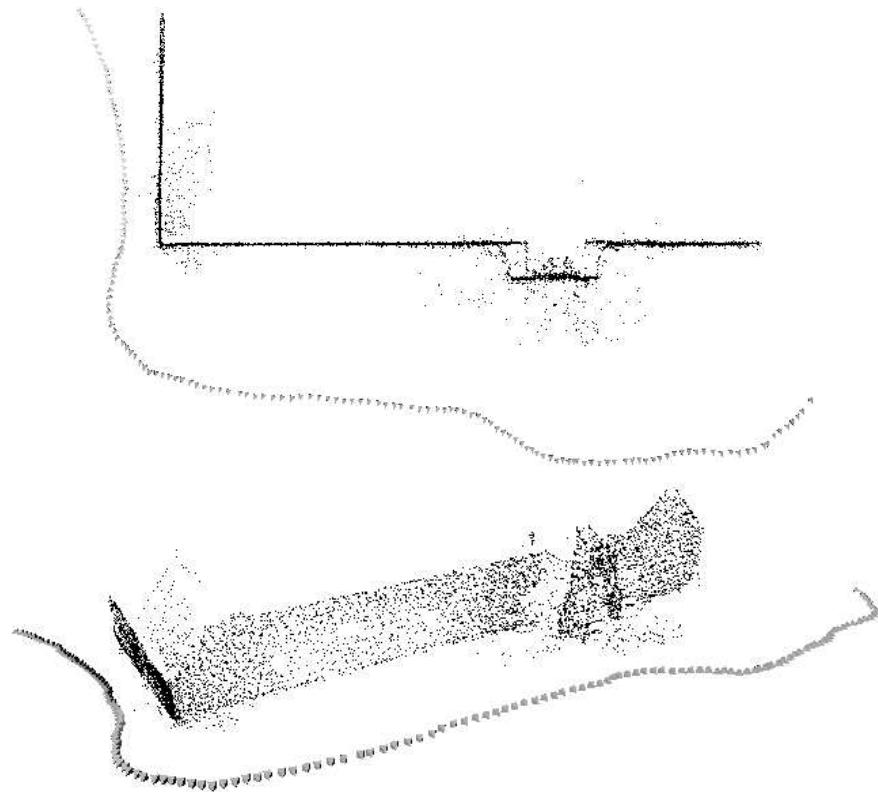

Figure 5.11: Combined structure and motion for the whole *farmhouse* sequence.

Figure 5.12: Textured 3D model of the *farmhouse*

Chapter 6

Self-calibration

The reconstruction obtained as described in the previous chapters is only determined up to an arbitrary projective transformation. This might be sufficient for some robotics or inspection applications, but certainly not for visualization. Therefore we need a method to upgrade the reconstruction to a metric one (i.e. determined up to an arbitrary Euclidean transformation and a scale factor).

In general three types of constraints can be applied to achieve this: scene constraints, camera motion constraints and constraints on the camera intrinsics. All of these have been tried separately or in conjunction. In the case of a hand-held camera and an unknown scene only the last type of constraints can be used. Reducing the ambiguity on the reconstruction by imposing restrictions on the intrinsic camera parameters is termed *self-calibration* (in the area of computer vision). In recent years many researchers have been working on this subject. Mostly self-calibration algorithms are concerned with unknown but constant intrinsic camera parameters (see for example Faugeras et al. [32], Hartley [48], Pollefeys and Van Gool [116, 118, 104], Heyden and Åström [57] and Triggs [157]). Recently, the problem of self-calibration in the case of varying intrinsic camera parameters was also studied (see Pollefeys et al. [115, 105, 99] and Heyden and Åström [58, 60]).

Many researchers proposed specific self-calibration algorithms for restricted motions (i.e. combining camera motion constraints and camera intrinsics constraints). In several cases it turns out that simpler algorithms can be obtained. However, the price to pay is that the ambiguity can often not be restricted to metric. Some interesting approaches were proposed by Moons et al. [89] for pure translation, Hartley [50] for pure rotations and by Armstrong et al. [2] (see also [28]) for planar motion.

Recently some methods were proposed to combine self-calibration with scene constraints. A specific combination was proposed in [117] to resolve a case with minimal information. Bondyfalat and Bougnoux [8] proposed a method of elimination to impose the scene constraints. Liebowitz and Zisserman [79] on the other hand formulate both the scene constraints and the self-calibration constraints as constraints on the absolute conic so that a combined approach is achieved.

Another important aspect of the self-calibration problem is the problem of critical motion sequences. In some cases the motion of the camera is not general enough to allow for self-calibration and an ambiguity remains on the reconstruction. A first complete analysis for constant camera parameters was given by Sturm [144]. Others have also worked on the subject (e.g. Pollefeys [99], Ma et al. [85] and Kahl [62]).

6.1 Calibration

In this section some existing calibration approaches are briefly discussed. These can be based on Euclidean or metric knowledge about the scene, the camera or its motion. One approach consists of first computing a projective reconstruction and then upgrading it a posteriori to a metric (or Euclidean) reconstruction by imposing some constraints. The traditional approaches however immediately go for a metric (or Euclidean) reconstruction.

6.1.1   Scene knowledge

The knowledge of (relative) distances or angles in the scene can be used to obtain information about the metric structure. One of the easiest means to calibrate the scene at a metric level is the knowledge of the relative position of 5 or more points in general position. Assume the points M_l' are the metric coordinates of the projectively reconstructed points M_l, then the transformation $\mathbf{T}$ which upgrades the reconstruction from projective to metric can be obtained from the following equations

$$M_l' \sim \mathbf{T}M_l \text{ or } \lambda_l M_l' = \mathbf{T}M_l \tag{6.1}$$

which can be rewritten as linear equations by eliminating λ_l. Boufama et al. [9] investigated how some Euclidean constraints could be imposed on an uncalibrated reconstruction. The constraints they dealt with are known 3D points, points on a ground plane, vertical alignment and known distances between points. Bondyfalat and Bougnoux [8] recently proposed a method in which the constraints are first processed by a geometric reasoning system so that a minimal representation of the scene is obtained. These constraints can be incidence, parallelism and orthogonality. This minimal representation is then fed to a constrained bundle adjustment.

The traditional approach taken by photogrammetrists [11, 41, 137, 42] consists of immediately imposing the position of known control points during reconstruction. These methods use bundle adjustment [12] which is a global minimization of the reprojection error. This can be expressed through the following criterion:

$$\mathcal{C}_{bundle} = \sum_{i=1}^{n} \sum_{l \in I_i} \left((x_{li} - \mathbf{P}_i(M_l))^2 + (y_{li} - \mathbf{P}_i(M_l))^2 \right) \tag{6.2}$$

where I_i is the set of indices corresponding to the points seen in view i and $\mathbf{P}_i(M_l)$ describes the projection of a point M_l with camera $\mathbf{P}_i$ taking all distortions into account. Note that M_l is known for control points and unknown for other points. It is clear that this approach results in a huge minimization problem and that, even if the special structure of the Jacobian is taken into account (in a similar way as was explained in Section A.2, it is computationally very expensive.

Calibration object   In the case of a calibration object, the parameters of the camera are estimated using an object with known geometry. The known calibration can then be used to obtain immediately metric reconstructions.

Many approaches exist for this type of calibration. Most of these methods consist of a two step procedure where a calibration is obtained first for a simplified (linear) model and then a more complex model, taking distortions into account, is fitted to the measurements. The difference between the methods mainly lies in the type of calibration object that is expected (e.g. planar or not) or the complexity of the camera model that is used. Some existing techniques are Faugeras and Toscani [27], Weng, Cohen and Herniou [170], Tsai [160, 161] (see also the implementation by Willson [173]) and Lenz and Tsai [77].

6.1.2   Camera knowledge

Knowledge about the camera can also be used to restrict the ambiguity on the reconstruction from projective to metric or even beyond. Different parameters of the camera can be known. Both knowledge about the extrinsic parameters (i.e. position and orientation) as the intrinsic parameters can be used for calibration.

Extrinsic parameters   Knowing the relative position of the viewpoints is equivalent to knowing the relative position of 3D points. Therefore the relative position of 5 viewpoints in general position suffices to obtain a metric reconstruction. This is the principle behind the omni-rig [134] recently proposed by Shashua (a similar but more restricted application was described in Pollefeys et al. [120, 119]).

It is less obvious to deal with the orientation parameters, except when the intrinsic parameters are also known (see below).

Intrinsic parameters   If the intrinsic camera parameters are known it is possible to obtain a metric reconstruction. E.g. this calibration can be obtained through off-line calibration with a calibration object. In the minimal case of 2 views and 5 points multiple solutions can exist [33], but in general a unique solution is easily found. Traditional structure from motion algorithms assume known intrinsic parameters and obtain metric reconstructions out of it (e.g. [80, 159, 5, 17, 140, 147]).

Intrinsic and extrinsic parameters   When both intrinsic and extrinsic camera parameters are known, the full camera projection matrix is determined. In this case a Euclidean reconstruction is obtained immediately by back-projecting the points.

In the case of known relative position and orientation of the cameras, the first view can be aligned with the world frame without loss of generality. If only the (relative) orientation and the intrinsic parameters are known, the first 3×3 part of the camera projection matrices is known and it is still possible to linearly obtain the transformation which upgrades the projective reconstruction to metric.

6.2 Self-calibration

In this section some importants concepts for self-calibration are introduced. These are then used to briefly describe some of the existing self-calibration methods.

6.2.1 A counting argument

To restrict the projective ambiguity (15 degrees of freedom) to a metric one (3 degrees of freedom for rotation, 3 for translation and 1 for scale), at least 8 constraints are needed. This thus determines the minimum length of a sequence from which self-calibration can be obtained, depending on the type of constraints which are available for each view. *Knowing* an intrinsic camera parameter for n views gives n constraints, *fixing* one yields only $n-1$ constraints.

$$n \times (\#known) + (n-1) \times (\#fixed) \geq 8$$

Of course this counting argument is only valid if all the constraints are independent. In this context critical motion sequences are of special importance (see Section 6.2.5).

Therefore the absence of skew (1 constraint per view) should in general be enough to allow self-calibration on a sequence of 8 or more images (this was shown in [114, 60, 99]). If in addition the aspect ratio is known (e.g. $f_x = f_y$) then 4 views should be sufficient. When the principal point is known as well a pair of images is sufficient.

6.2.2 Geometric interpretation of constraints

In this section a geometric interpretation of a camera projection matrix is given. It is seen that constraints on the internal camera parameters can easily be given a geometric interpretation in space.

A camera projection plane defines a set of three planes. The first one is parallel to the image and goes through the center of projection. This plane can be obtained by back-projecting the line at infinity of the image (i.e. $[001]^\top$). The two others respectively correspond to the back-projection of the image x- and y-axis (i.e. $[010]^\top$ and $[100]^\top$ resp.). A line can be back-projected through equation (3.9):

$$\Pi \sim \mathbf{P}^\top \mathbf{l} \sim \left[\begin{array}{c} \mathbf{R} \\ \text{-}\mathbf{t}^\top \mathbf{R} \end{array} \right] \mathbf{K}^\top \mathbf{l} \tag{6.3}$$

Let us look at the relative orientation of these planes. Therefore the rotation and translation can be left out without loss of generality (i.e. a camera centered representation is used). Let us then define the vectors $\mathbf{v}_x$, $\mathbf{v}_y$ and $\mathbf{v}_i$ as the first three coefficients of these planes. This then yields the following three vectors:

$$\mathbf{v}_x = \left[\begin{array}{c} 0 \\ f_y \\ c_y \end{array} \right], \ \mathbf{v}_y = \left[\begin{array}{c} f_x \\ s \\ c_x \end{array} \right], \ \mathbf{v}_i = \left[\begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right] \tag{6.4}$$

The vectors coinciding with the direction of the x and the y axis can be obtained by intersections of these planes:

$$\mathtt{l}_x = \mathtt{v}_x \times \mathtt{v}_i = \begin{bmatrix} f_y \\ 0 \\ 0 \end{bmatrix} \text{ and } \mathtt{l}_y = \mathtt{v}_y \times \mathtt{v}_i = \begin{bmatrix} s \\ -f_x \\ 0 \end{bmatrix} \quad . \tag{6.5}$$

The following dot products can now be taken:

$$\mathtt{l}_x.\mathtt{l}_y = s f_y \ , \ \mathtt{v}_x.\mathtt{v}_i = c_y \text{ and } \mathtt{v}_y.\mathtt{v}_i = c_x \tag{6.6}$$

Equation (6.6) proves that the constraint for rectangular pixels (i.e. $s = 0$), and zero coordinates for the principal point (i.e. $c_x = 0$ and $c_y = 0$) can all be expressed in terms of orthogonality between vectors in space. Note further that it is possible to pre-warp the image so that a known skew[1] or known principal point parameters coincide with zero. Similarly a known focal length or aspect ratio can be scaled to one.

The AC is also possible to give a geometric interpretation to a known focal length or aspect ratio. Put a plane parallel with the image at distance d from the center of projection (i.e. $Z = d$ in camera centered coordinates). In this case a horizontal motion in the image of f_x pixels will move the intersection point of the line of sight over a distance d. In other words a known focal length is equivalent to knowing that the length of two (typically orthogonal) vectors are equal. If the aspect ratio is defined as the ratio between the horizontal and vertical sides of a pixel (which makes it independent of s), a similar interpretation is possible.

6.2.3   The image of the absolute conic

One of the most important concepts for self-calibration is the Absolute Conic (AC) and its projection in the images (IAC) [2]. Since it is invariant under Euclidean transformations (see Section 2.2.3), its relative position to a moving camera is constant. For constant intrinsic camera parameters its image will therefore also be constant. This is similar to someone who has the impression that the moon is following him when driving on a straight road. Note that the AC is more general, because it is not only invariant to translations but also to arbitrary rotations.

It can be seen as a calibration object which is naturally present in all the scenes. Once the AC is localized, it can be used to upgrade the reconstruction to metric. It is, however, not always so simple to find the AC in the reconstructed space. In some cases it is not possible to make the difference between the true AC and other candidates. This problem will be discussed in the Section 6.2.5.

In practice the simplest way to represent the AC is through the Dual Absolute Quadric (DAQ). In this case both the AC and its supporting plane, the plane at infinity, are expressed through one geometric entity. The relationship between the AC and the IAC is easily obtained using the projection equation for the DAQ:

$$\omega_i^* \sim \mathbf{P}_i \Omega^* \mathbf{P}_i^\top \quad . \tag{6.7}$$

with ω_i^* representing the dual of the IAC, Ω^* the DAQ and $\mathbf{P}_i$ the projection matrix for view i. Figure 6.1 illustrates these concepts. For a Euclidean representation of the world the camera projection matrices can be factorized as: $\mathbf{P}_i = \mathbf{K}_i \mathbf{R}_i^\top [\mathbf{I} \,|\, \texttt{-}\mathtt{t}_i]$ (with $\mathbf{K}_i$ an upper triangular matrix containing the intrinsic camera parameters, $\mathbf{R}_i^\top$ representing the orientation and $\mathtt{t}_i$ the position) and the DAQ can be written as $\Omega^* = diag(1,1,1,0)$. Substituting this in Equation (6.7), one obtains:

$$\omega_i^* \sim \mathbf{K}_i \mathbf{K}_i^\top \tag{6.8}$$

This equation is very useful because it immediately relates the intrinsic camera parameters to the DIAC.

In the case of a projective representation of the world the DAQ will not be at its standard position, but will have the following form: $\Omega^* = \mathbf{T} \Omega_M^* \mathbf{T}^\top$ with $\mathbf{T}$ being the transformation from the metric to the projective representation. But, since the images were obtained in a Euclidean world, the images ω_i^* still satisfy Equation (6.8). If Ω^* is retrieved, it is possible to upgrade the geometry from projective to metric.

---

[1] In this case the skew should be given as an angle in the image plane. If the aspect ratio is also known, this corresponds to an angle in the retinal plane (e.g. CCD-array).
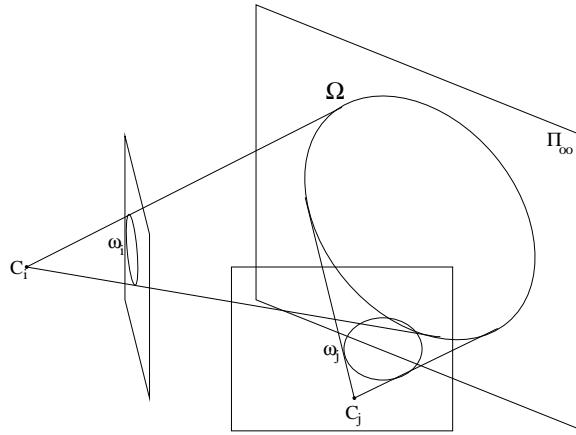
[2] See Section 2.2.3 for details.

Figure 6.1: *The absolute conic (located in the plane at infinity) and its projection in the images*
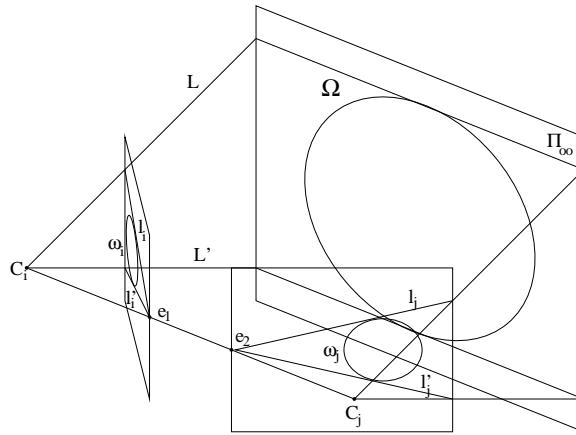

Figure 6.2: *The Kruppa equations impose that the image of the absolute conic satisfies the epipolar constraint. In both images the epipolar lines corresponding to the two planes through C_i and C_j tangent to Ω must be tangent to the images ω_i and ω_j.*

The IAC can also be transferred from one image to another through the homography of its supporting plane (i.e. the plane at infinity):

$$\omega_j \sim \mathbf{H}_{ij}^{\infty -\top} \omega_i \mathbf{H}_{ij}^{\infty -1} \text{ or } \omega_j^* \sim \mathbf{H}_{ij}^{\infty} \omega_i^* \mathbf{H}_{ij}^{\infty \top} \ . \tag{6.9}$$

It is also possible to restrict this constraint to the epipolar geometry. In this case one obtains the Kruppa equations [75] (see Figure 6.2):

$$[\mathbf{e}_{ij}]_\times^\top \mathbf{K}\mathbf{K}^\top [\mathbf{e}_{ij}]_\times \sim \mathbf{F}_{ij} \mathbf{K}\mathbf{K}^\top \mathbf{F}_{ij}^\top \tag{6.10}$$

with $\mathbf{F}_{ij}$ the fundamental matrix for views i and j and $\mathbf{e}_{ij}$ the corresponding epipole. In this case only 2 (instead of 5) independent equations can be obtained [176]. In fact restricting the self-calibration constraints to the epipolar geometry is equivalent to the elimination of the position of infinity from the equations. The result is that some artificial degeneracies are created (see [142]).

6.2.4 Self-calibration methods

In this section some self-calibration approaches are briefly discussed. Combining Equation (6.7) and (6.8) one obtains the following equation:

$$\mathbf{K}_i \mathbf{K}_i^\top \sim \mathbf{P}_i \Omega^* \mathbf{P}_i^\top \tag{6.11}$$

critical motion type	ambiguity
pure translation	affine transformation (5DOF)
pure rotation[3]	arbitrary position for plane at infinity (3DOF)
orbital motion	projective distortion along rotation axis (2DOF)
planar motion	scaling axis perpendicular to plane (1DOF)

Table 6.1: Critical motion sequences for constant intrinsic parameters

Several methods are based on this equation. For constant intrinsic parameters Triggs [157] proposed to minimize the deviation from Equation (6.11). A similar approach was proposed by Heyden and Åström [57]. Pollefeys and Van Gool [118] proposed a related approach based on the transfer equation (i.e. Equation (6.9)) rather than the projection equation. These different approaches are very similar as was shown in [118]. The more flexible self-calibration method which allows varying intrinsic camera parameters [105] is also based on Equation (6.11).

The first self-calibration method was proposed by Faugeras et al. [32] based on the Kruppa equations (Equation (6.10)). The approach was improved over the years [84, 176]. An interesting feature of this self-calibration technique is that no consistent projective reconstruction must be available, only pairwise epipolar calibration. This can be very useful is some cases where it is hard to relate all the images into a single projective frame. The price paid for this advantage is that 3 of the 5 absolute conic transfer equations are used to eliminate the dependence on the position of the plane at infinity. This explains why this method performs poorly compared to others when a consistent projective reconstruction can be obtained (see [104]).

When the homography of the plane at infinity $\mathbf{H}_{ij}^{\infty}$ is known, then Equation (6.9) can be reduced to a set of linear equations in the coefficients of ω_i or ω_i^* (this was proposed by Hartley [48]). Several self-calibration approaches rely on this possibility. Some methods follow a stratified approach and obtain the homographies of the plane at infinity by first reaching an affine calibration, based an a pure translation (see Moons et al. [89]) or using the modulus constraint (see Pollefeys et al. [104]). Other methods are based on pure rotations (see Hartley [50] for constant intrinsic parameters and de Agapito et al. [20] for a zooming camera).

6.2.5   Critical motion sequences

One noticed very soon that not all motion sequences are suited for self-calibration. Some obvious cases are the restricted motions described in the previous section (i.e. pure translation, pure rotation and planar motion). However there are more motion sequences which do not lead to unique solutions for the self-calibration problem. This means that at least two reconstructions are possible which satisfy all constraints on the camera parameters for all the images of the sequence and which are not related by a similarity transformation.

Several researchers realized this problem and mentioned some specific cases or did a partial analysis of the problem [157, 176, 121]. Sturm [144, 145] provided a complete catalogue of critical motion sequences (CMS) for constant intrinsic parameters. Additionally, he identified specific degeneracies for some algorithms [142].
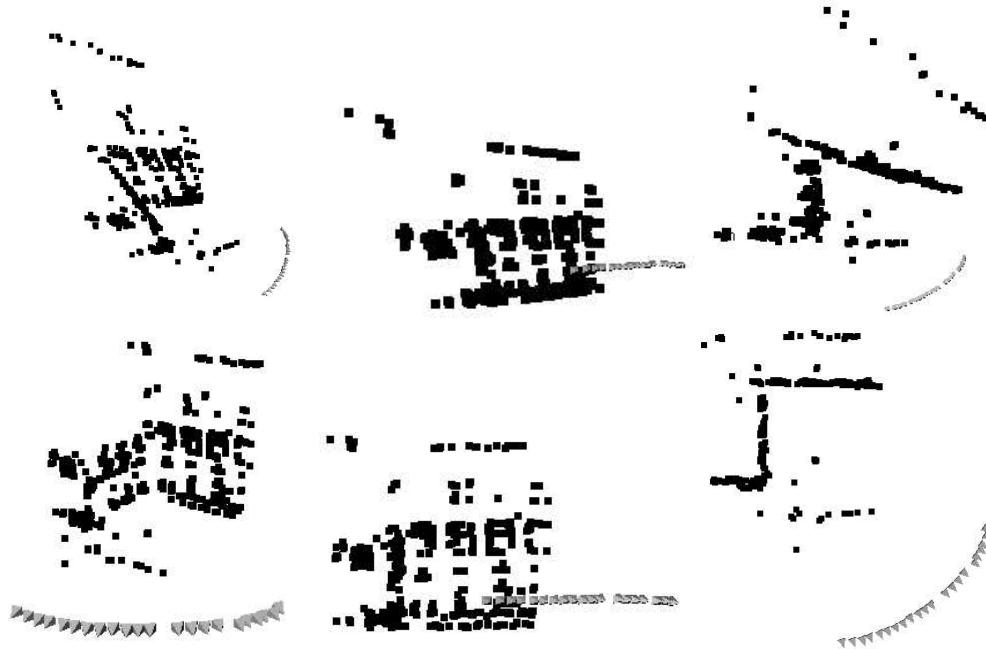
However it is very important to notice that the classes of CMS that exist depend on the constraints that are enforced during self-calibration. The extremes being all parameters known, in which case almost no degeneracies exist, and, no constraints at all, in which case all motion sequences are critical.

In table 6.1 and 6.2 the most important critical motion sequences for self-calibration using the constraint of constant -but unknown- intrinsics respectively intrinsics known up to a freely moving focal length are listed. More details can be found in [99]. For self-calibration to be successful it is important that the global motion over the sequence is general enough so that it is not contained in any of the critical motion sequence classes.

---

[3]In this case even a projective reconstruction is impossible since all the lines of sight of a point coincide.

critical motion type	ambiguity
pure rotation[4]	arbitrary position for plane at infinity (3DOF)
forward motion	projective distortion along optical axis (2DOF)
translation and	scaling optical axis (1DOF)
rotation about optical axis	
hyperbolic and/or elliptic motion	one extra solution

Table 6.2: Critical motion sequences for varying focal length

Figure 6.3: Structure and motion *before* (top) and *after* (bottom) self-calibration.

6.3 A practical approach to self-calibration

In the previous section several self-calibration methods were briefly presented. In this section we will work out a flexible self-calibration approach (this method was proposed in [114], see also [105] or [99]). This method can deal with varying intrinsic camera parameters. This is important since it allows the use of zoom and auto-focus available on most cameras.

The only assumption which is strictly needed by the method is that pixels are rectangular (see for a proof [114, 99]). In practice however it is interesting to make more assumptions. In many cases pixels are square and the principal point is located close to the center of the image. Our systems first uses a linear method to obtain an approximate calibration. This calibration is then refined through a non-linear optimization step in a second phase. The approach that is proposed here is based on [114] but was was modified to better take into account the a priori information on the intrinsic camera parameters, thereby reducing the problem of critical motion sequences.

In Figure 6.3 the retrieved structure and motion is shown before (top) and after (bottom) self-calibration. Note that metric properties such as orthogonality and parallelism can be observed after self-calibration.

6.3.1   linear self-calibration

The first step consists of normalizing the projection matrices. The following normalization is proposed:

$$\mathbf{P}_N = \mathbf{K}_N^{-1}\mathbf{P} \text{ with } \mathbf{K}_N = \begin{bmatrix} w+h & 0 & \frac{w}{2} \\ & w+h & \frac{h}{2} \\ & & 1 \end{bmatrix} \tag{6.12}$$

where w and h are the width, resp. height of the image. After the normalization the focal length should be of the order of unity and the principal point should be close to the origin. The above normalization would scale a focal length of a 60mm lens to 1 and thus focal lengths in the range of 20mm to 180mm would end up in the range $[1/3, 3]$. The aspect ratio is typically also around 1 and the skew can be assumed 0 for all practical purposes. Making these a priori knowledge more explicit and estimating reasonable standard deviations one could for example get $f \approx rf \approx 1 \pm 3$, $u \approx v \approx 0 \pm 0.1$, $r \approx 1 \pm 0.1$ and $s = 0$. It is now interesting to investigate the impact of this knowledge on ω^*:

$$\omega^* \sim \mathbf{KK}^\top = \begin{bmatrix} f^2+s^2+u^2 & srf+uv & u \\ srf+uv & r^2f^2+v^2 & v \\ u & v & 1 \end{bmatrix} \approx \begin{bmatrix} 1\pm9 & \pm0.01 & \pm0.1 \\ \pm0.01 & 1\pm9 & \pm0.1 \\ \pm0.1 & \pm0.1 & 1 \end{bmatrix} \tag{6.13}$$

and $\omega_{22}^*/\omega_{11}^* \approx 1 \pm 0.2$. The constraints on the left-hand side of Equation (6.7) should also be verified on the right-hand side (up to scale). The uncertainty can be take into account by weighting the equations.

$$\begin{aligned}
\frac{1}{9\nu}\left(P_1\Omega^* P_1{}^\top - P_3\Omega^* P_3{}^\top\right) &= 0 \\
\frac{1}{9\nu}\left(P_2\Omega^* P_2{}^\top - P_3\Omega^* P_3{}^\top\right) &= 0 \\
\frac{1}{0.2\nu}\left(P_1\Omega^* P_1{}^\top - P_2\Omega^* P_2{}^\top\right) &= 0 \\
\frac{1}{0.1\nu}\left(P_1\Omega^* P_2{}^\top\right) &= 0 \\
\frac{1}{0.1\nu}\left(P_1\Omega^* P_3{}^\top\right) &= 0 \\
\frac{1}{0.01\nu}\left(P_2\Omega^* P_3{}^\top\right) &= 0
\end{aligned} \tag{6.14}$$

with P_i the ith row of $\mathbf{P}$ and ν a scale factor that is initially set to 1 and later on to $P_3\tilde{\Omega}^* P_3{}^\top$ with $\tilde{\Omega}^*$ the result of the previous iteration. Since Ω^* is a symmetric 4×4 matrix it is parametrized through 10 coefficients. An estimate of the dual absolute quadric Ω^* can be obtained by solving the above set of equations for all views through linear least-squares. The rank-3 constraint should be imposed by forcing the smallest singular value to zero. This scheme can be iterated until the ν factors converge, but in practice this is often not necessary. Experimental validation has shown that this approach yields much better results than the original approach described in [114, 105]. This is mostly due to the fact that constraining all parameters (even with a small weight) allows to avoid most of the problems due to critical motion sequences [144, 63] (especially the specific additional case for the linear algorithm [99]).

coupled self-calibration

In some cases different (sub)sequences have been recorded with the same camera and the same settings. In this case it is possible to combine the equations derived above into a larger system of equations. This can for example be used to solve the problem of a dominant plane separating a sequence in two subsequences (see Section 5.4).

When choosing $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ for one of the projection matrices it can be seen from Equation (6.11) that Ω^* can be written as:

$$\Omega^* = \begin{bmatrix} \mathbf{KK}^\top & \mathbf{a} \\ \mathbf{a}^\top & b \end{bmatrix} \tag{6.15}$$

Now the set of equations (6.14) can thus be written as:

$$\begin{bmatrix} \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{k} \\ \begin{bmatrix} \mathbf{a} \\ b \end{bmatrix} \end{bmatrix} \tag{6.16}$$

where $\mathbf{k}$ is a vector containing 6 coefficients representing the matrix $\mathbf{KK}^\top$, $\mathbf{a}$ is a 3-vector and b a scalar and $\mathbf{C}$ and $\mathbf{D}$ are matrices containing the coefficients of the equations. Note that this can be done independently for every 3D subsequence.

If the sequence is recorded with constant intrinsics, the vector $\mathbf{k}$ will be common to all subsequences and one obtains the following coupled self-calibration equations:

$$
\begin{bmatrix}
\mathbf{C}_1 & \mathbf{D}_1 & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{C}_2 & \mathbf{0} & \mathbf{D}_2 & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\mathbf{C}_n & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}_n
\end{bmatrix}
\begin{bmatrix}
\mathbf{k} \\
\begin{bmatrix} \mathbf{a}_1 \\ b_1 \end{bmatrix} \\
\begin{bmatrix} \mathbf{a}_2 \\ b_2 \end{bmatrix} \\
\vdots \\
\begin{bmatrix} \mathbf{a}_n \\ b_n \end{bmatrix}
\end{bmatrix}
\tag{6.17}
$$

As will be seen in the experiments this approach is very successful. The most important feature is that through the coupling it allows to get good results even for the shorter subsequences. For each subsequence a transformation to upgrade the reconstruction from projective to metric can be obtained from the constraint $\mathbf{T}_i \boldsymbol{\Omega}_i^* \mathbf{T}_i^\top = \mathrm{diag}(1, 1, 1, 0)$ (through Cholesky factorization).

6.3.2 non-linear self-calibration refinement

Before going for a bundle-adjustment it can still be interesting to refine the linear self-calibration results through a minimization that only involves the camera projection matrices (although in practice this is often not necessary since the results of the linear computation are good enough to allow the convergence of the bundle adjustment to the desired solution). Let us define the functions $f(.), r(.), u(.), v(.)$ and $s(.)$ that respectively extract the focal length, aspect ratio, coordinates of the principal point and skew from a projection matrix (in practice this is done based on QR-decomposition). Then our expectations for the distributions of the parameters could be translated to the following criterion (for a projection matrix normalized as in Equation (6.12)):

$$
\mathcal{C}(\mathbf{T}) = \sum_i \left(\frac{\log(f(\mathbf{P}_i\mathbf{T}^{-1}))^2}{\log(3)^2} + \frac{\log(r(\mathbf{P}_i\mathbf{T}^{-1}))^2}{\log(1.1)^2} + \frac{u(\mathbf{P}_i\mathbf{T}^{-1})^2}{0.1^2} + \frac{v(\mathbf{P}_i\mathbf{T}^{-1})^2}{0.1^2} + \frac{s(\mathbf{P}_i\mathbf{T}^{-1})^2}{0.01^2} \right)
\tag{6.18}
$$

Note that since f and r indicate relative and not absolute values, it is more meaningful to use logarithmic values in the minimization. This also naturally avoids that the focal length would collapse to zero for some degenerate cases. In this criterion $\mathbf{T}$ should be parametrized with 8 parameters and initialized with the solution of the linear algorithm. The refined solution for the transformation can then be obtained as:

$$
\mathbf{T}_{opt} = \arg\min \mathcal{C}(\mathbf{T})
\tag{6.19}
$$

Some terms can also be added to enforce constant parameters, e.g. $\frac{(\log(f(\mathbf{P}_i\mathbf{T}^{-1})) - \overline{\log f})^2}{\log(0.1)^2}$ with $\overline{\log f}$ the average logarithm of the observed focal length. The metric structure and motion is then obtained as

$$
\mathbf{P}_M = \mathbf{P}\mathbf{T}^{-1} \text{ and } \mathbf{M}_M = \mathbf{T}\mathbf{M}
\tag{6.20}
$$

This result can then further be refined through bundle adjustment. In this case the constraints on the intrinsics should also be enforced during the minimization process. For more details the reader is referred to [158].

6.3.3 Metric bundle adjustment

For high accuracy the recovered metric structure should be refined using a maximum likelihood approach such as the bundle adjustment (see Appendix A). In this case, however, the metric structure and not the

projective structure is retrieved. This means that the camera projection matrices should be parametrized using intrinsic and extrinsic parameters (and not in homogeneous form as in the projective case). If one assumes that the error is only due to mislocalization of the image features and that this error is uniform and normally distributed[5], the bundle adjustment corresponds to a maximum likelihood estimator. For this to be satisfied the camera model should be general enoguh so that no systematic errors remain in the data (e.g. due to lens distortion). In these circumstances the maximum likelihood estimation corresponds to the solution of a least-squares problem. In this case a criterion of the type of equation (6.2) should be minimized:

$$\mathcal{C}_{ML}(\mathtt{M}_l, \mathbf{K}_i, \mathbf{R}_i, \mathtt{t}_i) = \sum_{i=1}^{n} \sum_{l \in I_i} \left((x_{li} - \frac{\mathtt{P}_{i1}\mathtt{M}_l}{\mathtt{P}_{i3}\mathtt{M}_l})^2 + (y_{li} - \frac{\mathtt{P}_{i2}\mathtt{M}_l}{\mathtt{P}_{i3}\mathtt{M}_l})^2 \right) \tag{6.21}$$

where I_i is the set of indices corresponding to the points seen in view i and $\mathbf{P}_i \equiv \left[\mathtt{P}_{i1}^\top \mathtt{P}_{i2}^\top \mathtt{P}_{i3}^\top \right]^\top = \mathbf{K}_i [\mathbf{R}_i^\top | -\mathbf{R}_i^\top \mathtt{t}_i]$. This criterion should be extended with terms that reflect the (un)certainty on the intrinsic camera parameters. This would yield a criterion of the following form:

$$\mathcal{C}'_{ML}(\mathtt{M}_l, \mathbf{K}_i, \mathbf{R}_i, \mathtt{t}_i) \;=\; \sum_{i=1}^{n} \sum_{l \in I_i} \left((x_{li} - \frac{\mathtt{P}_{i1}\mathtt{M}_l}{\mathtt{P}_{i3}\mathtt{M}_l})^2 + (y_{li} - \frac{\mathtt{P}_{i2}\mathtt{M}_l}{\mathtt{P}_{i3}\mathtt{M}_l})^2 \right) \\ + \sum_{i=1}^{n} \sum_{k=1}^{m} \lambda_k C_{ki}(\mathbf{K}_i)^2 \tag{6.22}$$

with λ_k a regularization factor and $C_{ki}(\mathbf{K}_i)$ representing the constraints on the intrinsic camera parameters, e.g. $C_{1i} = f_{x i} - f_{y_i}$ (known aspect ratio), $C_{2i} = u_{x i}$ (known principal point) or $f_{x i} - f_x$ (constant focal length). The values of the factors λ_k depend on how strongly the constraints should be enforced.

6.4   Conclusion

In this chapter we discussed how to restrict the projective ambiguity of the reconstruction to metric (i.e. Euclidean up to scale). After a brief discussion of traditional calibration approaches, we focussed on the problem of self-calibration. The general concepts were introduced and the most important methods briefly presented. Then a flexible self-calibration approach that can deal with focusing/zooming cameras was worked out in detail.

---

[5]This is a realistic assumption since outliers should have been removed at this stage of the processing.

Chapter 7

Dense depth estimation

With the camera calibration given for all viewpoints of the sequence, we can proceed with methods developed for calibrated structure from motion algorithms. The feature tracking algorithm already delivers a sparse surface model based on distinct feature points. This however is not sufficient to reconstruct geometrically correct and visually pleasing surface models. This task is accomplished by a dense disparity matching that estimates correspondences from the grey level images directly by exploiting additional geometrical constraints.

This chapter is organized as follows. In a first section rectification is discussed. This makes it possible to use standard stereo matching techniques on image pairs. Stereo matching is discussed in a second section. Finally a multi-view approach that allows to integrate the results obtained from several pairs is presented.

7.1 Image pair rectification

The stereo matching problem can be solved much more efficiently if images are rectified. This step consists of transforming the images so that the epipolar lines are aligned horizontally. In this case stereo matching algorithms can easily take advantage of the epipolar constraint and reduce the search space to one dimension (i.e. corresponding rows of the rectified images).

The traditional rectification scheme consists of transforming the image planes so that the corresponding space planes are coinciding [4]. There exist many variants of this traditional approach (e.g. [4, 29, 96, 179]), it was even implemented in hardware [15]. This approach fails when the epipoles are located in the images since this would have to results in infinitely large images. Even when this is not the case the image can still become very large (i.e. if the epipole is close to the image).

Roy et al. [128] proposed a method to avoid this problem, but their approach is relatively complex and shows some problems. Recently Pollefeys et al. [106] proposed a simple method which guarantees minimal image size and works for all possible configuration. This method will be presented in detail further on, but first the standard planar rectification is briefly discussed.

7.1.1 Planar rectification

The standard rectification approach is relatively simple. It consists of selecting a plane parallel with the baseline. The two image are then reprojected into this plane. This is illustrated in Figure 7.1. These new images satisfy the standard stereo setup. The different methods for rectification mainly differ in how the remaining degrees of freedom are chosen. In the calibrated case one can choose the distance from the plane to the baseline so that no pixels are compressed during the warping from the images to the rectified images and the normal on the plane can be chosen in the middle of the two epipolar planes containing the optical axes. In the uncalibrated case the choice is less obvious. Several approaches were proposed (e.g. [29, 179]).
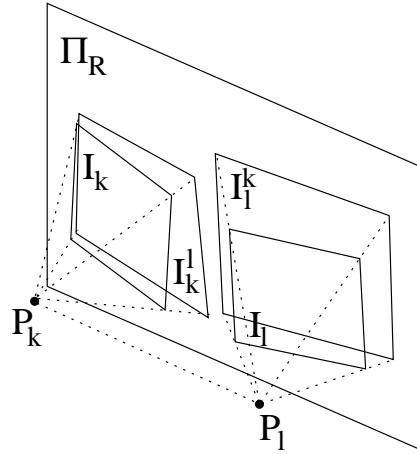
Figure 7.1: Planar rectification: $(\mathbf{I}_k^l, \mathbf{I}_l^k)$ are the rectified images for the pair $(\mathbf{I}_k, \mathbf{I}_l)$ (the plane Π_R should be parallel to the baseline $(\mathbf{P}_k, \mathbf{P}_l)$).

7.1.2   Polar rectification

Here we present a simple algorithm for rectification which can deal with all possible camera geometries. Only the oriented fundamental matrix is required. All transformations are done in the images. The image size is as small as can be achieved without compressing parts of the images. This is achieved by preserving the length of the epipolar lines and by determining the width independently for every half epipolar line.

For traditional stereo applications the limitations of standard rectification algorithms are not so important. The main component of camera displacement is parallel to the images for classical stereo setups. The limited vergence keeps the epipoles far from the images. New approaches in uncalibrated structure and motion as presented in this text however make it possible to retrieve 3D models of scenes acquired with hand-held cameras. In this case forward motion can no longer be excluded. Especially when a street or a similar kind of scene is considered.

Epipolar geometry

The epipolar geometry describes the relations that exist between two images. The epipolar geometry is described by the following equation:

$$\mathbf{m'}^\top \mathbf{F} \mathbf{m} = 0 \tag{7.1}$$

where $\mathbf{m}$ and $\mathbf{m'}$ are homogeneous representations of corresponding image points and $\mathbf{F}$ is the fundamental matrix. This matrix has rank two, the right and left null-space correspond to the epipoles $\mathbf{e}$ and $\mathbf{e'}$ which are common to all epipolar lines. The epipolar line corresponding to a point $\mathbf{m}$ is given by $\mathbf{l'} \sim \mathbf{F}\mathbf{m}$ with $\sim$ meaning equality up to a non-zero scale factor (a strictly positive scale factor when oriented geometry is used, see further).

Epipolar line transfer   The transfer of corresponding epipolar lines is described by the following equations:

$$\mathbf{l'} \sim \mathbf{H}^{-\top} \mathbf{l} \text{ or } \mathbf{l} \sim \mathbf{H}^\top \mathbf{l'} \tag{7.2}$$

with $\mathbf{H}$ a homography for an arbitrary plane. As seen in [83] a valid homography can be obtained immediately from the fundamental matrix:

$$\mathbf{H} = [\mathbf{e'}]_\times \mathbf{F} + \mathbf{e'} \mathbf{a}^\top \tag{7.3}$$

with $\mathbf{a}$ a random vector for which $\det \mathbf{H} \neq 0$ so that $\mathbf{H}$ is invertible. If one disposes of camera projection matrices an alternative homography is easily obtained as:

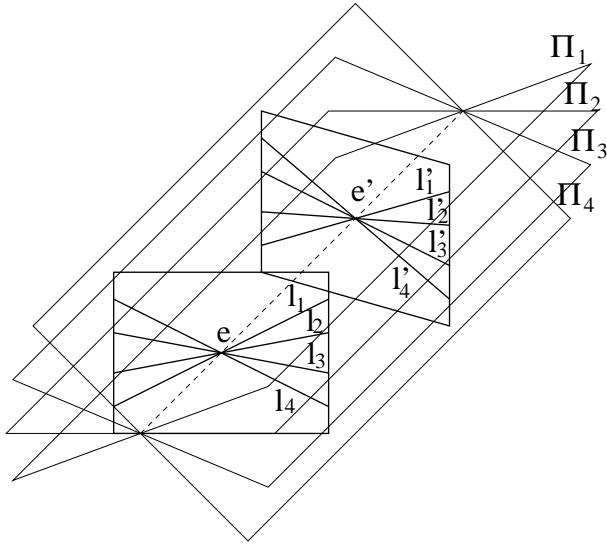$$\mathbf{H}^{-\top} = \left(\mathbf{P'}^\top\right)^\dagger \mathbf{P}^\top \tag{7.4}$$

Figure 7.2: Epipolar geometry with the epipoles in the images. Note that the matching ambiguity is reduced to half epipolar lines.
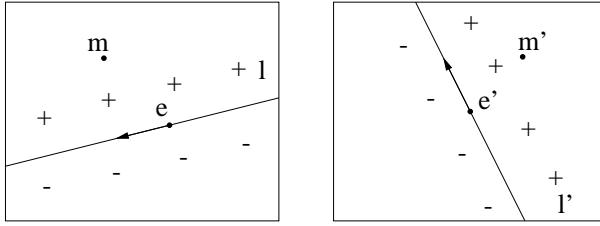

Figure 7.3: Orientation of the epipolar lines.

where † indicates the Moore-Penrose pseudo inverse.

Orienting epipolar lines   The epipolar lines can be oriented such that the matching ambiguity is reduced to half epipolar lines instead of full epipolar lines. This is important when the epipole is in the image. This fact was ignored in the approach of Roy et al. [128].

Figure 7.2 illustrates this concept. Points located in the right halves of the epipolar planes will be projected on the right part of the image planes and depending on the orientation of the image in this plane this will correspond to the right or to the left part of the epipolar lines. These concepts are explained more in detail in the work of Laveau [76] on oriented projective geometry (see also [47]).

In practice this orientation can be obtained as follows. Besides the epipolar geometry one point match is needed (note that 7 or more matches were needed anyway to determine the epipolar geometry). An oriented epipolar line l separates the image plane into a positive and a negative region:

$$f_l(m) = l^\top m \text{ with } m = [x\, y\, 1]^\top \tag{7.5}$$

Note that in this case the ambiguity on l is restricted to a strictly positive scale factor. For a pair of matching points (m, m') both $f_l(m)$ and $f_{l'}(m')$ should have the same sign . Since l' is obtained from l through equation (7.2), this allows to determine the sign of $\mathbf{H}$. Once this sign has been determined the epipolar line transfer is oriented. We take the convention that the positive side of the epipolar line has the positive region of the image to its right. This is clarified in Figure 7.3.
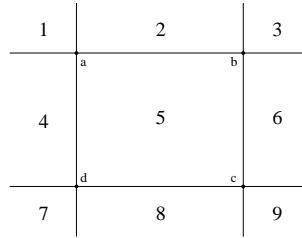
Figure 7.4: the extreme epipolar lines can easily be determined depending on the location of the epipole in one of the 9 regions. The image corners are given by a, b, c, d.
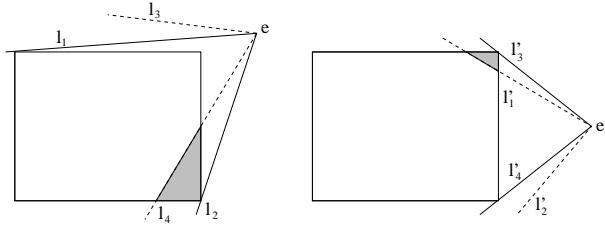

Figure 7.5: Determination of the common region. The extreme epipolar lines are used to determine the maximum angle.

Rectification method

The key idea of our new rectification method consists of reparameterizing the image with polar coordinates (around the epipoles). Since the ambiguity can be reduced to half epipolar lines only positive longitudinal coordinates have to be taken into account. The corresponding half epipolar lines are determined through equation (7.2) taking orientation into account.

The first step consists of determining the common region for both images. Then, starting from one of the extreme epipolar lines, the rectified image is built up line by line. If the epipole is in the image an arbitrary epipolar line can be chosen as starting point. In this case boundary effects can be avoided by adding an overlap of the size of the matching window of the stereo algorithm (i.e. use more than 360 degrees). The distance between consecutive epipolar lines is determined independently for every half epipolar line so that no pixel compression occurs. This non-linear warping allows to obtain the minimal achievable image size without losing image information.

The different steps of this methods are described more in detail in the following paragraphs.

Determining the common region   Before determining the common epipolar lines the extremal epipolar lines for a single image should be determined. These are the epipolar lines that touch the outer image corners. The different regions for the position of the epipole are given in Figure 7.4. The extremal epipolar lines always pass through corners of the image (e.g. if the epipole e is in region 1 the area between eb and ed). The extreme epipolar lines from the second image can be obtained through the same procedure. They should then be transfered to the first image. The common region is then easily determined as in Figure 7.5

Determining the distance between epipolar lines   To avoid losing pixel information the area of every pixel should be at least preserved when transformed to the rectified image. The worst case pixel is always located on the image border opposite to the epipole. A simple procedure to compute this step is depicted in Figure 7.6. The same procedure can be carried out in the other image. In this case the obtained epipolar line should be transferred back to the first image. The minimum of both displacements is carried out.

Constructing the rectified image   The rectified images are built up row by row. Each row corresponds to a certain angular sector. The length along the epipolar line is preserved. Figure 7.7 clarifies these concepts.
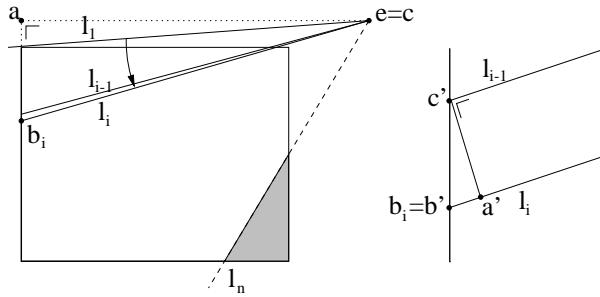
Figure 7.6: Determining the minimum distance between two consecutive epipolar lines. On the left a whole image is shown, on the right a magnification of the area around point b_i is given. To avoid pixel loss the distance $|a'c'|$ should be at least one pixel. This minimal distance is easily obtained by using the congruence of the triangles abc and $a'b'c'$. The new point b is easily obtained from the previous by moving $\frac{|bc|}{|ac|}$ pixels (down in this case).
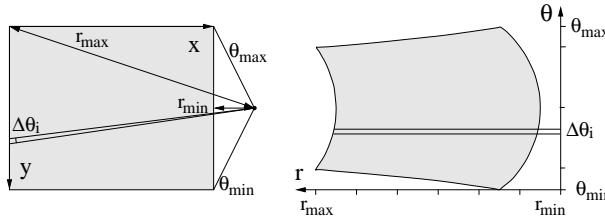

Figure 7.7: The image is transformed from (x,y)-space to (r,θ)-space. Note that the θ-axis is non-uniform so that every epipolar line has an optimal width (this width is determined over the two images).

The coordinates of every epipolar line are saved in a list for later reference (i.e. transformation back to original images). The distance of the first and the last pixels are remembered for every epipolar line. This information allows a simple inverse transformation through the constructed look-up table.

Note that an upper bound for the image size is easily obtained. The height is bound by the contour of the image $2 \times (W + H)$. The width is bound by the diagonal $\sqrt{W^2 + H^2}$. Note that the image size is uniquely determined with our procedure and that it is the minimum that can be achieved without pixel compression.

Transferring information back   Information about a specific point in the original image can be obtained as follows. The information for the corresponding epipolar line can be looked up from the table. The distance to the epipole should be computed and subtracted from the distance for the first pixel of the image row. The image values can easily be interpolated for higher accuracy.

To warp back a complete image a more efficient procedure than a pixel-by-pixel warping can be designed. The image can be reconstructed radially (i.e. radar like). All the pixels between two epipolar lines can then be filled in at once from the information that is available for these epipolar lines. This avoids multiple look-ups in the table. More details on digital image warping can be found in [175].

7.1.3   Examples

As an example a rectified image pair from the Arenberg castle is shown for both the standard rectification and the new approach. Figure 7.8 shows the original image pair and Figure 7.9 shows the rectified image pair for both methods.

A second example shows that the method works properly when the epipole is in the image. Figure 7.10 shows the two original images while Figure 7.11 shows the two rectified images. In this case the standard rectification procedure can not deliver rectified images.

Figure 7.8: Image pair from an Arenberg castle in Leuven scene.

Figure 7.9: Rectified image pair for both methods: standard homography based method (top), new method (bottom).

Figure 7.10: Image pair of the author's desk a few days before a deadline. The epipole is indicated by a white dot (top-right of 'Y' in 'VOLLEYBALL').

A stereo matching algorithm was used on this image pair to compute the disparities. The raw and interpolated disparity maps can be seen in Figure 7.12. Figure 7.13 shows the depth map that was obtained. Note from these images that there is an important depth uncertainty around the epipole. In fact the epipole forms a singularity for the depth estimation. In the depth map of Figure 7.13 an artifact can be seen around the position of the epipole. The extend is much longer in one specific direction due to the matching ambiguity in this direction (see the original image or the middle-right part of the rectified image).

7.2 Stereo matching

Stereo matching is a problem that has been studied over several decades in computer vision and many researchers have worked at solving it. The proposed approaches can be broadly classified into feature- and correlation-based approaches [24]. Some important feature based approaches were proposed by Marr and Poggio [86], Grimson [40], Pollard, Mayhem and Frisby [98] (all relaxation based methods), Gimmel'Farb [38] and Baker and Binford [6] and Ohta and Kanade [94] (using dynamic programming).

Successful correlation based approaches were for example proposed by Okutomi and Kanade [95] or Cox et al.[16]. The latter was recently refined by Koch [69] and Falkenhagen [25, 26]. It is this last algorithm that will be presented in this section. Another approach based on optical flow was proposed by Proesmans et al. [125].

7.2.1 Exploiting scene constraints

The epipolar constraint restricts the search range for a corresponding point m_k in one image to the epipolar line in the other image. It imposes no restrictions on the object geometry other that the reconstructed object point M lays on the line of sight L_k from the projection center of $\mathbf{P}_k$ and through the corresponding point m_k as seen in Figure 7.14(left). The search for the corresponding point m_l is restricted to the epipolar line but no restrictions are imposed along the search line.

If we now think of the epipolar constraint as being a plane spanned by the line of sight L_k and the baseline connecting the camera projection centers, then we will find the epipolar line by intersecting the image plane $\mathbf{I}_l$ with this epipolar plane.

This plane also intersects the image plane $\mathbf{I}_k$ and it cuts a 3D profile out of the surface of the scene objects. The profile projects onto the corresponding epipolar lines in $\mathbf{I}_k$ and $\mathbf{I}_l$ where it forms an ordered set of neighboring correspondences, as indicated in Figure 7.14 (right).

For well behaved surfaces this ordering is preserved and delivers an additional constraint, known as 'ordering constraint'. Scene constraints like this can be applied by making weak assumptions about the object geometry. In many real applications the observed objects will be opaque and composed out of piecewise continuous surfaces. If this restriction holds then additional constraints can be imposed on the correspondence estimation. Koschan[74] listed as many as 12 different constraints for correspondence estimation in stereo pairs. Of them, the most important apart from the epipolar constraint are:

1. Ordering Constraint: For opaque surfaces the order of neighboring correspondences on the corresponding epipolar lines is always preserved. This ordering allows the construction of a dynamic

Figure 7.11: Rectified pair of images of the desk. It can be verified visually that corresponding points are located on corresponding image rows. The right side of the images corresponds to the epipole.

programming scheme which is employed by many dense disparity estimation algorithms [38, 16, 26].

2. Uniqueness Constraint: The correspondence between any two corresponding points is bidirectional as long as there is no occlusion in one of the images. A correspondence vector pointing from an image point to its corresponding point in the other image always has a corresponding reverse vector pointing back. This test is used to detect outliers and occlusions.

3. Disparity Limit: The search band is restricted along the epipolar line because the observed scene has only a limited depth range (see Figure 7.14, right).

4. Disparity continuity constraint: The disparities of the correspondences vary mostly continuously and step edges occur only at surface discontinuities. This constraint relates to the assumption of piecewise continuous surfaces. It provides means to further restrict the search range. For neighboring image pixels along the epipolar line one can even impose an upper bound on the possible disparity change. Disparity changes above the bound indicate a surface discontinuity.

All above mentioned constraints operate along the epipolar lines which may have an arbitrary orientation in the image planes. The matching procedure is greatly simplified if the image pair is rectified to a standard geometry. How this can be achieved for an arbitrary image pair is explained in the Section 7.1.2. In standard geometry both image planes are coplanar and the epipoles are projected to infinity. The rectified image planes are oriented such that the epipolar lines coincide with the image scan lines. This corresponds to a camera translated in the direction of the x-axis of the image. An example is shown in figure 7.15. In this case the image displacements between the two images or *disparities* are purely horizontal.
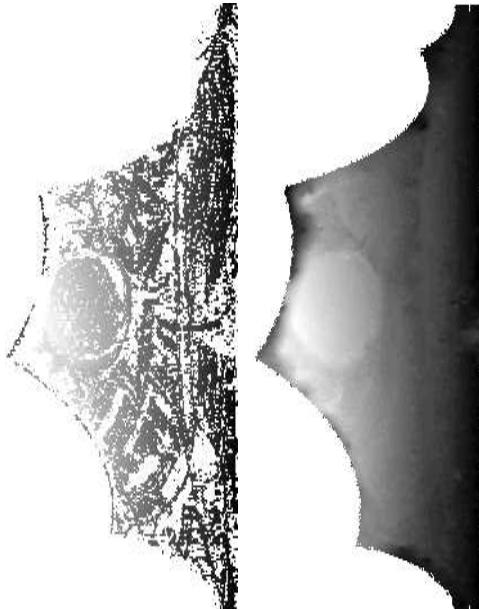
Figure 7.12: Raw and interpolated disparity estimates for the far image of the desk image pair.
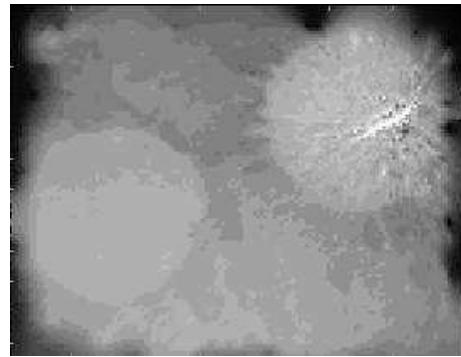

Figure 7.13: Depth map for the far image of the desk image pair.

7.2.2 Constrained matching

For dense correspondence matching a disparity estimator based on the dynamic programming scheme of Cox *et al.* [16], is employed that incorporates the above mentioned constraints. It operates on rectified image pairs where the epipolar lines coincide with image scan lines. The matcher searches at each pixel in image $\mathbf{I}_{\mathbf{k}}^{\mathbf{l}}$ for maximum normalized cross correlation in $\mathbf{I}_{\mathbf{l}}^{\mathbf{k}}$ by shifting a small measurement window (kernel size 5x5 or 7x7) along the corresponding scan line. The selected search step size ΔD (usually 1 pixel) determines the search resolution and the minimum and maximum disparity values determine the search region. This is illustrated in Figure 7.16.

Matching ambiguities are resolved by exploiting the ordering constraint in the dynamic programming approach [69]. The algorithm was further adapted to employ extended neighborhood relationships and a pyramidal estimation scheme to reliably deal with very large disparity ranges of over 50% of the image size [26]. The estimate is stored in a disparity map $D_{(k,l)}$ with one of the following values:

– a valid correspondence $\mathbf{m}_l^k = D_{(k,l)}[\mathbf{m}_k^l]$,
– an undetected search failure which leads to an outlier,
– a detected search failure with no correspondence.

A confidence value is kept together with the correspondence that tells if a correspondence is valid
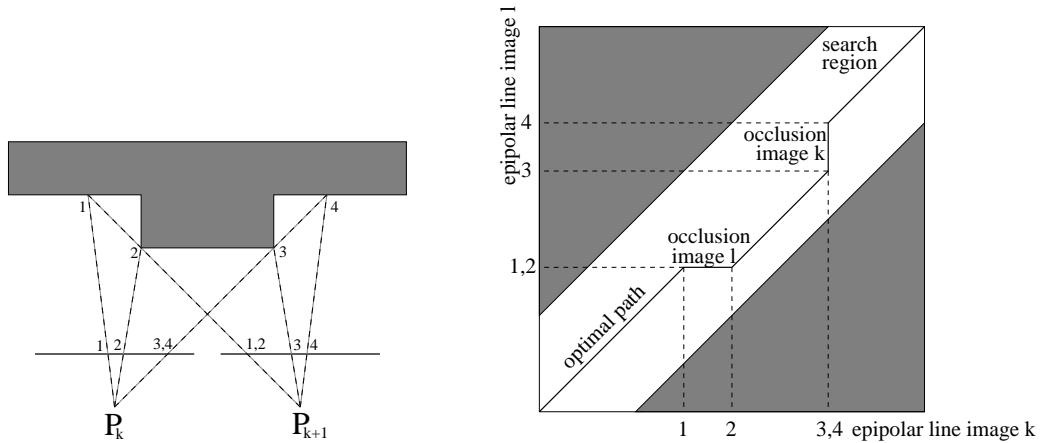
Figure 7.14: Object profile triangulation from ordered neighboring correspondences (left). Rectification and correspondence between viewpoints k and l (right).
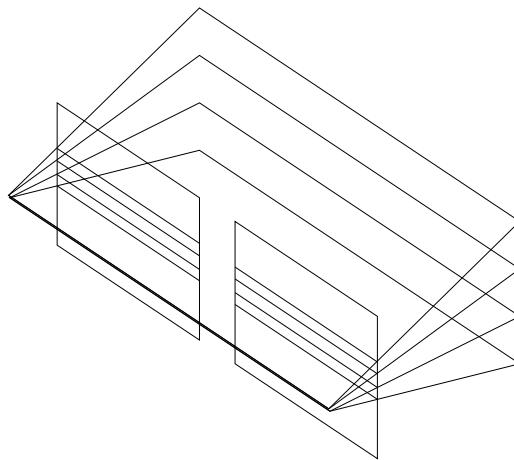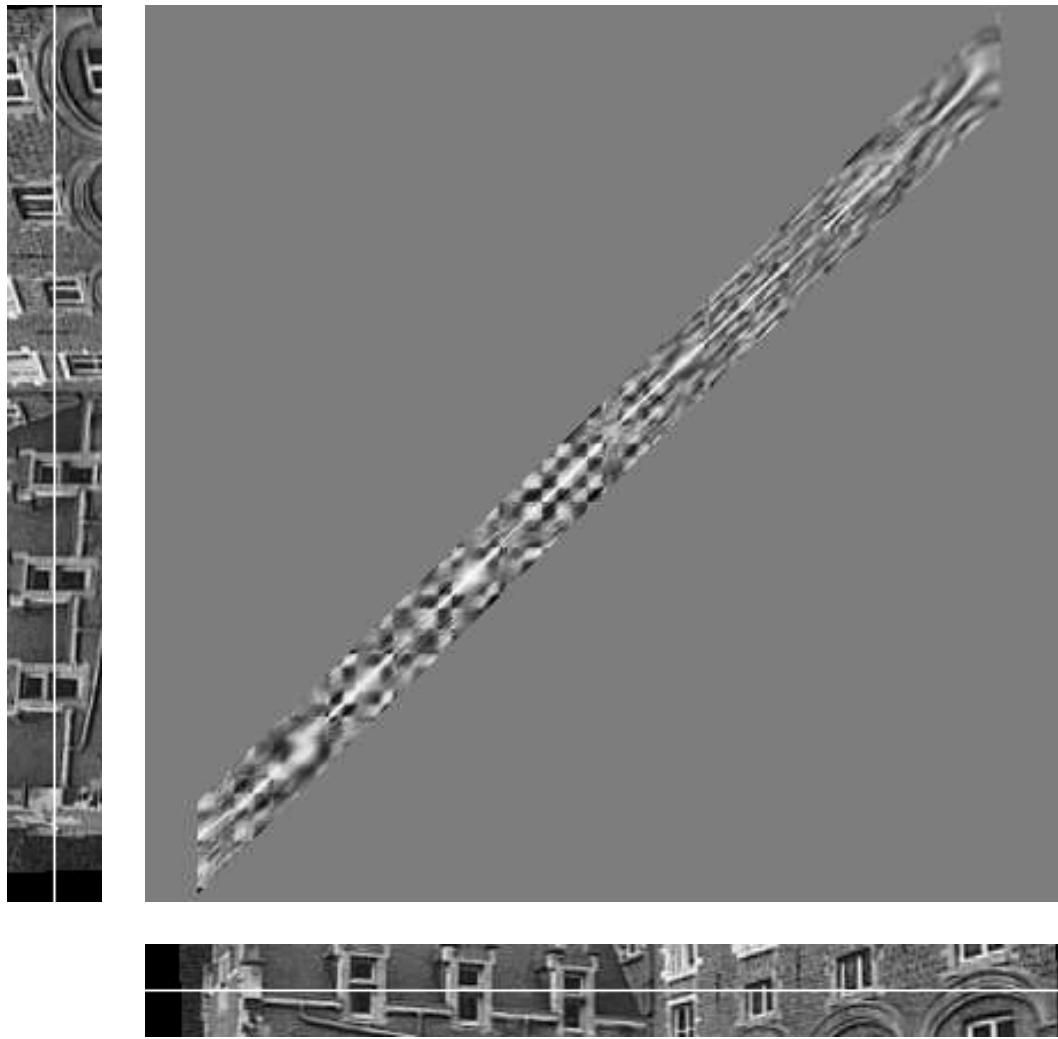

Figure 7.15: Standard stereo setup

Figure 7.16: Cross-correlation for two corresponding epipolar lines (light means high cross-correlation). A dynamic programming approach is used to estimate the optimal path.

and how good it is. The confidence is derived from the local image variance and the maximum cross correlation[73]. To further reduce measurement outliers the uniqueness constraint is employed by estimating correspondences bidirectionally $D(k \rightarrow l), D(l \rightarrow k)$. Only the consistent correspondences with

$|D(k \rightarrow l) - D(l \rightarrow k)| < \Delta D$ are kept as valid correspondences.

7.3   Multi-view stereo

The pairwise disparity estimation allows to compute image to image correspondences between adjacent rectified image pairs, and independent depth estimates for each camera viewpoint. An optimal joint estimate will be achieved by fusing all independent estimates into a common 3D model. The fusion can be performed in an economical way through controlled correspondence linking as described in this section. The approach utilizes a flexible multi-viewpoint scheme by combining the advantages of small baseline and wide baseline stereo.

As **small baseline stereo** we define viewpoints where the baseline is much smaller than the observed average scene depth. This configuration is usually valid for image sequences were the images are taken as a spatial sequence from many slightly varying view-points. The advantages (+) and disadvantages (–) are

+ easy correspondence estimation, since the views are similar,

+ small regions of viewpoint related occlusions[1],

– small triangulation angle, hence large depth uncertainty.

The **wide baseline stereo** in contrast is used mostly with still image photographs of a scene where few images are taken from a very different viewpoint. Here the depth resolution is superior but correspondence and occlusion problems appear:

– hard correspondence estimation, since the views are not similar,

– large regions of viewpoint related occlusions,

+ big triangulation angle, hence high depth accuracy.

The **multi-viewpoint linking** combines the virtues of both approaches. In addition it will produce denser depth maps than either of the other techniques, and allows additional features for depth and texture fusion. Advantages are:

+ very dense depth maps for each viewpoint,

+ no viewpoint dependent occlusions,

+ highest depth resolution through viewpoint fusion,

+ texture enhancement (mean texture, highlight removal, super-resolution texture).

7.3.1   Correspondence Linking Algorithm

The correspondence linking is described in this section. It concatenates corresponding image points over multiple viewpoints by correspondence tracking over adjacent image pairs. This of course implies that the individually measured pair matches are accurate. To account for outliers in pair matches, some robust control strategies need to be employed to check the validity of the correspondence linking. Consider an image sequence taken from $k = [1, N]$ viewpoints. Assume that the sequence is taken by a camera moving sideways while keeping the object in view. For any view point k let us consider the image triple $[\mathbf{I}_{k-1}, \mathbf{I}_k, \mathbf{I}_{k+1}]$. The image pairs $(\mathbf{I}_{k-1}, \mathbf{I}_k)$ and $(\mathbf{I}_k, \mathbf{I}_{k+1})$ form two stereoscopic image pairs with correspondence estimates as described above. We have now defined 3 representations of image and camera matrices for each viewpoint: the original image $\mathbf{I}_k$ and projection matrix $\mathbf{P}_k$, their transformed versions $\mathbf{I}_k^{k-1}, \mathbf{P}_k^{k-1}$ rectified towards view point $k-1$ with transformation $\mathbf{R}_k^{k-1}$ and the transformed $\mathbf{I}_k^{k+1}, P_k^{k+1}$ rectified towards viewpoint $k+1$ with mapping $\mathbf{R}_k^{k+1}$. The Disparity map $D_{(k,k-1)}$ holds the downward correspondences from $\mathbf{I}_k^{k-1}$ to $\mathbf{I}_{k-1}^k$ while the map $D_{(k,k+1)}$ contains the upward correspondences from $\mathbf{I}_k^{k+1}$ to $\mathbf{I}_{k+1}^k$. We can now create two chains of correspondence links for an image point $\mathbf{m}_k$, one up and one down the image index k.

---

[1]As view point related occlusions we consider those parts of the object that are visible in one image only, due to object self-occlusion.

Upwards linking: $\quad \mathtt{m_{k+1}} = \left(\mathbf{R}_{k+1}^{k}\right)^{-1} D_{(k,k+1)}[\mathbf{R}_{k}^{k+1}\mathtt{m}_k]$

Downwards linking: $\quad \mathtt{m_{k-1}} = \left(\mathbf{R}_{k-1}^{k}\right)^{-1} D_{(k,k-1)}[\mathbf{R}_{k}^{k-1}\mathtt{m}_k]$

This linking process is repeated along the image sequence to create a chain of correspondences upwards and downwards. Every correspondence link requires 2 mappings and 1 disparity lookup. Throughout the sequence of N images, $2(N-1)$ disparity maps are computed. The multi-viewpoint linking is then performed efficiently via fast lookup functions on the pre-computed estimates.

Due to the rectification mapping transformed image point will normally not fall on integer pixel co-ordinates in the rectified image. The lookup of an image disparity in the disparity map D will therefore require an interpolation function. Since disparity maps for piecewise continuous surfaces have a spatially low frequency content, a bilinear interpolation between pixels suffices.

Occlusions and visibility

In a triangulation sensor with two viewpoints k and l two types of occlusion occur. If parts of the object are hidden in both viewpoints due to object self-occlusion, then we speak of **object occlusions** which cannot be resolved from this viewpoint. If a surface region is visible in viewpoint k but not in l, we speak of a **shadow occlusion**. The regions have a shadow-like appearance of undefined disparity values since the occlusions at view l cast a shadow on the object as seen from view k. Shadow occlusions are in fact detected by the uniqueness constraint discussed in section 7.2. A solution to avoid shadow occlusions is to incorporate a symmetrical multi-viewpoint matcher as proposed in this contribution. Points that are shadowed in the (right) view $k+1$ are normally visible in the (left) view $k-1$ and vice versa. The exploitation of up- and down-links will resolve for most of the shadow occlusions. A helpful measure in this context is the visibility V that defines for a pixel in view k the maximum number of possible correspondences in the sequence. $V = 1$ is caused by a shadow occlusion, $V >= 2$ allows a depth estimate.

Depth estimation and outlier detection

Care must be taken to exclude invalid disparity values or outliers from the chain. If an invalid disparity value is encountered, the chain is terminated immediately. Outliers are detected by controlling the statistics of the depth estimate computed from the correspondences. Inliers will update the depth estimate using a 1-D Kalman filter.

Depth and uncertainty Assume a 3D surface point M that is projected onto its corresponding image points $\mathtt{m}_k = \mathbf{P}_k\mathtt{M}, \mathtt{m}_l = \mathbf{P}_l\mathtt{M}$. The inverse process holds for triangulating M from the corresponding point pair $(\mathtt{m}_k, \mathtt{m}_l)$. We can in fact exploit the calibrated camera geometry and express the 3D point M as a depth value $d_\mathtt{M}$ along the known line of sight $\mathtt{L}_{\mathtt{m}_k}$ that extends from the camera projection center through the image correspondence $\mathtt{m}_k$. Triangulation computes the depth as the length of $\mathtt{L}_{\mathtt{m}_k}$ connecting the camera projection center and the locus of minimum distance between the corresponding lines of sight. The triangulation is computed for each image point and stored in a dense depth map associated with the viewpoint.

The depth for each reference image point $\mathtt{x}_k$ is improved by the correspondence linking that delivers two lists of image correspondences relative to the reference, one linking down from $k \to 1$ and one linking up from $k \to N$. For each valid corresponding point pair $(\mathtt{m}_i, \mathtt{m}_k)$ we can triangulate a consistent depth estimate $d(\mathtt{m}_k, \mathtt{m}_l)$ along $\mathtt{L}_{\mathtt{m}_k}$ with e_l representing the depth uncertainty. Figure 7.17(left) visualizes the decreasing uncertainty interval during linking. While the disparity measurement resolution ΔD in the image is kept constant (at 1 pixel), the reprojected depth error e_l decreases with the baseline.

Outlier detection and inlier fusion As measurement noise we assume a contaminated Gaussian distribution with a main peak within a small interval (of 1 pixel) and a small percentage of outliers. Inlier noise is caused by the limited resolution of the disparity matcher and by the interpolation artifacts. Outliers are undetected correspondence failures and may be arbitrarily large. As threshold to detect the outliers we utilize the depth uncertainty interval e_k. The detection of an outlier at k terminates the linking at $k-1$. All depth values $[d_k, d_{k+1}, ..., d_{l-1}]$ are inlier depth values that fall within the uncertainty interval around
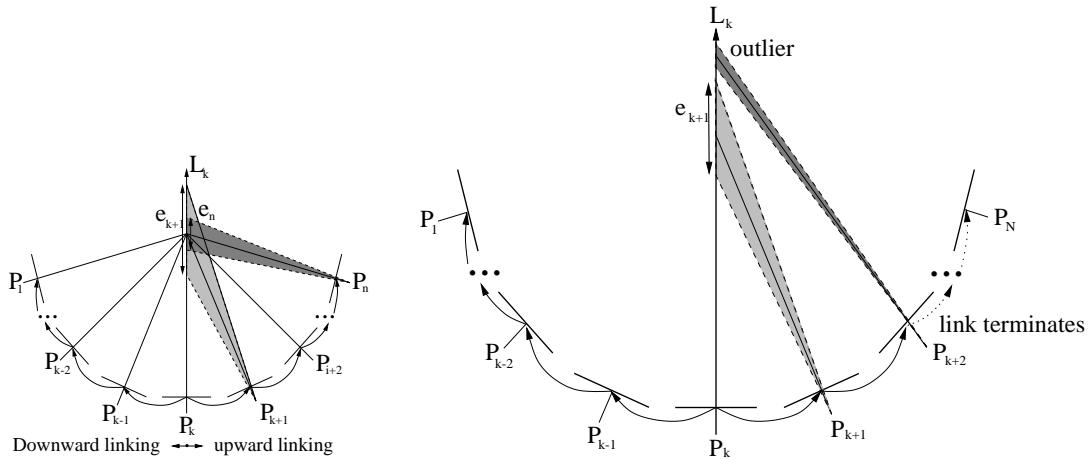
Figure 7.17: Depth fusion and uncertainty reduction from correspondence linking (left). Detection of correspondence outliers by depth interval testing (right).

the mean depth estimate. They are fused by a simple 1-D kalman filter to obtain an optimal mean depth estimate.

Figure 7.17(right) explains the outlier selection and link termination for the up-link. The outlier detection scheme is not optimal since it relies on the position of the outlier in the chain. Valid correspondences behind the outlier are not considered anymore. It will, however, always be as good as a single estimate and in general superior to it. In addition, since we process bidirectionally up- and down-link, we always have two correspondence chains to fuse which allows for one outlier per chain.

7.3.2 Some results

In this section the performance of the algorithm is tested on the two outdoor sequences *Castle* and *Fountain*.

Castle sequence   The *Castle* sequence consists of images of 720x576 pixel resolution taken with a standard semi-professional camcorder that was moved freely in front of a building. The quantitative performance of correspondence linking can be tested in different ways. One measure already mentioned is the visibility of an object point. In connection with correspondence linking, we have defined visibility V as the number of views linked to the reference view. Another important feature of the algorithm is the density and accuracy of the depth maps. To describe its improvement over the 2-view estimator, we define the fill rate F and the average relative depth error E as additional measures.

Visibility $V[views]$:   average number of views linked to the reference image.

Fill Rate $F[\%]$: $\dfrac{\text{Number of valid pixels}}{\text{Total number of pixels}}$

Depth error $E[\%]$:   standard deviation of relative depth error e_d for all valid pixels.

The 2-view disparity estimator is a special case of the proposed linking algorithm, hence both can be compared on an equal basis. The 2-view estimator operates on the image pair $(k, k+1)$ only, while the multi-view estimator operates on a sequence $1 < k < N$ with $N >= 3$. The above defined statistical measures were computed for different sequence lengths N. Figure 7.18 displays visibility and relative depth error for sequences from 2 to 15 images, chosen symmetrically around the reference image. The average visibility V shows that for up to 5 images nearly all views are utilized. For 15 images, at average 9 images are linked. The amount of linking is reflected in the relative depth error that drops from 5% in the 2 view estimator to about 1.2% for 15 images.

Linking two views is the minimum case that allows triangulation. To increase the reliability of the estimates, a surface point should occur in more than two images. We can therefore impose a minimum
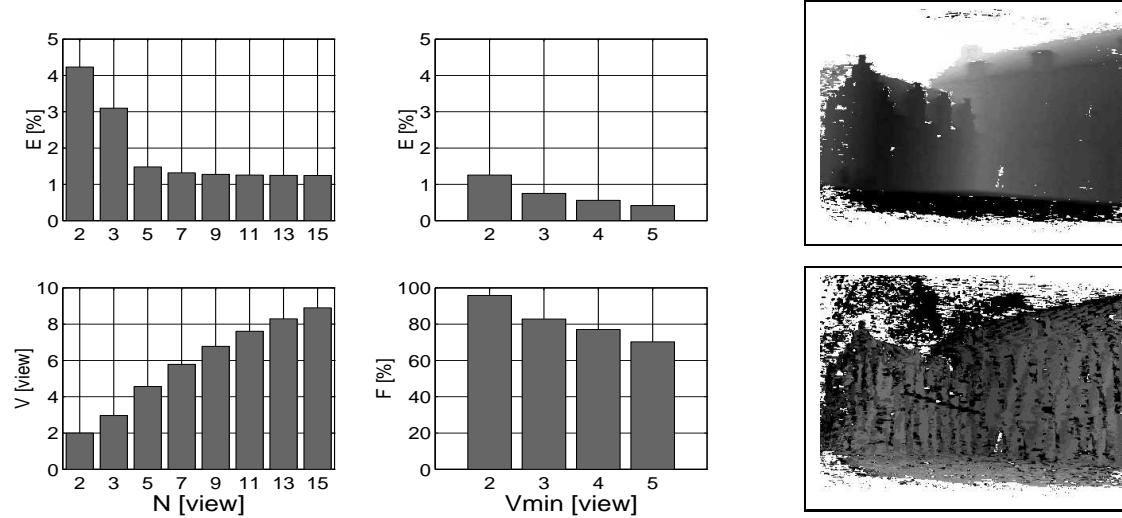
Figure 7.18: Statistics of the castle sequence. Influence of sequence length N on visibility V and relative depth error E. (left) Influence of minimum visibility V_{min} on fill rate F and depth error E for $N = 11$ (center). Depth map (above: dark=near, light=far) and error map (below: dark=large error, light=small error) for $N = 11$ and $V_{min} = 3$ (right).

visibility V_{min} on a depth estimate. This will reject unreliable depth estimates effectively, but will also reduce the fill rate of the depth map.

The graphs in figure 7.18(center) show the dependency of the fill rate and depth error on minimum visibility for N=11. The fill rate drops from 92% to about 70%, but at the same time the depth error is reduced to 0.5% due to outlier rejection. The depth map and the relative error distribution over the depth map is displayed in Figure 7.18(right). The error distribution shows a periodic structure that in fact reflects the quantization uncertainty of the disparity resolution when it switches from one disparity value to the next.

Fountain sequence The *Fountain* sequence consists of 5 images of the back wall of the Upper Agora at the archaeological site of Sagalassos in Turkey, taken with a digital camera with 573x764 pixel resolution. It shows a concavity in which once a statue was situated.

N[view]	$V[views]$	$F[\%]$	$E[\%]$
2	2	89.8728	0.294403
3	2.85478	96.7405	0.208367
5	4.23782	96.4774	0.121955

Table 7.1: Statistics of the fountain sequence for visibility V, fill rate F and depth error E.

The performance characteristics are displayed in the table 7.1. The fill rate is high and the relative error is rather low because of a fairly wide baseline between views. This is reflected in the high geometric quality of depth the map and the reconstruction. Figure 7.19 shows from left to right images 1 and 3 of the sequence, the depth map as computed with the 2-view estimator, and the depth map when using all 5 images. The white (undefined) regions in the 2-view depth map are due to shadow occlusions which are almost completely removed in the 5-view depth map. This is reflected in the fill rate that increases from 89 to 96%. It should be noted that for this sequence a very large search range of 400 pixels was used, which is over 70% of the image width. Despite this large search range only few matching errors occurred.

Figure 7.19: First and last image of fountain sequence (left). Depth maps from the 2-view and the 5-view estimator (from left to right) showing the very dense depth maps (right).

7.4 Conclusion

In this chapter we presented a scheme that computes dense and accurate depth maps based on the sequence linking of pairwise estimated disparity maps. First a matching algorithm was presented which computes corresponding points for an image pair in standard stereo configuration. Then it was explained how images can be rectified so that any pair of images can be brought to this configuration. Finally a multi-view linking approach was presented which allows to combine the results to obtain more accurate and dense depth maps. The performance analysis showed that very dense depth maps with fill rates of over 90 % and a relative depth error of 0.1% can be measured with off-the-shelf cameras even in unrestricted outdoor environments such as an archaeological site.

Chapter 8

Modeling

In the previous chapters we have seen how the information needed to build a 3D model could automatically be obtained from images. This chapter explains how this information can be combined to build realistic representations of the scene. It is not only possible to generate a surface model or volumetric model easily, but all the necessary information is available to build lightfield models or even augmented video sequences. These different cases will now be discussed in more detail.

8.1   Surface model

The 3D surface is approximated by a triangular mesh to reduce geometric complexity and to tailor the model to the requirements of computer graphics visualization systems. A simple approach consists of overlaying a 2D triangular mesh on top of the image and then build a corresponding 3D mesh by placing the vertices of the triangles in 3D space according to the values found in the depth map. To reduce noise it is recommended to first smooth the depth image (the kernel can be chosen of the same size as the mesh triangles). The image itself can be used as texture map (the texture coordinates are trivially obtained as the 2D coordinates of the vertices).

It can happen that for some vertices no depth value is available or that the confidence is too low (see Section 7.2.2). In these cases the corresponding triangles are not reconstructed. The same happens when triangles are placed over discontinuities. This is achieved by selecting a maximum angle between the normal of a triangle and the line of sight through its center (e.g. 85 degrees).

This simple approach works very well on the depth maps obtained after multi-view linking. On simple stereo depth maps it is recommended to use a more advanced technique described in [73]. In this case the boundaries of the objects to be modeled are computed through depth segmentation. In a first step, an object is defined as a connected region in space. Simple morphological filtering removes spurious and very small regions. Then a bounded thin plate model is employed with a second order spline to smooth the surface and to interpolate small surface gaps in regions that could not be measured.

The surface reconstruction approach is illustrated in Figure 8.1. The obtained 3D surface model is shown in Figure 8.2 with shading and with texture. Note that this surface model is reconstructed from the viewpoint of a reference image. If the whole scene can not be seen from one image, it it necessary to apply a technique to fuse different surfaces together (e.g. [162, 19]).

8.1.1   Texture enhancement

The correspondence linking builds a controlled chain of correspondences that can be used for texture enhancement as well. At each reference pixel one may collect a sorted list of image color values from the corresponding image positions. This allows to enhance the original texture in many ways by accessing the color statistics. Some features that are derived naturally from the linking algorithm are:

Figure 8.1: Surface reconstruction approach: A triangular mesh (left) is overlaid on top of the image (middle). The vertices are back-projected in space according to the value found in the depth map (right).
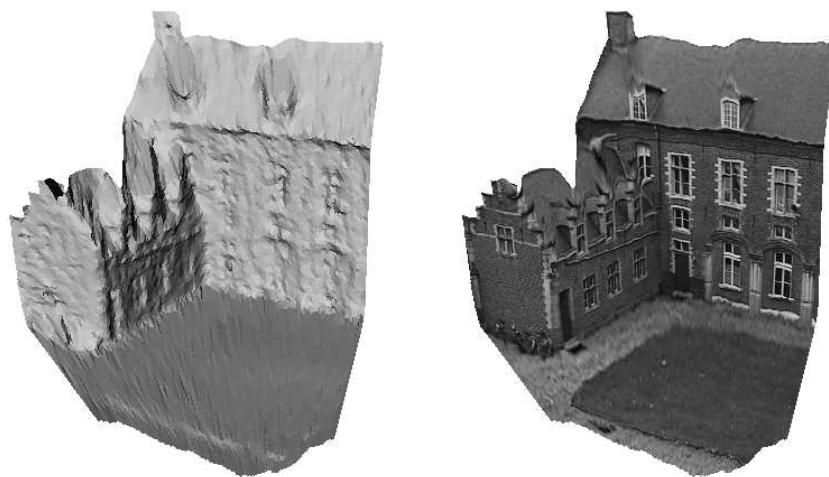

Figure 8.2: *3D surface model obtained automatically from an uncalibrated image sequence, shaded (left), textured (right).*

Figure 8.3: close-up view (left), 4x zoomed original region (top-right), generation of median-filtered super-resolution texture (bottom-right).

Highlight and reflection removal   : A median or robust mean of the corresponding texture values is computed to discard imaging artifacts like sensor noise, specular reflections and highlights[93]. An example of highlight removal is shown in Figure 8.3.

Super-resolution texture   : The correspondence linking is not restricted to pixel-resolution, since each sub-pixel-position in the reference image can be used to start a correspondence chain. The correspondence values are queried from the disparity map through interpolation. The object is viewed by many cameras of limited pixel resolution, but each image pixel grid will in general be slightly displaced. This can be exploited to create super-resolution texture by fusing all images on a finer resampling grid[61].

Best view selection for highest texture resolution   : For each surface region around a pixel the image which has the highest possible texture resolution is selected, based on the object distance and viewing angle. The composite image takes the highest possible resolution from all images into account.

Example

We have recorded a short video sequence from a medusa head decorating an ancient fountain in Sagalassos (an ancient city in Turkey). The 20 second video sequence was recorded with a hand-held consumer video camera (Sony TRV-900). Each twentiest frame was used as a key-frame by our video to 3D processing pipeline. Three of these frames are seen on the top part of Figure 8.4. The compute structure and motion is also seen in this figure (middle-left). The camera viewpoints are represented by small pyramids. The depth map used to construct the 3D model is seen on the middle-right of the figure. Finally, the model -with and without texture- is seen at the bottom of the figure. From the shaded model one can see that most of the geometric detail is accurately recovered. By using the image itself as texture map a photorealistic model is obtained. Note from the rightmost view that the 3D model allows to render realistic views that are very different from the original views.

8.1.2   Volumetric integration

To generate a complete 3D model from different depth maps we propose to use the volumetric integration approach of Curless and Levoy [19]. This approach is described in this section.
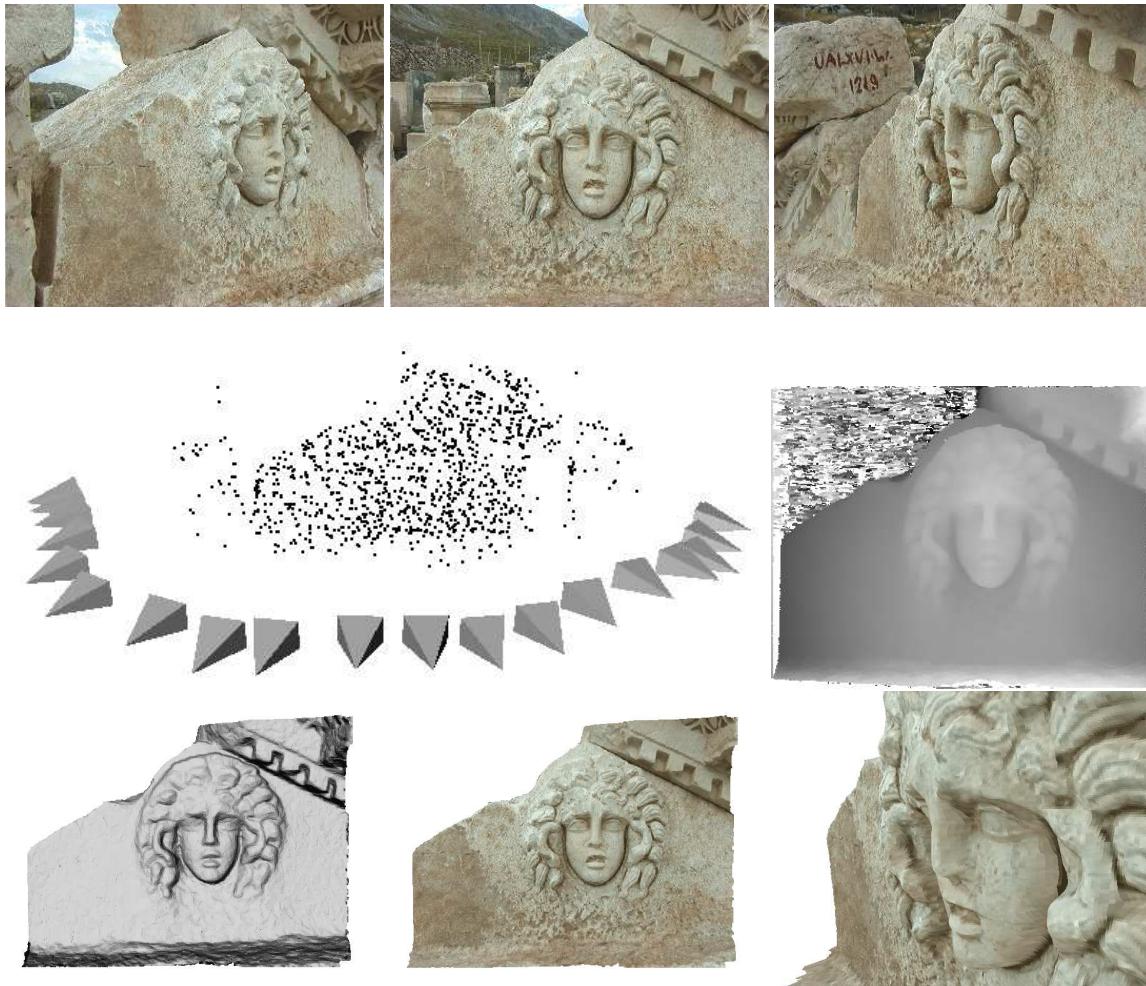
Figure 8.4: 3D model of a decorative Medusa head recorded at the ancient site of Sagalassos in Turkey. Top: 3 views of the original video, middle: reconstruction of 3D feature points with computed camera motion for the keyframes and one of the computed depth/range images, bottom: shaded and textured views of the recovered 3D model.
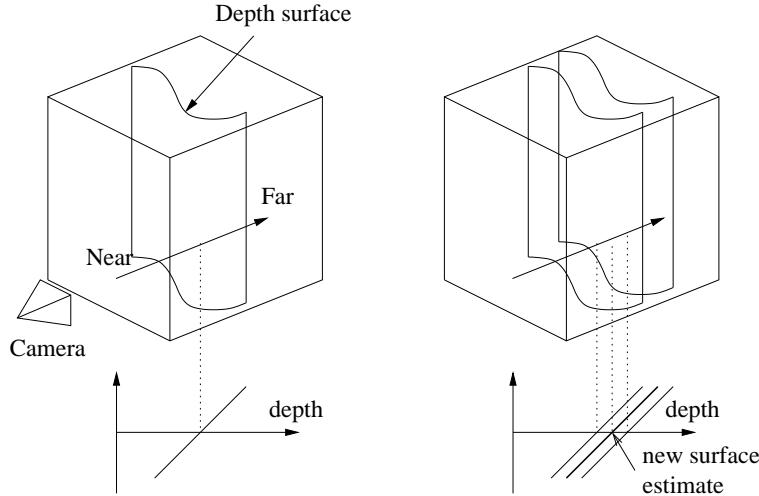
Figure 8.5: Unweighted signed distance functions in 3D. (a) A camera looking down the x-axis observes a depth image, shown here as a reconstructed surface. Following one line of sight down the x-axis, a signed distance function as shown can be generated. The zero-crossing of this function is a point on the surface. (b) Another depth map yields a slightly different surface due to noise. Following the same line of sight as before, we obtain another signed distance function. By summing these functions, we arrive at a cumulative function with a new zero-crossing positioned midway between the original range measurements.

The algorithm employs a continuous implicit function, $D(\texttt{M})$, represented by samples. The function we represent is the weighted signed distance of each point to the nearest range surface along the line of sight to the sensor. We construct this function by combining signed distance functions, $d_1(\texttt{M}), d_2(\texttt{M}) \ldots d_n(\texttt{M})$ and weight functions $w_1(\texttt{M}), w_2(\texttt{M}) \ldots w_n(\texttt{M}_n)$ obtained from the depth maps for the different images. The combining rules gives a cumulative signed distance function for each voxel, $D(\texttt{M})$, and a cumulative weight $W(\texttt{M})$. These functions are represented on a discrete voxel grid and an isosurface corresponding to $D(x) = 0$ is extracted. Under a certain set of assumptions, this isosurface is optimal in the least squares sense [18].

Figure 8.5 illustrates the principle of combining unweighted signed distances for the simple case of two range surfaces sampled from the same direction. Note that the resulting isosurface would be the surface created by averaging the two range surfaces along the sensor's lines of sight. In general, however, weights are necessary to represent variations in certainty across the range surfaces. The choice of weights should be specific to the depth estimation procedure. It is proposed to make weights depend on the dot product between each vertex normal and the viewing direction, reflecting greater uncertainty when the observation is at grazing angles to the surface, as Soucy [138] proposed for optical triangulation scanners. Depth values at the boundaries of a mesh typically have greater uncertainty, requiring more down-weighting.

Figure 8.6 illustrates the construction and usage of the signed distance and weight functions in 1D. In Figure 8.6a, the sensor is positioned at the origin looking down the +x axis and has taken two measurements, r_1 and r_2. The signed distance profiles, $d_1(x)$ and $d_2(x)$ may extend indefinitely in either direction, but the weight functions, $w_1(x)$ and $w_2(x)$, taper off behind the range points for reasons discussed below.

Figure 8.6b is the weighted combination of the two profiles. The combination rules are straightforward:

$$D(x) \quad = \quad \frac{\sum w_i(x) d_i(x)}{\sum w_i(x)} \tag{8.1}$$

$$W(x) \quad = \quad \sum w_i(x) \tag{8.2}$$

where, $d_i(x)$ and $w_i(x)$ are the signed distance and weight functions from the ith range image. Expressed as an incremental calculation, the rules are:

$$D_{i+1}(x) \quad = \quad \frac{W_i(x) D_i(x) + w_i(x) d_i(x)}{\sum W_i(x) + w_{i+1}(x)} \tag{8.3}$$
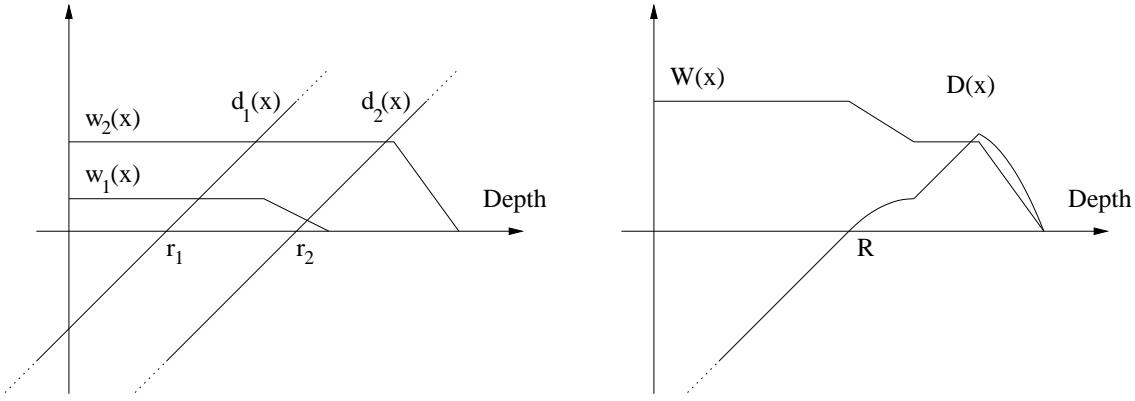
Figure 8.6: Signed distance and weight functions in one dimension. (a) The sensor looks down the x-axis and takes two measurements, r_1 and r_2 . $d_1(x)$ and $d_2(x)$ are the signed distance profiles, and $w_1(x)$ and $w_2(x)$ are the weight functions. In 1D, we might expect two sensor measurements to have the same weight magnitudes, but we have shown them to be of different magnitude here to illustrate how the profiles combine in the general case. (b) $D(x)$ is a weighted combination of $d_1(x)$ and $d_2(x)$, and $W(x)$ is the sum of the weight functions. Given this formulation, the zero-crossing, R, becomes the weighted combination of r_1 and r_2 and represents our best guess of the location of the surface. In practice, we truncate the distance ramps and weights to the vicinity of the range points.

$$W_{i+1}(x) \quad = \quad W_i(x) + w_i(x) \tag{8.4}$$

where $D_I(x)$ and $W_i(x)$ are the cumulative signed distance and weight functions after integrating the ith range image. In the special case of one dimension, the zero-crossing of the cumulative function is at a range, R given by:

$$R = \frac{\sum w_i r_i}{\sum w_i} \tag{8.5}$$

i.e., a weighted combination of the acquired range values, which is what one would expect for a least squares minimization.

In principle, the distance and weighting functions should extend indefinitely in either direction. However, to prevent surfaces on opposite sides of the object from interfering with each other, we force the weighting function to taper off behind the surface. There is a trade-off involved in choosing where the weight function tapers off. It should persist far enough behind the surface to ensure that all distance ramps will contribute in the vicinity of the final zero crossing, but, it should also be as narrow as possible to avoid influencing surfaces on the other side. To meet these requirements, we force the weights to fall off at a distance equal to half the maximum uncertainty interval of the depth measurements. Similarly, the signed distance and weight functions need not extend far in front of the surface. Restricting the functions to the vicinity of the surface yields a more compact representation and reduces the computational expense of updating the volume.

In two and three dimensions, the depth measurements correspond to curves or surfaces with weight functions, and the signed distance ramps have directions that are consistent with the primary directions of sensor uncertainty.

For three dimensions, we can summarize the whole algorithm as follows. First, we set all voxel weights to zero, so that new data will overwrite the initial grid values. The signed distance contribution is computed by making the difference between the depth read out at the projection of the grid point in the depth map and the actual distance between the point and the camera projection center. The weight is obtained from a weight map that has been precomputed. Having determined the signed distance and weight we can apply the update formulae described in equations (8.3) and (8.4).

At any point during the merging of the range images, we can extract the zero-crossing isosurface from the volumetric grid. We restrict this extraction procedure to skip samples with zero weight, generating triangles only in the regions of observed data. The procedure used for this is marching cubes [81].
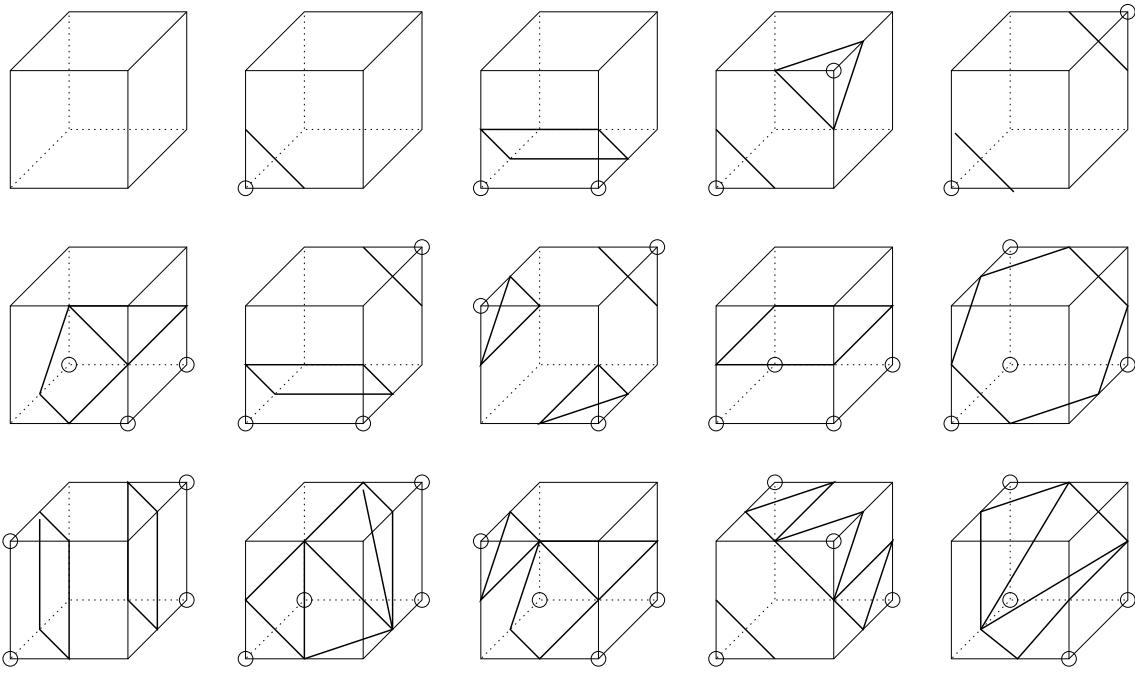
Figure 8.7: The 14 different configurations for marching cubes.

Marching cubes   Marching Cubes is an algorithm for generating isosurfaces from volumetric data. If one or more voxels of a cube have values less than the targeted isovalue, and one or more have values greater than this value, we know the voxel must contribute some component of the isosurface. By determining which edges of the cube are intersected by the isosurface, triangular patches can be created which divide the cube between regions within the isosurface and regions outside. By connecting the patches from all cubes on the isosurface boundary, a surface representation is obtained.

There are two major components of this algorithm. The first is deciding how to define the section or sections of surface which chop up an individual cube. If we classify each corner as either being below or above the isovalue, there are 256 possible configurations of corner classifications. Two of these are trivial; where all points are inside or outside the cube does not contribute to the isosurface. For all other configurations we need to determine where, along each cube edge, the isosurface crosses, and use these edge intersection points to create one or more triangular patches for the isosurface.

If you account for symmetries, there are really only 14 unique configurations in the remaining 254 possibilities. When there is only one corner less than the isovalue, this forms a single triangle which intersects the edges which meet at this corner, with the patch normal facing away from the corner. Obviously there are 8 related configurations of this sort. By reversing the normal we get 8 configurations which have 7 corners less than the isovalue. We don't consider these really unique, however. For configurations with 2 corners less than the isovalue, there are 3 unique configurations, depending on whether the corners belong to the same edge, belong the same face of the cube, or are diagonally positioned relative to each other. For configurations with 3 corners less than the isovalue there are again 3 unique configurations, depending on whether there are 0, 1, or 2 shared edges (2 shared edges gives you an 'L' shape). There are 7 unique configurations when you have 4 corners less than the isovalue, depending on whether there are 0, 2, 3 (3 variants on this one), or 4 shared edges. The different cases are illustrated in Figure 8.7

Each of the non-trivial configurations results in between 1 and 4 triangles being added to the isosurface. The actual vertices themselves can be computed by interpolation along edges.

Now that we can create surface patches for a single voxel, we can apply this process to the entire volume. We can process the volume in slabs, where each slab is comprised of 2 slices of pixels. We can either treat each cube independently, or we can propogate edge intersections between cubes which share the edges. This sharing can also be done between adjacent slabs, which increases storage and complexity a
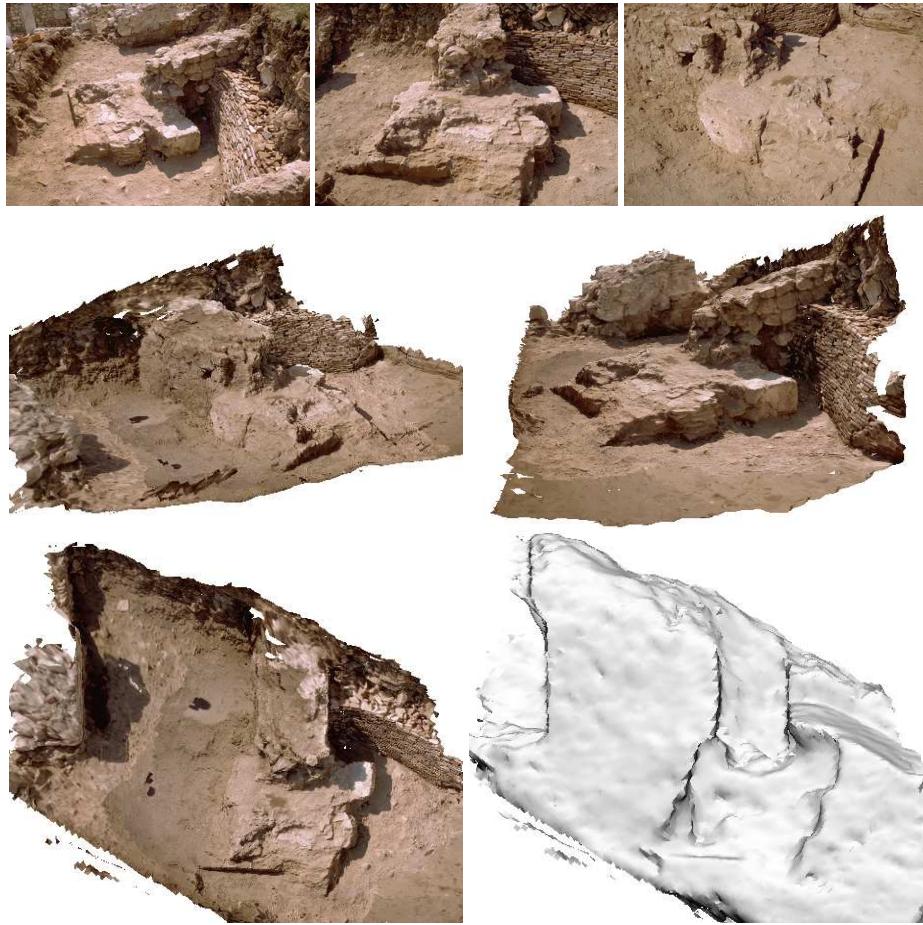
Figure 8.8: Integrated 3D representation of the excavations of an ancient roman villa in Sagalassos. Top: two side-views of the 3D model, bottom: texture and shaded top-view of the 3D model.

bit, but saves in computation time. The sharing of edge/vertex information also results in a more compact model, and one that is more amenable to interpolated shading.

Example

This example was also recorded on the archaeological site of Sagalassos. In this case the remains of a Roman villa were recorded at different stages during the excavations. Here we consider a specific layer for which 26 pictures were taken with a hand-held photo camera (Nikon F50) and scanned to PhotoCD. The on site acquisition of the images only takes a few minutes so it does not slow down the excavation process. Some of the recorded pictures can be seen on the top part of Figure 8.8. Note that in this case the geometry of the observed scene is too complex to be reconstructed from a single depth map. Therefore, in this case the 3D model was obtained by combining all the depth maps using a volumetric approach.

8.2   Lightfield model

In this section our goal is to create a lightfield model from a scene to render new views interactively. Our approach has been presented in a number of consecutive papers [68, 67, 56]. For rendering new views two major concepts are known in literature. The first one is the geometry based concept. The scene geometry is reconstructed from a stream of images and a single texture is synthesized which is mapped onto this geometry. For this approach, a limited set of camera views is sufficient, but specular effects can

not be handled appropriately. This approach has been discussed extensively in this text. The second major concept is image-based rendering. This approach models the scene as a collection of views all around the scene without an exact geometrical representation [78]. New (virtual) views are rendered from the recorded ones by interpolation in real-time. Optionally approximate geometrical information can be used to improve the results [39]. Here we concentrate on this second approach. Up to now, the known scene representation has a fixed regular structure. If the source is an image stream taken with a hand-held camera, this regular structure has to be resampled. Our goal is to use the recorded images themselve as scene representation and to directly render new views from them. Geometrical information is considered as far as it is known and as detailed as the time for rendering allows. The approach is designed such, that the operations consist of projective mappings only which can efficiently be performed by the graphics hardware (this comes very close to the approach described in [23]). For each of these scene modeling techniques the camera parameters for the original views are supposed to be known. We retrieve them by applying known structure and motion techniques as described in the previous chapters. Local depth maps are calculated applying stereo techniques on rectified image pairs as previously explained.

8.2.1 structure and motion

To do a dense lightfield modeling as described below, we need many views from a scene from many directions. For this, we can record an extended image sequence moving the camera in a zigzag like manner. The camera can cross its own moving path several times or at least gets close to it. Known calibration methods usually only consider the neighborhoods within the image stream. Typically no linking is done between views whose position is close to each other in 3-D space but which have a large distance in the sequence. To deal with this problem, we therefore exploit the 2-D topology of the camera viewpoints to further stabilize the calibration. We process not only the next sequential image but search for those images in the stream that are nearest in the topology to the current viewpoint. Typically we can establish a reliable matching to 3-4 neighboring images which improves the calibration considerably. The details were described in Section 5.2.2. We will also show how to use local depth maps for improving rendering results. To this end dense correspondence maps are computed for adjacent image pairs of the sequence (see Chapter 7).

8.2.2 Lightfield modeling and rendering

In [87] the appearance of a scene is described through all light rays (2D) that are emitted from every 3D scene point, generating a 5D radiance function. Subsequently two equivalent realizations of the plenoptic function were proposed in form of the lightfield [78], and the lumigraph [39]. They handle the case when the observer and the scene can be separated by a surface. Hence the plenoptic function is reduced to four dimensions. The radiance is represented as a function of light rays passing through the separating surface. To create such a plenoptic model for real scenes, a large number of views is taken. These views can be considered as a collection of light rays with according color values. They are discrete samples of the plenoptic function. The light rays which are not represented have to be interpolated from recorded ones considering additional information on physical restrictions. Often real objects are supposed to be lambertian, meaning that one point of the object has the same radiance value in all possible directions. This implies that two viewing rays have the same color value, if they intersect at a surface point. If specular effects occur, this is not true any more. Two viewing rays then have similar color values if their direction is similar and if their point of intersection is near the real scene point which originates their color value. To render a new view we suppose to have a virtual camera looking at the scene. We determine those viewing rays which are nearest to those of this camera. The nearer a ray is to a given ray, the greater is its support to the color value.

The original 4D lightfield [78] data structure employs a two-plane parameterization. Each light ray passes through two parallel planes with plane coordinates (s, t) and (u, v). The (u, v)-plane is the *viewpoint plane* in which all camera focal points are placed on regular grid points. The (s, t)-plane is the *focal plane*. New views can be rendered by intersecting each viewing ray of a virtual camera with the two planes at (s, t, u, v). The resulting radiance is a look-up into the regular grid. For rays passing in between the (s, t) and (u, v) grid coordinates an interpolation is applied that will degrade the rendering quality depending

on the scene geometry. In fact, the lightfield contains an implicit geometrical assumption, i.e. the scene geometry is planar and coincides with the focal plane. Deviation of the scene geometry from the focal plane causes image degradation (i.e. blurring or ghosting). To use hand-held camera images, the solution proposed in [39] consists of *rebinning* the images to the regular grid. The disadvantage of this rebinning step is that the interpolated regular structure already contains inconsistencies and ghosting artifacts because of errors in the scantily approximated geometry. During rendering the effect of ghosting artifacts is repeated so duplicate ghosting effects occur.

Rendering from recorded images   Our goal is to overcome the problems described in the last section by relaxing the restrictions imposed by the regular lightfield structure and to render views directly from the calibrated sequence of recorded images with use of local depth maps. Without loosing performance the original images are directly mapped onto one or more planes viewed by a virtual camera.

To obtain a high-quality image-based scene representation, we need many views from a scene from many directions. For this, we can record an extended image sequence moving the camera in a zigzag like manner. The camera can cross its own moving path several times or at least gets close to it. To obtain a good quality structure-and-motion estimation from this type of sequence it is important to use the extensions proposed in Section 5.2 to match close views that are not predecessors or successors in the image stream. To allow to construct the local geometrical approximation depth maps should also be computed as described in the previous section.

Fixed plane approximation   In a first approach, we approximate the scene geometry by a single plane L by minimizing the least square error. We map all given camera images onto plane L and view it through a virtual camera. This can be achieved by directly mapping the coordinates x_i, y_i of image i onto the virtual camera coordinates $[x_V \ y_V \ 1]^\top = \mathbf{H}_{iV}[x_i \ y_i \ 1]^\top$. Therefore we can perform a direct look-up into the originally recorded images and determine the radiance by interpolating the recorded neighboring pixel values. This technique is similar to the lightfield approach [78] which implicitly assumes the focal plane as the plane of geometry. Thus to construct a specific view we have to interpolate between neighboring views. Those views give the most support to the color value of a particular pixel whose projection center is close to the viewing ray of this pixel. This is equivalent to the fact that those views whose projected camera centers are close to its image coordinate give the most support to a specified pixel. We restrict the support to the nearest three cameras (see Figure 8.9). We project all camera centers into the virtual image and perform a 2D triangulation. Then the neighboring cameras of a pixel are determined by the corners of the triangle which this pixel belongs to. Each triangle is drawn as a sum of three triangles. For each camera we look up the color values in the original image like described above and multiply them with weight 1 at the corresponding vertex and with weight 0 at both other vertices. In between, the weights are interpolated linearly similar to the Gouraud shading. Within the triangle the sum of weights is 1 at each point. The total image is built up as a mosaic of these triangles. Although this technique assumes a very sparse approximation of geometry, the rendering results show only small ghosting artifacts (see experiments).

View-dependent geometry approximation   The results can be further improved by considering local depth maps. Spending more time for each view, we can calculate the approximating plane of geometry for each triangle in dependence on the actual view. This improves the accuracy further as the approximation is not done for the whole scene but just for that part of the image which is seen through the actual triangle. The depth values are given as functions D_i of the coordinates in the recorded images $D_i(x, y)$. They describe the distance of a point to the projection center. Using this depth function, we calculate the 3D coordinates of those scene points which have the same 2D image coordinates in the virtual view as the projected camera centers of the real views. The 3D point $\mathtt{M}_i$ which corresponds to view i can be calculated as

$$\mathtt{M}_i = s D_i(\mathbf{P}_k \mathtt{C}_V) n(\mathtt{C}_k - \mathtt{C}_V) + \mathtt{C}_k \tag{8.6}$$

where $n(\mathtt{A}) = \frac{\mathtt{A}}{\|\mathtt{A}\|}$ and $s = \text{sign}\,(\mathtt{P}_{3i}.(\mathtt{C}_k - \mathtt{C}_V))$ with $\mathtt{P}_{3i}$ the third row of $\mathbf{P}_i$ is needed for a correct orientation. We can interpret the points $\mathtt{M}_i$ as the intersection of the line $\overline{\mathtt{C}_V \mathtt{C}_k}$ with the scene geometry.
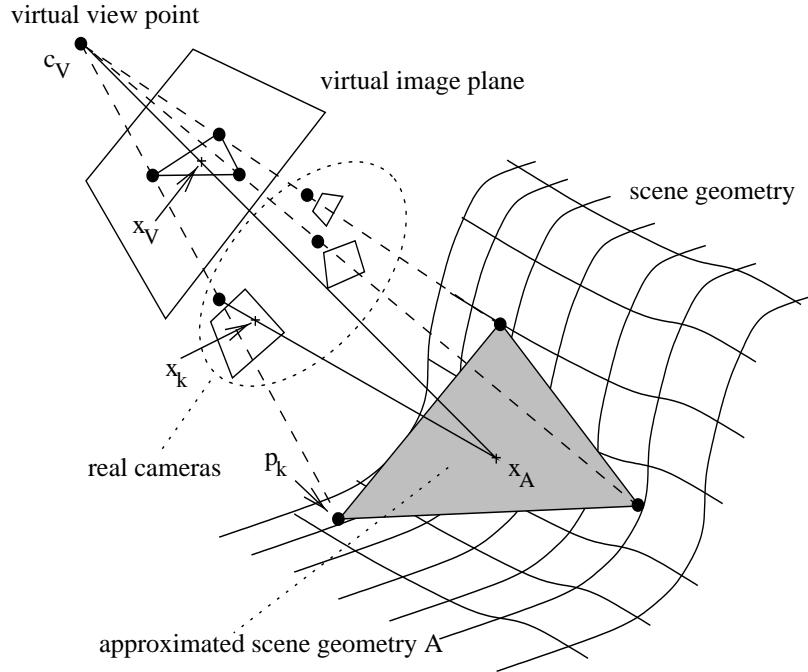
Figure 8.9: Drawing triangles of neighboring projected camera centers and approximating geometry by one plane for the whole scene, for one camera triple or by several planes for one camera triple.

Knowing the 3D coordinates of triangle corners, we can define a plane through them and apply the same rendering technique as described above.

Finally, if the triangles exceed a given size, they can be subdivided into four sub-triangles by splitting the three sides into two parts, each. For each of these sub-triangles, a separate approximative plane is calculated in the above manner. We determine the midpoint of the side and use the same look-up method as used for radiance values to find the corresponding depth. After that, we reconstruct the 3D point and project it into the virtual camera resulting in a point near the side of the triangle. Of course, further subdivision can be done in the same manner to improve accuracy. Especially, if just few triangles contribute to a single virtual view, this subdivision is really necessary. It should be done in a resolution according to performance demands and to the complexity of geometry.

8.2.3 Experiments

We have tested our approaches with an uncalibrated sequence of 187 images showing an office scene. Figure 8.10 (left) shows one particular image. A digital consumer video camera was swept freely over a cluttered scene on a desk, covering a viewing surface of about $1m^2$. Figure 8.10 (right) shows the calibration result. Figure 8.11 illustrates the success of the modified structure and motion algorithm as described in Section 5.2.2. Features that are lost are picked up again when they reappear in the images. Figure 8.12 (left) shows the calibration results with the viewpoint mesh. One result of a reconstructed view is shown in Figure 8.12 (right). Figure 8.13 shows details for the different methods. In the case of one global plane (left image), the reconstruction is sharp where the approximating plane intersects the actual scene geometry. The reconstruction is blurred where the scene geometry diverges from this plane. In the case of local planes (middle image), at the corners of the triangles, the reconstruction is almost sharp, because there the scene geometry is considered directly. Within a triangle, ghosting artifacts occur where the scene geometry diverges from the particular local plane. If these triangles are subdivided (right image) these artifacts are reduced further.
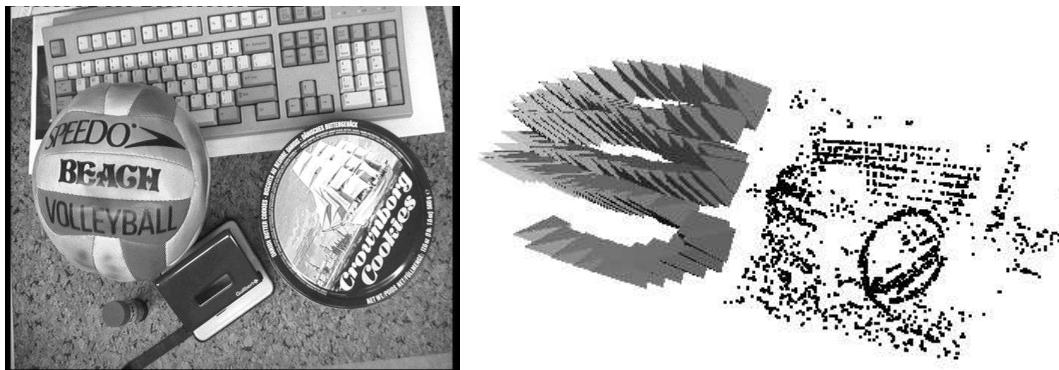
Figure 8.10: Image of the *desk* sequence (left) and result of calibration step (right). The cameras are represented by little pyramids.
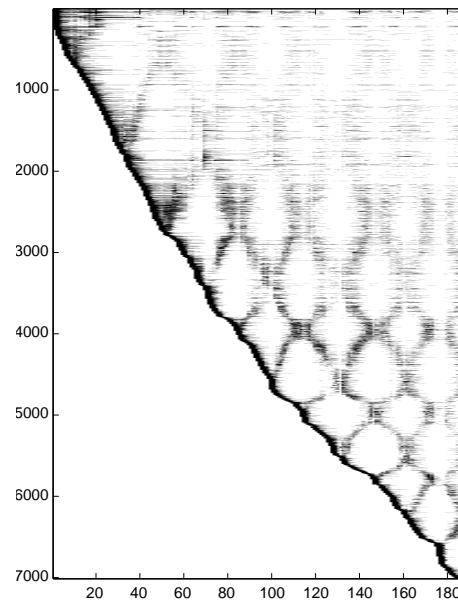

Figure 8.11: Tracking of the points over the sequence. Points (vertical) versus images (horizontal).
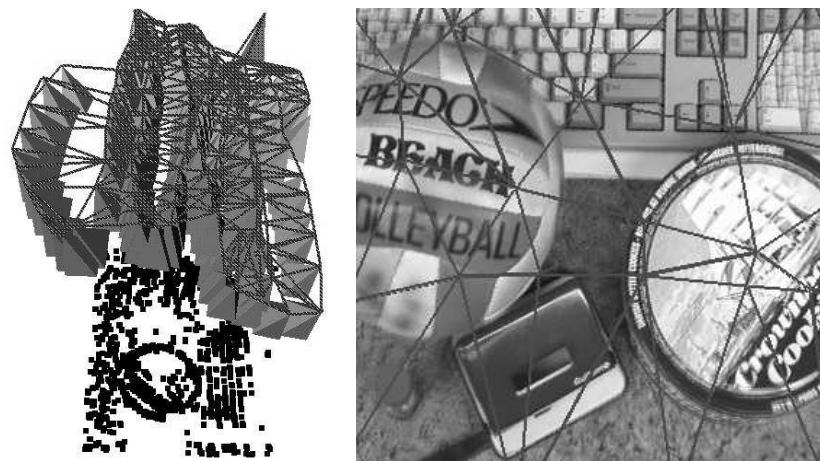
Figure 8.12: Calibration result and viewpoint mesh(left) and reconstructed scene view using one plane per image triple.

Figure 8.13: Details of rendered images showing the differences between the approaches: one global plane of geometry (left), one local plane for each image triple (middle) and refinement of local planes (right).

8.2.4   conclusion

In this section, we have shown how the proposed approach for modeling from images could easily be extended to allow the acquisition of lightfield models. The quality of rendered images can be varied by adjusting the resolution of the considered scene geometry. Up to now, our approaches are calculated in software. But they are designed such, that using alpha blending and texture mapping facilities of graphics hardware, rendering can be done in real-time. More details on this approach can be found in [68, 67, 56].

8.3   Fusion of real and virtual scenes

Another interesting possibility offered by the presented approach is to combine real and virtual scene elements. This allows to augment real environments with virtual objects. A first approach consists of virtualizing the real environment and then to place virtual objects in it. This can readily be done using the techniques presented in Section 8.1. An example is shown in Figure 8.14. The landscape of Sagalassos (an archaeological site in Turkey) was modeled from a dozen photographs taken from a nearby hill. Virtual reconstructions of ancient monuments have been made based on measurements and hypotheses of archaeologists. Both could then be combined in a single virtual world.

Figure 8.14: Virtualized landscape of Sagalassos combined with virtual reconstructions of monuments.

8.3.1   Augmenting video footage

Another challenging application consists of seamlessly merging virtual objects with real video. In this case the ultimate goal is to make it impossible to differentiate between real and virtual objects. Several problems need to be overcome before achieving this goal. Amongst them are the rigid registration of virtual objects into the real environment, the problem of mutual occlusion of real and virtual objects and the extraction of the illumination distribution of the real environment in order to render the virtual objects with this illumination model.

Here we will concentrate on the first of these problems, although the computations described in the previous section also provide most of the necessary information to solve for occlusions and other interactions between the real and virtual components of the augmented scene. Accurate registration of virtual objects into a real environment is still a challenging problems. Systems that fail to do so will fail to give the user a real-life impression of the augmented outcome. Since our approach does not use markers or a-priori knowledge of the scene or the camera, this allows us to deal with video footage of unprepared environments or archive video footage. More details on this approach can be found in [14].

An important difference with the applications discussed in the previous sections is that in this case all frames of the input video sequence have to be processed while for 3D modeling often a sparse set of views is sufficient. Therefore, in this case features should be tracked from frame to frame. As already mentioned in Section 5.1 it is important that the structure is initialized from frames that are sufficiently separated. Another key component is the bundle adjustment. It does not only reduce the frame to frame jitter, but removes the largest part of the error that the structure and motion approach accumulates over the sequence. According to our experience it is very important to extend the perspective camera model with at least one parameter for radial distortion to obtain an undistorted metric structure (this will be clearly demonstrated in the example). Undistorted models are required to position larger virtual entities correctly in the model and to avoid drift of virtual objects in the augmented video sequences. Note however that for the rendering of the virtual objects the computed radial distortion can most often be ignored (except for sequences where radial distortion is immediately noticeable from single images).

examples A first set of experiments was carried out on video sequences of the *Béguinage* in Leuven (the same as in Figure 9.7). The sequence was recorded with a digital camcorder in progressive-scan mode to avoid interlacing problems. Once the structure and motion has been computed, the next step consists of positioning the virtual objects with respect to the real scene. This process is illustrated in Figure 8.15. The virtual objects are positioned within the computed 3D structure. To allow a precise positioning, feedback is immediately given by rendering the virtual object in some selected key-frames. After satisfactory placement of each single virtual object the computed camera corresponding to each image is used to render the virtual objects on top of the video. Anti-aliasing can be obtained by merging multiple views of the virtual objects obtained with a small offset on the principal point. Some frames of the *Béguinage* video sequence augmented with a cube are also shown in Figure 8.15.

Another example was recorded at Sagalassos in Turkey, where the footage of the ruins of an ancient fountain was taken. The *fountain* video sequence consists of 250 frames. A large part of the original monument is missing. Based on results of archaeological excavations and architectural studies, it was possible to generate a virtual copy of the missing part. Using the proposed approach the virtual reconstruction could be placed back on the remains of the original monument, at least in the recorded video sequence. This material is of great interest to the archaeologists, not only for education and dissemination, but also for fund raising to achieve a real restoration of the fountain. The top part of Figure 8.16 shows a top view of the recovered structure before and after bundle-adjustment. Besides the larger reconstruction error it can also be noticed that the non-refined structure is slightly bent. This effect mostly comes from not taking the radial distortion into account in the initial structure recovery. Therefore, a bundle adjustment that did not model radial distortion would not yield satisfying results. In the rest of Figure 8.16 some frames of the augmented video are shown.

8.4 Conclusion

In this chapter different methods were proposed to obtain 3D models from data computed as described in the previous chapters. The flexibility of the approach also allowed us to compute plenoptic models from image sequences acquired with a hand-held camera and to develop a flexible augmented reality system that can augment video seamlessly.
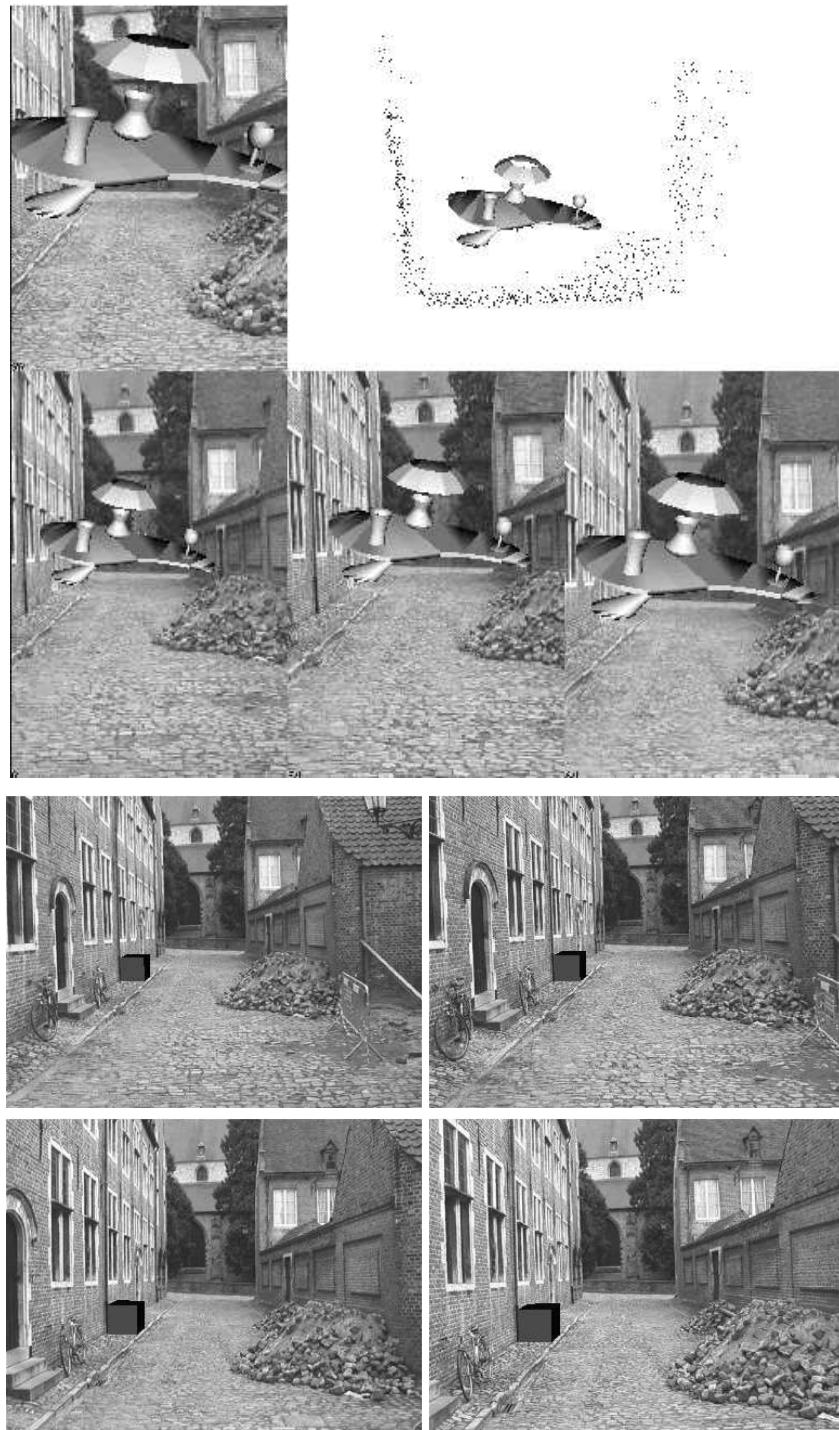
Figure 8.15: *Béguinage* sequence: positioning of virtual object (top), frames of video augmented with cube (bottom).
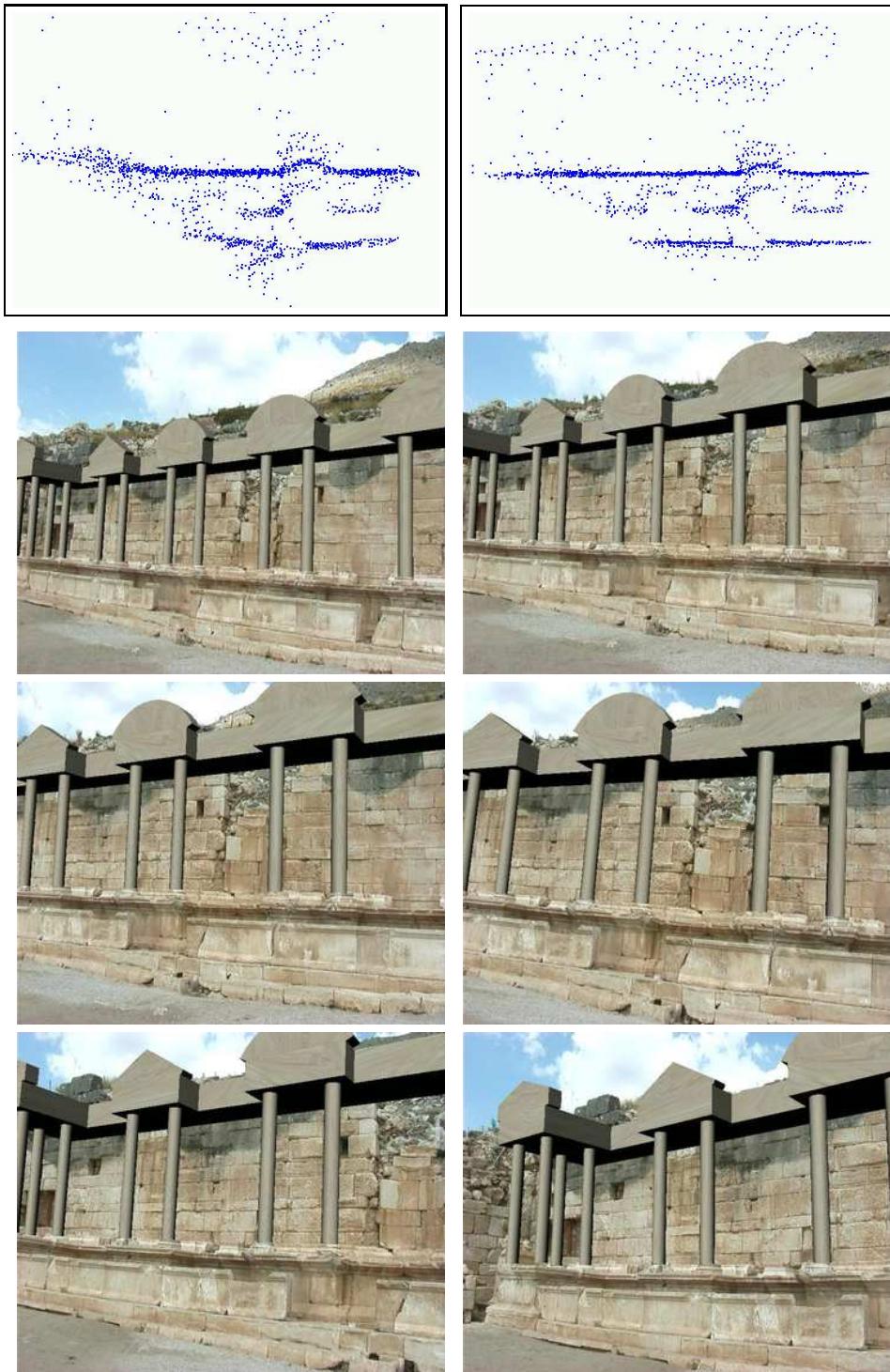
Figure 8.16: Fusion of real and virtual fountain parts. Top: structure-and-motion recovery before and after bundle adjustment. Bottom: 6 of the 250 frames of the fused video sequence

Chapter 9

Some results

In this chapter we will focus on the results obtained by the system described in the previous chapter. First some more results on 3D reconstruction from photographs are given. Then the flexibility of our approach is shown by reconstructing an amphitheater from old film footage. Finally several applications in archaeology are discussed. The application of our system to the construction of a virtual copy of the archaeological site of Sagalassos (Turkey) –a *virtualized* Sagalassos– is described. Some more specific applications in the field of archaeology are also discussed.

9.1   Acquisition of 3D models from photographs

The main application for our system is the generation of 3D models from images. One of the simplest methods to obtain a 3D model of a scene is therefore to use a photo camera and to shoot a few pictures of the scene from different viewpoints. Realistic 3D models can already be obtained with a restricted number of images. This is illustrated in this section with a detailed model of a part of a Jain temple in India.

A Jain Temple in Ranakpur

These images were taken during a tourist trip after ICCV'98 in India. A sequence of images was taken of a highly decorated part of one of the smaller Jain temples at Ranakpur, India. These images were taken with a standard Nikon F50 photo camera and then scanned. All the images which were used for the reconstruction can be seen in Figure 9.1. Figure 9.2 shows the reconstructed interest points together with the estimated pose and calibration of the camera for the different viewpoints. Note that only 5 images were used and that the global change in viewpoint between these different images is relatively small. In Figure 9.3 a global view of the reconstruction is given. In the lower part of the image the texture has been left out so that the recovered geometry is visible. Note the recovered shape of the statues and details of the temple wall. In Figure 9.4 two detail views from very different angles are given. The visual quality of these images is still very high. This shows that the recovered models allow to extrapolate viewpoints to some extent. Since it is difficult to give an impression of 3D shape through images we have put three views of the same part –but slightly rotated each time– in Figure 9.5. This reconstruction shows that the proposed approach is able to recover realistic 3D models of complex shapes. To achieve this no calibration nor prior knowledge about the scene was required.
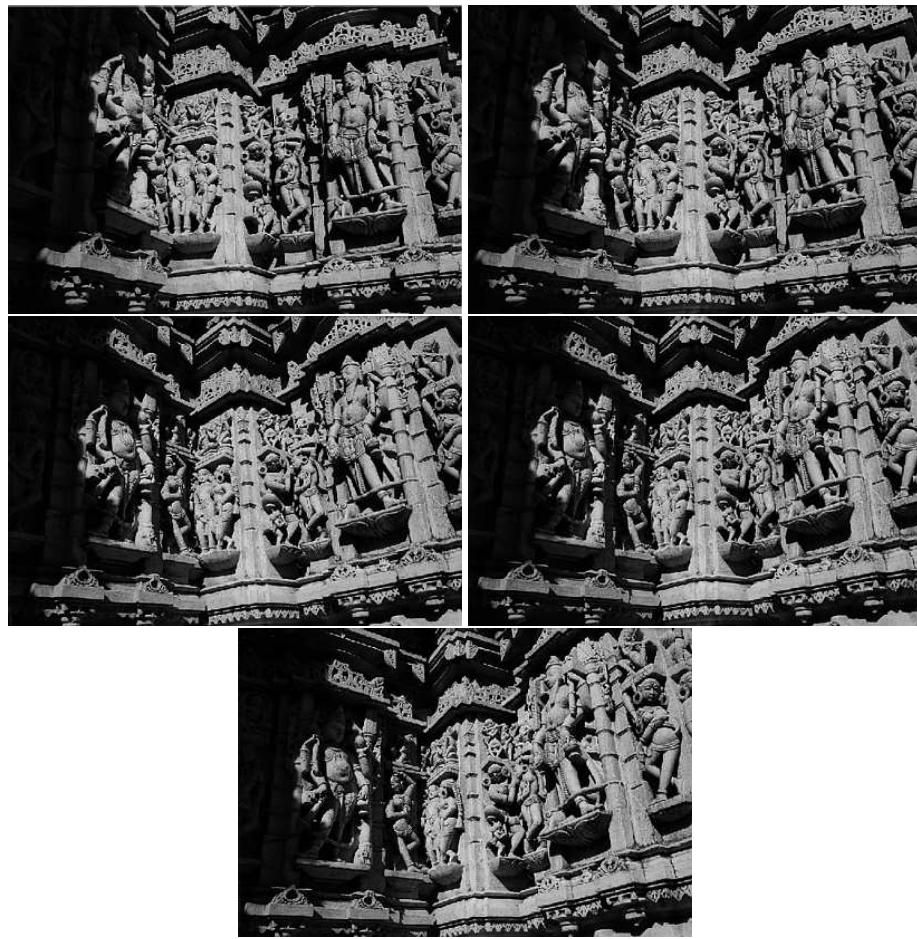
Figure 9.1: Photographs which were used to generate a 3D model of a detail of a Jain temple of Ranakpur.

Figure 9.2: Reconstruction of interest points and cameras. The system could automatically reconstruct a realistic 3D model of this complex scene without any additional information.
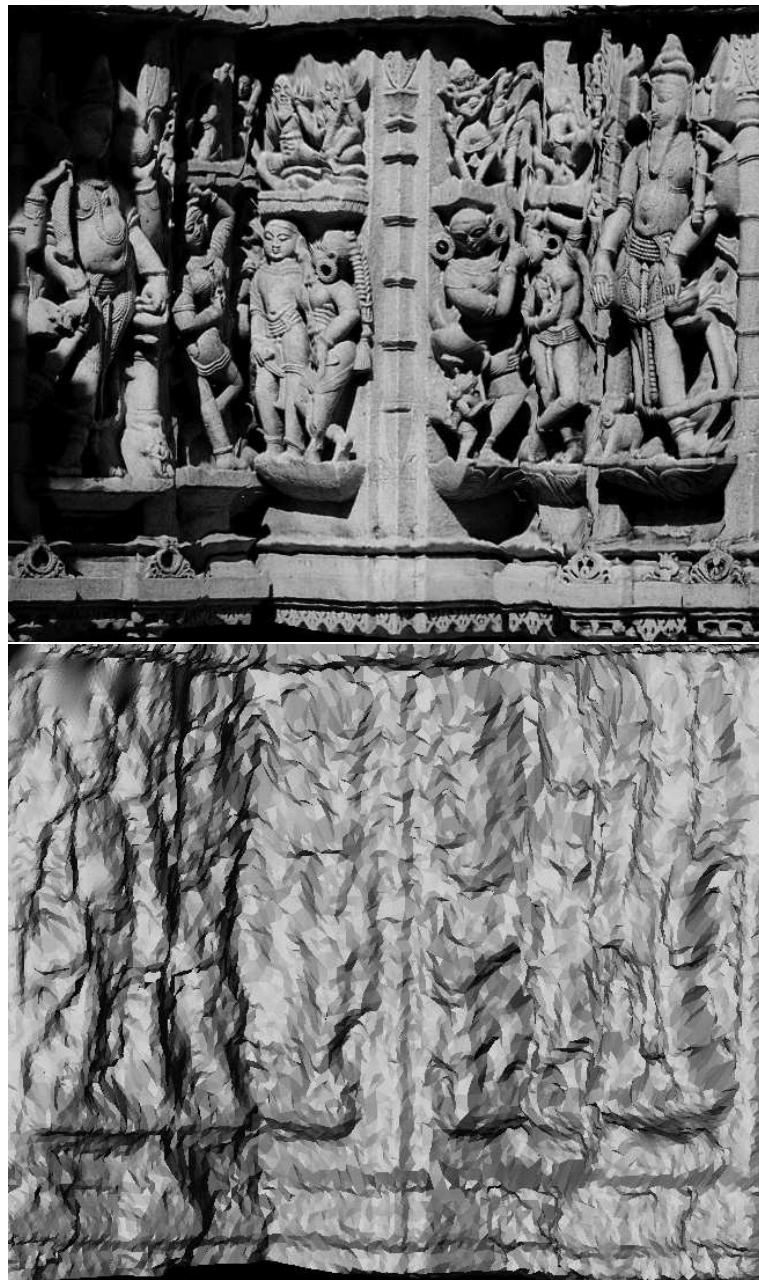
Figure 9.3: Reconstruction of a part of a Jain temple in Ranakpur (India). Both textured (top) and shaded (bottom) views are given to give an impression of the visual quality and the details of the recovered shape.

Figure 9.4: Two detail views of the reconstructed model.

Figure 9.5: Three rotated views of a detail of the reconstructed model.

Figure 9.6: View of the Béguinages of Leuven

The Béguinages of Leuven

Having university buildings on the UNESCO World Heritage list, we couldn't resist applying our 3D modeling techniques to it. In Figure 9.6 a view of the Béguinages of Leuven is given. Narrow streets are not very easy to model. Using the presented technique we were able to reconstruct 3D models from video sequences acquired with a digital video camera. This was only made possible through the use of the polar rectification since the epipoles were always located in the image. An example of a rectified image pair is given in Figure 9.7. Note that the top part of the rectified images correspond to the epipole. In Figure 9.8 three orthographic views of the reconstruction obtained from a single image pair are shown. These allow to verify the metric quality of the reconstruction (e.g. orthogonality and parallelism). To have a more complete model of the reconstructed street it is necessary to combine results from more than one image pair. This could for example be done using the volumetric approach presented in Section 8.1.2). A simpler approach consists of loading different surfaces at the same time in the visualization software. There is no need for registration since this was automatically performed during the structure and motion recovery. Figure 9.9 contains four views of a model consisting of 7 independently reconstructed 3D surfaces.
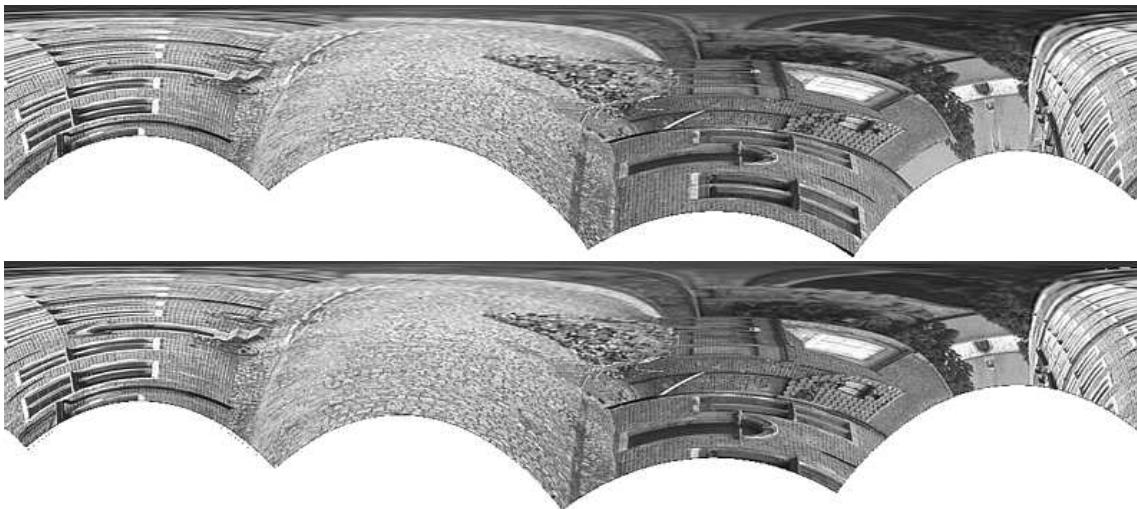

Figure 9.7: Rectified image pair (corresponding pixels are vertically aligned).
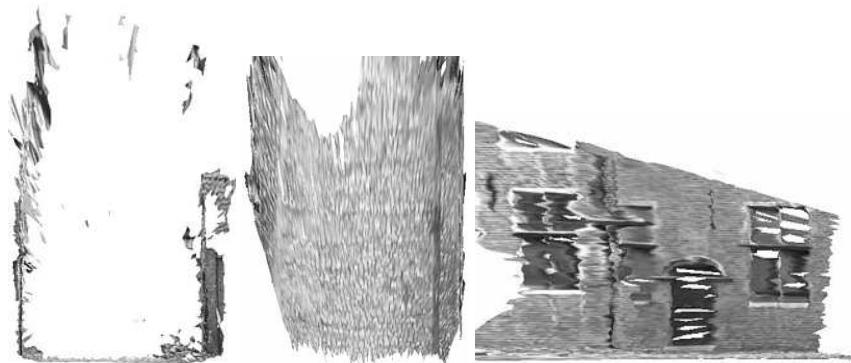
Figure 9.8: Orthographic views of a reconstruction obtained from a single image pair: front (left), top (middle) and side (right).
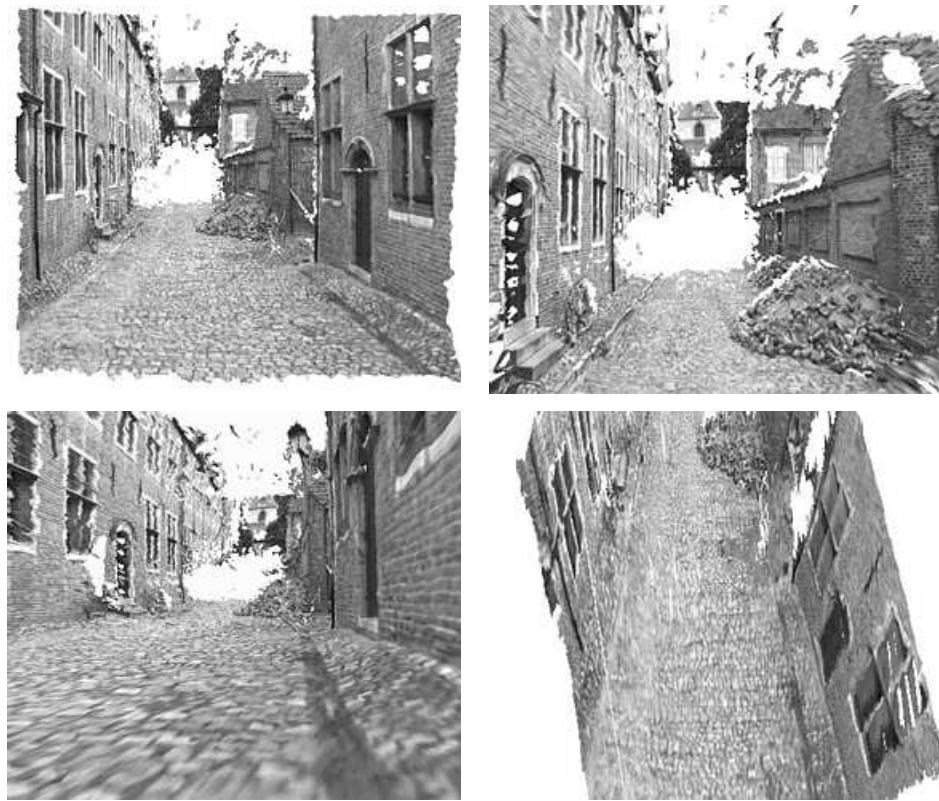

Figure 9.9: Views of a reconstruction obtained by combining results from more images.
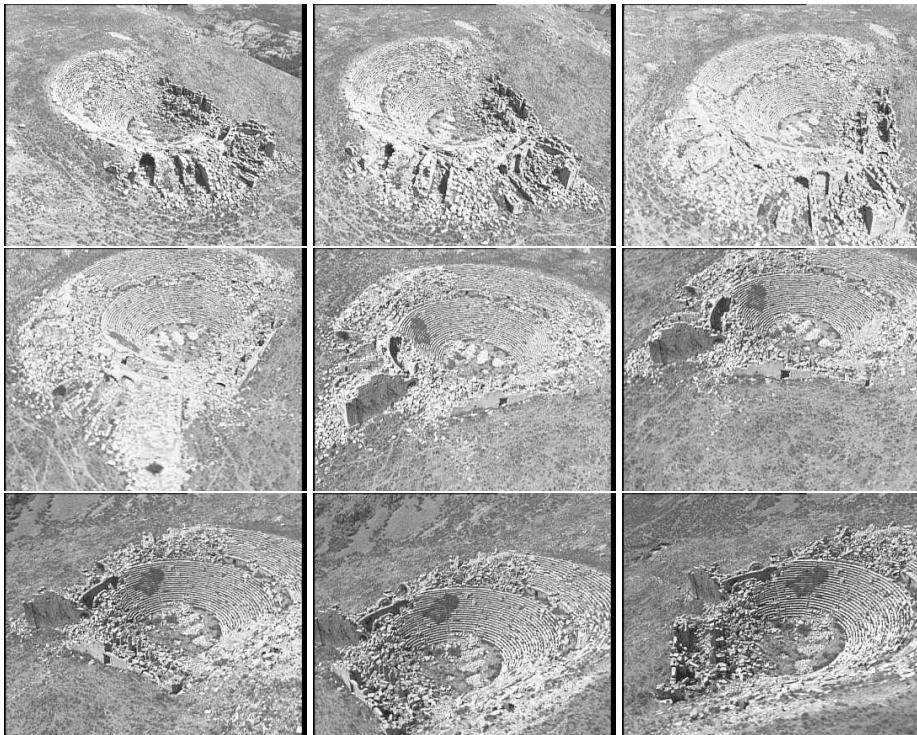
Figure 9.10: This sequence was filmed from a helicopter in 1990 by a cameraman of the belgian television to illustrate a TV program on Sagalassos (an archaeological site in Turkey).

9.2 Acquisition of 3D models from pre-existing image sequences

Here the reconstruction of the ancient theater of Sagalassos is shown. Sagalassos is an archaeological site in Turkey. More results obtained at this site are presented in Sections 9.3 and 9.4. The reconstruction is based on a sequence filmed by a cameraman from the BRTN (Belgische Radio en Televisie van de Neder-landstalige gemeenschap) in 1990. The sequence was filmed to illustrate a TV program about Sagalassos. Because of the motion only fields –and not frames– could be used. The resolution of the images we could use was thus restricted to 768×288. The sequence consisted of about hundred images, every tenth image is shown in Figure 9.10. We recorded approximately 3 images per second.

In Figure 9.11 the reconstruction of interest points and cameras is given. This shows that the approach can deal with long image sequences.

Dense depth maps were generated from this sequence and a dense textured 3D surface model was
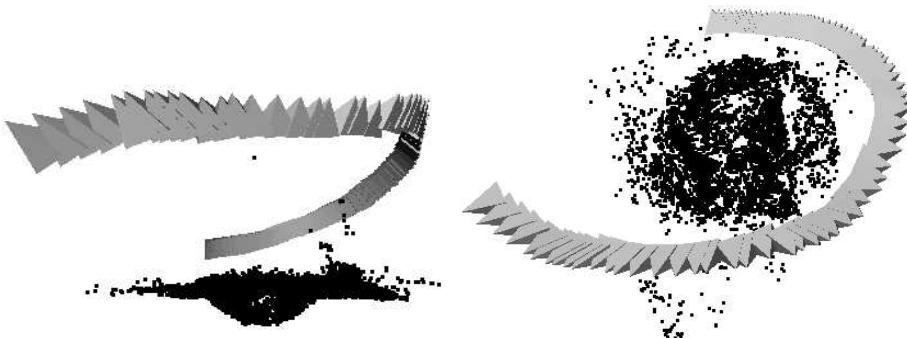

Figure 9.11: The reconstructed interest points and camera poses recovered from the TV sequence.
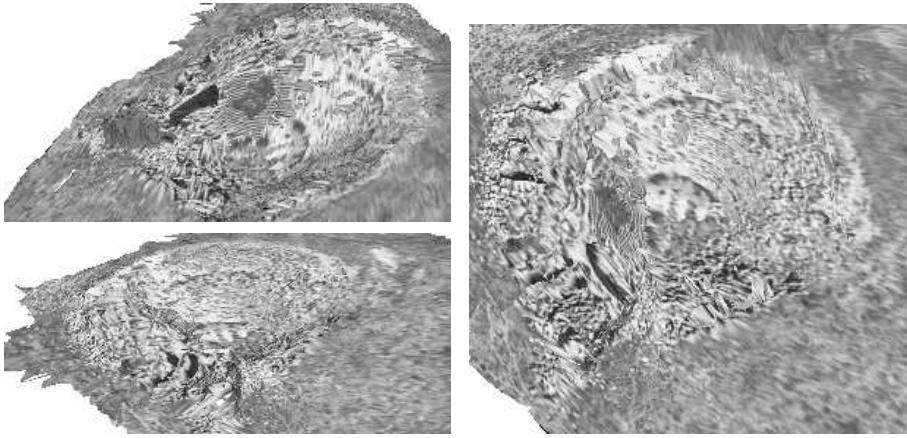
Figure 9.12: Some views of the reconstructed model of the ancient theater of Sagalassos.

constructed from this. Some views of this model are given in Figure 9.12.

9.3   Virtualizing archaeological sites

Virtual reality is a technology that offers promising perspectives for archaeologists. It can help in many ways. New insights can be gained by immersion in ancient worlds, unaccessible sites can be made available to a global public, courses can be given "on-site" and different periods or building phases can coexist.

One of the main problems however is the generation of these virtual worlds. They require a huge amount of on-site measurements. In addition the whole site has to be reproduced manually with a CAD- or 3D modeling system. This requires a lot of time. Moreover it is difficult to model complex shapes and to take all the details into account. Obtaining realistic surface texture is also a critical issue. As a result walls are often approximated by planar surfaces, stones often all get the same texture, statues are only crudely modeled, small details are left out, etc.

An alternative approach consists of using images of the site. Some software tools exist, but require a lot of human interaction [97] or preliminary models [22]. Our system offers unique features in this context. The flexibility of acquisition can be very important for field measurements which are often required on archaeological sites. The fact that a simple photo camera can be sufficient for acquisition is an important advantage compared to methods based on theodolites or other expensive hardware. Especially in demanding weather conditions (e.g. dust, wind, heat, humidity).

The ancient site of Sagalassos (south-west Turkey) was used as a test case to illustrate the potential of the approach developed in this work. The images were obtained with a consumer photo camera (digitized on photoCD) and with a consumer digital video camera.

9.3.1   Virtualizing scenes

The 3D surface acquisition technique that we have developed can be applied readily to archaeological sites. The on-site acquisition procedure consists of recording an image sequence of the scene that one desires to *virtualize*. To allow for the algorithms to yield good results viewpoint changes between consecutive images should not exceed 5 to 10 degrees. An example of such a sequence is given in Figure 9.13. The result for the image sequence under consideration can be seen in Figure 9.14. An important advantage is that details like missing stones, not perfectly planar walls or symmetric structures are preserved. In addition the surface texture is directly extracted from the images. This does not only result in a much higher degree of realism, but is also important for the authenticity of the reconstruction. Therefore the reconstructions obtained with this system could also be used as a scale model on which measurements can be carried out or as a tool for planning restorations.

Figure 9.13: Image sequence which was used to build a 3D model of the corner of the Roman baths
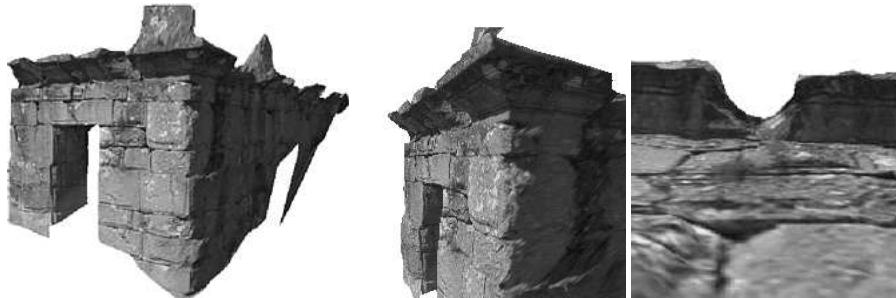

Figure 9.14: Virtualized corner of the Roman baths, on the right some details are shown

As a second example, the reconstruction of the remains of an ancient fountain is shown. In Figure 9.15 three of the six images used for the reconstruction are shown. All images were taken from the same ground level. They were acquired with a digital camera with a resolution of approximately 1500x1000. Half resolution images were used for the computation of the shape. The texture was generated from the full resolution images.

The reconstruction can be seen in Figure 9.16, the left side shows a view with texture, the right view gives a shaded view of the model without texture. In Figure 9.17 two close-up shots of the model are shown.

9.3.2   Reconstructing an overview model

A first approach to obtain a virtual reality model for a whole site consists of taking a few overview photographs from the distance. Since our technique is independent of scale this yields an overview model of the whole site. The only difference with the modeling of smaller objects is the distance needed between two camera poses. For most active techniques it is impossible to cope with scenes of this size. The use of a stereo rig would also be very hard since a baseline of several tens of meters would be required. Therefore one of the promising applications of the proposed technique is large scale terrain modeling.

In Figure 9.18, 3 of the 9 images taken from a hillside near the excavation site are shown. These were used to generate the 3D surface model seen in Figure 9.19. In addition one can see from the right side of this figure that this model could be used to generate a Digital Terrain Map or an orthomap at low cost. In this case only 3 reference measurements –GPS and altitude– are necessary to localize and orient the model in the world reference frame.
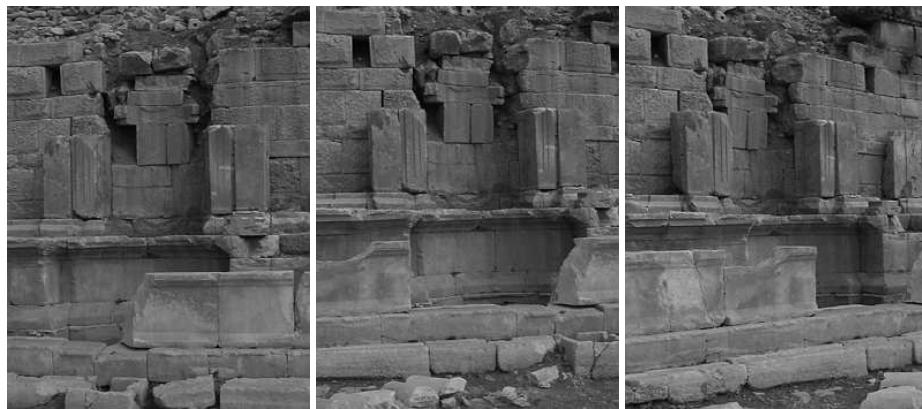
Figure 9.15: Three of the six images of the Fountain sequence
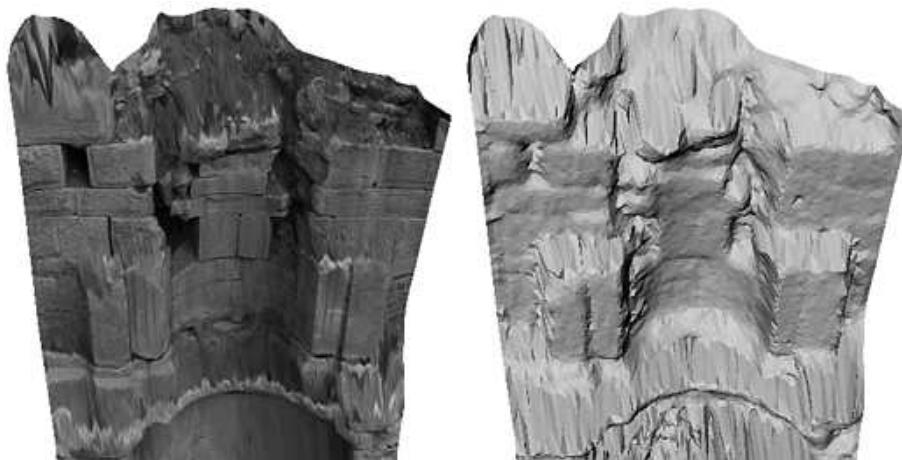

Figure 9.16: Perspective views of the reconstructed fountain with and without texture

Figure 9.17: Close-up views of some details of the reconstructed fountain

Figure 9.18: Some of the images of the Sagalassos Site sequence

Figure 9.19: Perspective views of the 3D reconstruction of the Sagalassos site (left). Top view of the reconstruction of the Sagalassos site (right).
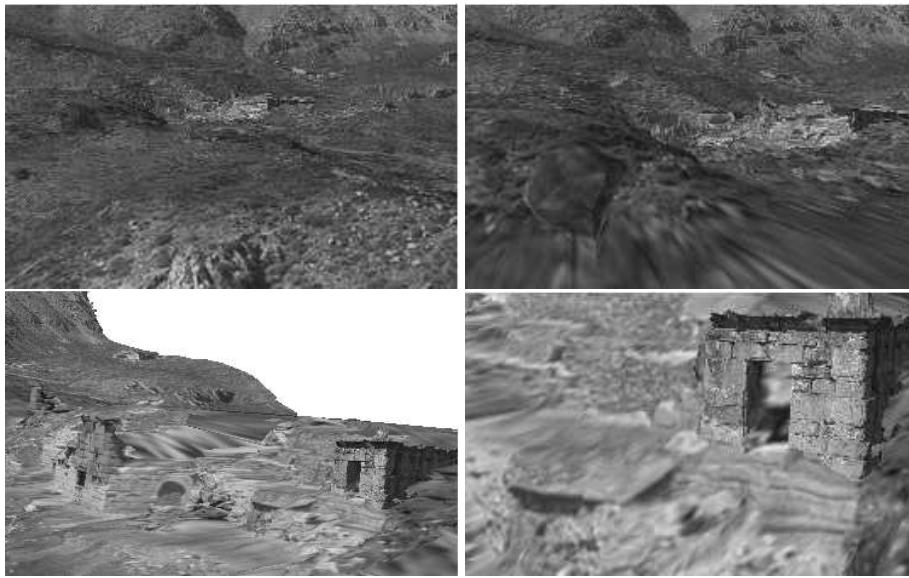
Figure 9.20: Integration of models of different scales: site of Sagalassos, Roman baths and corner of the Roman baths.

9.3.3 Reconstructions at different scales

The problem is that this kind of overview model is too coarse to be used for realistic walk-throughs around the site or for looking at specific monuments. Therefore it is necessary to integrate more detailed models into this overview model. This can be done by taking additional image sequences for all the interesting areas on the site. These are used to generate reconstructions of the site at different scales, going from a global reconstruction of the whole site to a detailed reconstruction for every monument.

These reconstructions thus naturally fill in the different levels of details which should be provided for optimal rendering. In Figure 9.20 an integrated reconstruction containing reconstructions at three different scales can be seen.

At this point the integration was done by interactively positioning the local reconstructions in the global 3D model. This is a cumbersome procedure since the 7 degrees of freedom of the similarity ambiguity have to be taken into account. Researchers are working on methods to automate this. Two different approaches are possible. The first approach is based on matching features which are based on both photometric and geometric properties, the second on minimizing a global alignment measure. A combination of both approaches will probably yield the best results.

9.4   More applications in archaeology

Since these 3D models can be generated automatically and the on-site acquisition time is very short, several new applications come to mind. In this section a few possibilities are illustrated.

9.4.1   3D stratigraphy

Archaeology is one of the sciences were annotations and precise documentation are most important because evidence is destroyed during work. An important aspect of this is the stratigraphy. This reflects the different layers of soil that correspond to different time periods in an excavated sector. Due to practical limitations this stratigraphy is often only recorded for some slices, not for the whole sector.

Our technique allows a more optimal approach. For every layer a complete 3D model of the excavated sector can be generated. Since this only involves taking a series of pictures this does not slow down the

Figure 9.21: 3D stratigraphy, the excavation of a Roman villa at two different moments.
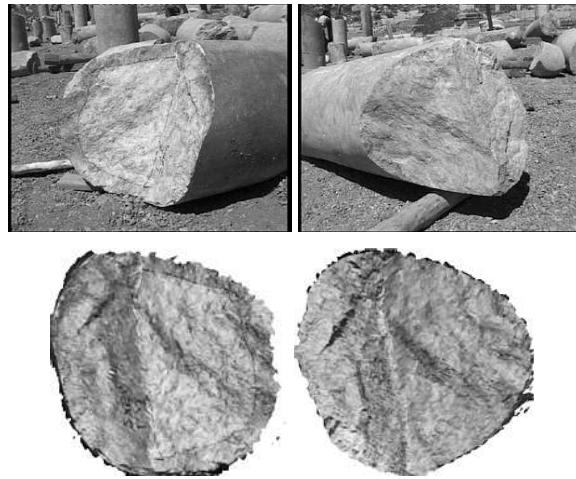
Figure 9.22: Two images of parts of broken pillars (top) and two orthographic views of the matching surfaces generated from the 3D models (bottom)

progress of the archaeological work. In addition it is possible to model artifacts separately which are found in these layers and to include the models in the final 3D stratigraphy.

This concept is illustrated in Figure 9.21. The excavations of an ancient Roman villa at Sagalassos were recorded with our technique. In the figure a view of the 3D model of the excavation is provided for two different layers.

9.4.2 Generating and testing building hypotheses

The technique also has a lot to offer for generating and testing building hypotheses. Due to the ease of acquisition and the obtained level of detail, one could reconstruct every building block separately. The different construction hypotheses can then interactively be verified on a virtual building site. Some testing could even be automated.

The matching of the two parts of Figure 9.22 for example could be verified through a standard registration algorithm [13]. An automatic procedure can be important when dozens of broken parts have to be matched against each other.

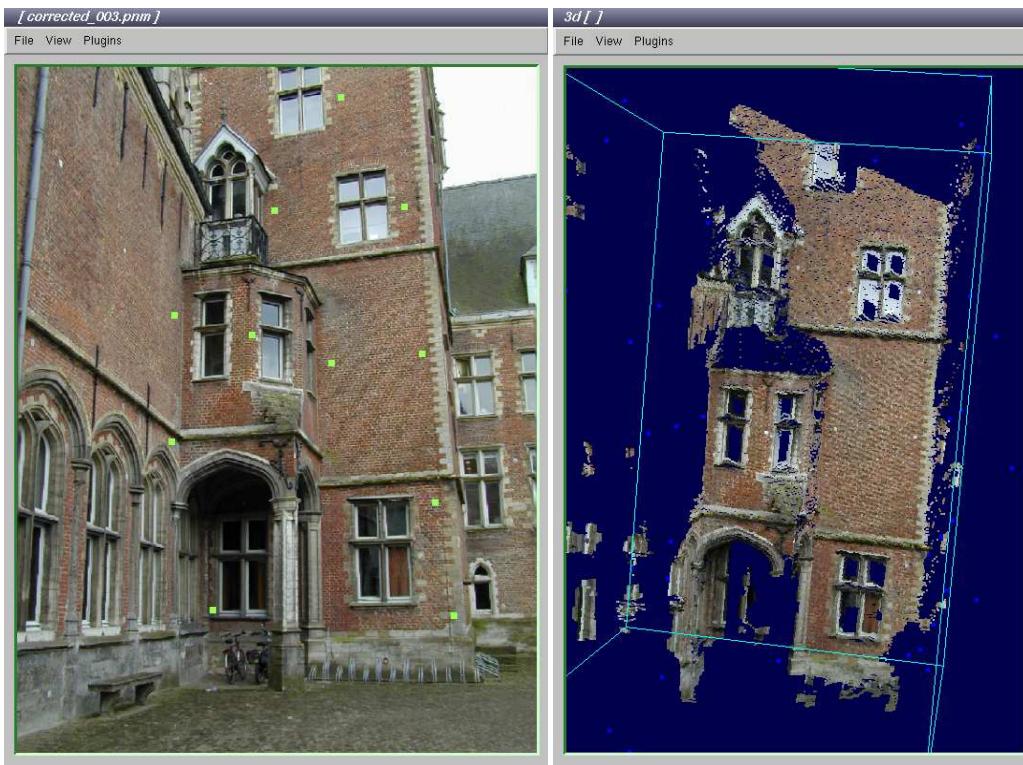Figure 9.23: On the left one of the input images can be seen with the measured reference points superimposed. On the right the cloud of reconstructed points is shown.

9.5   Architecture and heritage conservation

At this moment many architects involved in conservation still work in the traditional way. They use hand-measured (tapes, plumb-bobs, levels...) or instrument based (theodolite, total station, photogrammetry) survey methods. This information is usually transferred to 2D paper drawings: plans, sections and facades. The main drawback of this approach is that all information is distributed in different types of documents (2D drawings, texts, photographs...). This makes it often very difficult for policy makers, engineers or others persons involved in one particular phase of the process, to get a complete and unambiguous overview of the available information. In addition, it is very difficult to use this material for exchange with other architects or researchers (for comparative studies...) or for distribution to the public (publications in periodicals, tourist information...).

As many architects are shifting towards computer-aided design for new buildings, they also try to apply these programs to renovation or conservation projects. However, the number of tools available to accomplish the task of 'getting the existing building in the CAD program' is limited, and mainly directed to 'translate' traditional methods to CAD (automatic import of full station co-ordinates, error-adjustment of triangulation...). Based on a limited number of actually measured points, 2D plans and sections or a 3D model can be constructed. This typically results in a very 'simplified' representation of the building, which is absolutely not in line with the high requirements for conservation purposes.

The technology presented in these notes can be very useful in this context. We have an ongoing project with architects that aims at developing a technology that enables an operator to build up an accurate three dimensional model - without too much repetitive work - starting from photos of the objects. For a number of reasons, such as the need for absolute coordinates, the choice was made to also measure reference points using a theodolyte. This allows to simplify a number of calibration issues. In Figure 9.23 an example is shown. A more detailed description of this project can be found in [92].

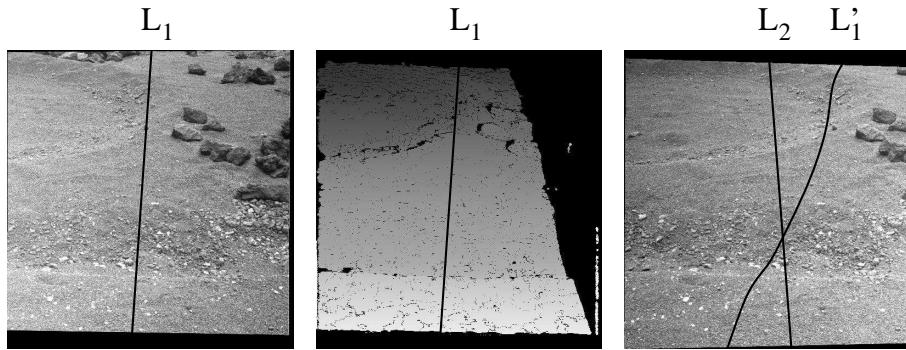$$L_1 \qquad\qquad L_1 \qquad\qquad L_2 \quad L_1'$$

Figure 9.24: Digital Elevation Map generation in detail. The figure contains the left rectified image (left), the corresponding disparity map (middle) and the right rectified image (right). The 3D line L corresponding to a point of the DEM projects to L_1 resp. L_2. Through the disparity map the *shadow* of L_1 on the right image can also be computed. The intersection point of L_2 and L_1 corresponds to the point where L intersects the surface.

Figure 9.25: Mosaic of images of the testbed taken by the stereo head.

9.6   Planetary rover control

In this section a system is presented that we developed for ESA for the support of planetary exploration. More information can be found in [167, 168, 169]. The system that is send to the planetary surface consists of a rover and lander. The lander has a stereo head equipped with a pan-tilt mechanism. This vision system is used both for modeling of the terrain and for localization of the rover. Both tasks are necessary for the navigation of the rover. Due to the stress that occurs during the flight a recalibration of the stereo vision system is required once it is deployed on the planet. Due to practical limitations it is infeasible to use a known calibration pattern for this purpose and therefore a new calibration procedure had to be developed that can work on images of the planetary environment. This automatic procedure recovers the relative orientation of the cameras and the pan- and tilt-axis, besides the exterior orientation for all the images. The same images are subsequently used to recover the 3D structure of the terrain. For this purpose a dense stereo matching algorithm is used that -after rectification- computes a disparity map. Finally, all the disparity maps are merged into a single digital terrain model. This procedure is illustrated in Figure 9.24. The fact that the same images can be used for both calibration and 3D reconstruction is important since in general the communication bandwidth is very limited. In addition to the use for navigation and path planning, the 3D model of the terrain is also used for Virtual Reality simulation of the mission, in which case the model is texture mapped with the original images. A first test of the complete system was performed at the ESA-ESTEC test facilities in Noordwijk (The Netherlands) where access to a planetary testbed of about 7 by 7 meters was available. The Imaging Head was set up next to the testbed. Its first task was the recording of the terrain. A mosaic of the pictures taken by this process can be seen in figure 9.25. The autonomous calibration procedure was launched and it computed the extrinsic calibration of the cameras based on the images. Once the calibration had been computed the system rectified the images and computed dense disparity maps. Based on these, a Digital Elevation Map was constructed. The result can be seen in
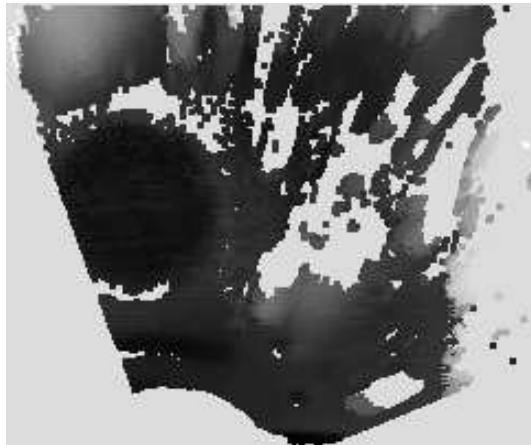
Figure 9.26: Digital Elevation Map of the ESTEC planetary testbed. A significant amount of cells is not filled in because they are located in occluded areas.

figure 9.26. Because of the relatively low height of the Imaging Head (approximately 1.5 meters above the testbed) and the big rocks in the testbed, a large portion of the Digital Elevation Map could not be filled in because of occlusions.

It can be expected that global terrain model will not be sufficiently accurate to achieve fine steering of the rover for some specific operations. An important task of the rover is to carry out surface measurements on interesting rocks. For this purpose it is important to be able to very accurately position the measurement device. Since probably the rover will also be equipped with a (single) camera the technique proposed in this text could be used to reconstruct a local model of an interesting rock while the rover is approaching it. A preliminary test was carried out on a rock in the ESA testbed. Some results can be seen in Figure 9.27.

9.7   Conclusion

The flexibility of the proposed systems allows applications in many domains. In some cases further developments would be required to do so, in others the system (or parts of it) could just be used as is. Some interesting areas are forensics (e.g. crime scene reconstruction), robotics (e.g. autonomous guided vehicles), augmented reality (e.g. camera tracking) or post-production (e.g. generation of virtual sets).

In this chapter some results were presented in more detail to illustrate the possibilities of this work. It was shown that realistic 3D models of existing monuments could be obtained automatically from a few photographs. The flexibility of the technique allows it to be used on existing photo or video material. This was illustrated through the reconstruction of an ancient theater from a video extracted from the archives of the Belgian television.

The archaeological site of Sagalassos (Turkey) was used as a test case for our system. Several parts of the site were modeled. Since our approach is independent of scale it was also used to obtain a 3D model of the whole site at once. Some potential applications are also illustrated, i.e. 3D stratigraphy and generating/testing building hypotheses. A few other possible applications were also briefly discussed.
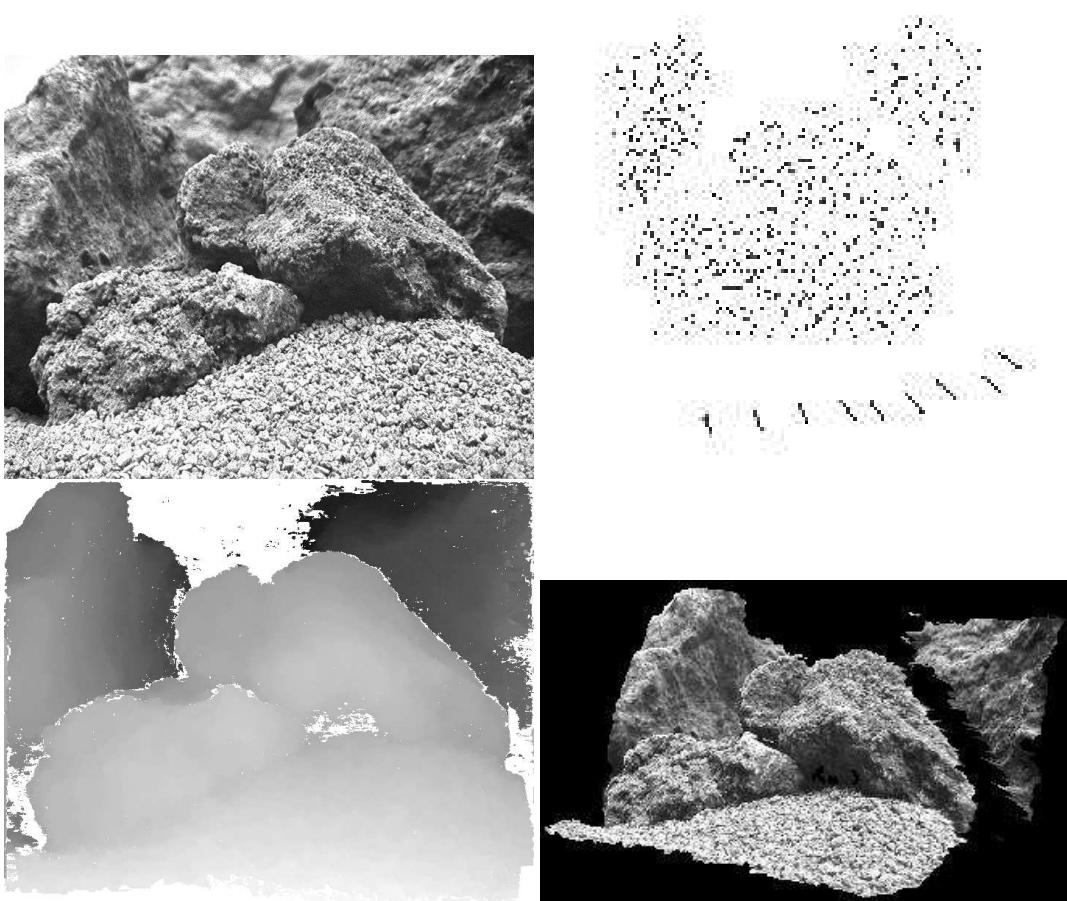
Figure 9.27: Different steps of the 3D reconstruction of a rock from images.

Appendix A

Bundle adjustment

Once the structure and motion has been obtained for the whole sequence, it is recommended to refine it through a global minimization step. A maximum likelihood estimation can be obtained through *bundle adjustment* [12, 137]. The goal is to find the projection matrices $\hat{\mathbf{P}}_k$ and the 3D points $\hat{\mathtt{M}}_i$ for which the mean squared distances between the observed image points $\mathtt{m}_{ki}$ and the reprojected image points $\hat{\mathtt{m}}_{ki}$ is minimized. For m views and n points the following criterion should be minimized:

$$\min_{\hat{\mathbf{P}}_k, \hat{\mathtt{M}}_i} \sum_{k=1}^{m} \sum_{i=1}^{n} D\big(\mathtt{m}_{ki}, \hat{\mathbf{P}}_k \hat{\mathtt{M}}_i\big)^2 \tag{A.1}$$

where $D(\hat{\mathtt{m}}, \mathtt{m})$ is the Euclidean image distance. If the image error is zero-mean Gaussian then bundle adjustment is the Maximum Likelihood Estimator. Although it can be expressed very simply, this minimization problem is huge. For a typical sequence of 20 views and 2000 points, a minimization problem in more than 6000 variables has to be solved. A straight-forward computation is obviously not feasible. However, the special structure of the problem can be exploited to solve the problem much more efficiently. The observed points $\mathtt{m}_{ki}$ being fixed, a specific residual $r_{ki} = D\big(\mathtt{m}_{ki}, \mathbf{P}_k \mathtt{M}_i\big)^2$ is only dependent on the point i-th point and the k-th camera view. This results in a sparse structure for the normal equations. Using this structure the points $\mathtt{M}_i$ can be eliminated from the equations, yielding a much smaller but denser problem. Views that have features in common are now related. For a long sequence where features tend to only be seen in a few consecutive views, the matrix that has to be solved is still sparse (typically band diagonal). In this vcase it can be very interesting to make use of sparse linear algebra algorithms, e.g. [3].

Before going more into detail on efficiently solving the bundle adjustment, the Levenberg-Marquardt minimization is presented. Based one this an efficient method for bundle adjustment will be proposed in Section A.2.

A.1   Levenberg-Marquardt minimization

Given a vector relation $\mathtt{y} = f(\mathtt{x})$ where $\mathtt{x}$ and $\mathtt{y}$ can have different dimensions and an observation $\hat{\mathtt{y}}$, we want to find the vector $\mathtt{x}$ which best satisfies the given relation. More precisely, we are looking for the vector $\hat{\mathtt{x}}$ satisfying $\hat{\mathtt{y}} = f(\hat{\mathtt{x}}) + \hat{\mathtt{e}}$ for which $\|\hat{\mathtt{e}}\|$ is minimal.

A.1.1   Newton iteration

Newton's approach starts from an initial value $\mathtt{x}_0$ and refines this value using the assumption that f is locally linear. A first order approximation of $f(\mathtt{x}_0 + \Delta)$ yields:

$$f(\mathtt{x}_0 + \Delta) = f(\mathtt{x}_0) + \mathbf{J}\Delta \tag{A.2}$$

with $\mathbf{J}$ the Jacobian matrix and Δ a small displacement. Under these assumptions minimizing $\hat{\mathtt{e}} = \hat{\mathtt{e}}_0 - \mathbf{J}\Delta$ can be solved through linear least-squares. A simple derivation yields

$$\mathbf{J}^{\top}\mathbf{J}\Delta = \mathbf{J}^{\top}\hat{\mathtt{e}} \ . \tag{A.3}$$

$$\mathbf{P}_1 \qquad \mathbf{P}_2 \qquad \mathbf{P}_3 \qquad \mathtt{M}_1\ \mathtt{M}_2\ \mathtt{M}_3\ \mathtt{M}_4$$
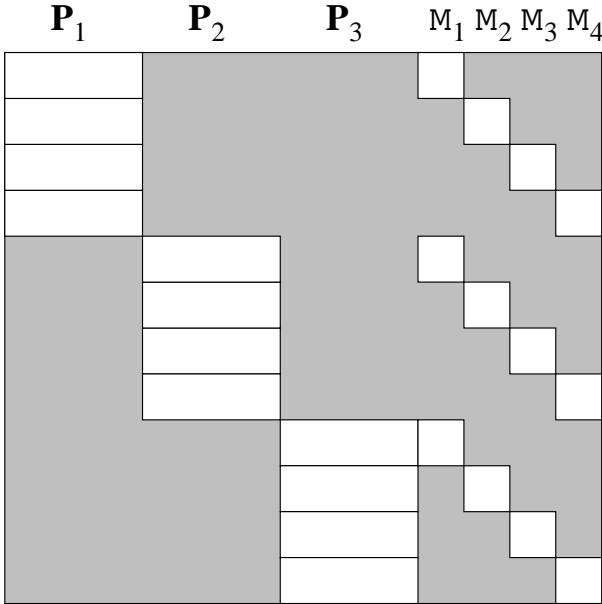

Figure A.1: Sparse structure of Jacobian for bundle adjustment.

This equation is called the normal equation. The solution to the problem is found by starting from an initial solution and refining it based on successive iterations

$$\mathtt{x}_{i+1} = \mathtt{x}_i + \Delta_i \tag{A.4}$$

with Δ_i the solution of the normal equation A.3 evaluated at $\mathtt{x}_i$. One hopes that this algorithm will converge to the desired solution, but it could also end up in a local minimum or not converge at all. This depends a lot on the initial value $\mathtt{x}_0$.

A.1.2   Levenberg-Marquardt iteration

The levenberg-Marquardt iteration is a variation on the Newton iteration. The normal equations $\mathbf{N}\Delta = \mathbf{J}^\top \mathbf{J}\Delta = \mathbf{J}^\top \hat{\mathtt{e}}$ are augmented to $\mathbf{N}'\Delta = \mathbf{J}^\top \hat{\mathtt{e}}$ where $N'_{ij} = (1 + \delta_{ij}\lambda)N_{ij}$ with δ_{ij} the Kronecker delta.

The value λ is initialized to a small value, e.g. 10^{-3}. If the value obtained for Δ reduces the error, the increment is accepted and λ is divided by 10 before the next iteration. On the other hand, if the error increases then λ is multiplied by 10 and the augmented normal equations are solved again, until an increment is obtained that reduces the error. This is bound to happen, since for a large λ the method approaches a steepest descent.

A.2   Bundle adjustment

The observed points $\mathtt{m}_{ki}$ being fixed, a specific residual $r_{ki} = D(\mathtt{m}_{ki}, \hat{\mathbf{P}}_k\hat{\mathtt{M}}_i)^2$ is only dependent on the point i-th point and the k-th projection matrix. This results in a very sparse matrix for the Jacobian. This is illustrated in figure A.1 for 3 views and 4 points. Because of the block structure of the Jacobian solving the normal equations $\mathbf{J}^\top \mathbf{J}\mathbf{x} = \mathbf{b}$ have a structure as seen in figure A.2. It is possible to write down explicit formulas for each block. Let us first introduce the following notation:

$$\mathbf{U}_k = \sum_i \left(\frac{\partial \hat{\mathtt{m}}_{ki}}{\partial \hat{\mathbf{P}}_k}\right)^\top \frac{\partial \hat{\mathtt{m}}_{ki}}{\partial \hat{\mathbf{P}}_k} \tag{A.5}$$

$$\mathbf{V}_i = \sum_k \left(\frac{\partial \hat{\mathtt{m}}_{ki}}{\partial \hat{\mathtt{M}}_i}\right)^\top \frac{\partial \hat{\mathtt{m}}_{ki}}{\partial \hat{\mathtt{M}}_i} \tag{A.6}$$
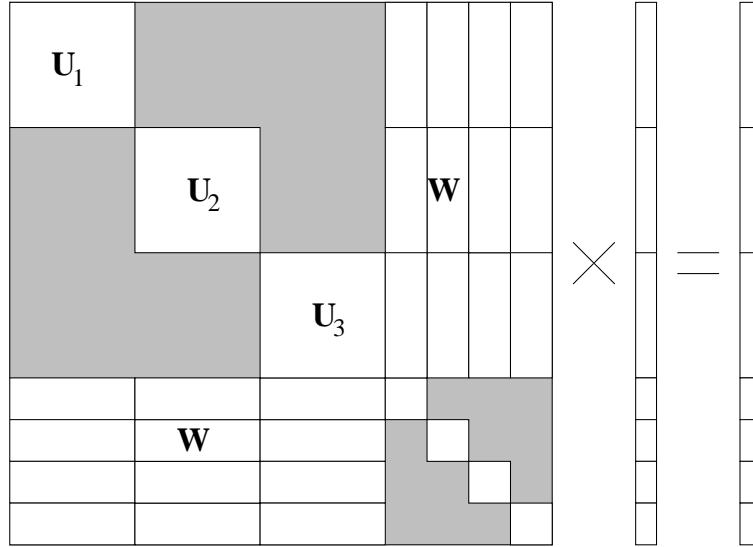
Figure A.2: Block structure of normal equations.

$$\mathbf{W}_{ki} \quad = \quad (\frac{\partial \hat{\mathbf{m}}_{ki}}{\partial \hat{\mathbf{P}}_k})^\top \frac{\partial \hat{\mathbf{m}}_{ki}}{\partial \hat{\mathbf{M}}_i} \tag{A.7}$$

$$\epsilon(\mathbf{P}_k) \quad = \quad \sum_i (\frac{\partial \hat{\mathbf{m}}_{ki}}{\partial \hat{\mathbf{P}}_k})^\top \epsilon_{ki} \tag{A.8}$$

$$\epsilon(\mathbf{M}_i) \quad = \quad \sum_i (\frac{\partial \hat{\mathbf{m}}_{ki}}{\partial \hat{\mathbf{M}}_i})^\top \epsilon_{ki} \tag{A.9}$$

with $\frac{\partial \hat{\mathbf{m}}_{ki}}{\partial \hat{\mathbf{P}}_k}$ and $\frac{\partial \hat{\mathbf{m}}_{ki}}{\partial \hat{\mathbf{M}}_i}$ matrices containing the partial derivatives from the coordinates of $\hat{\mathbf{m}}_{ki}$ to the parameters of $\hat{\mathbf{P}}_k$ and $\hat{\mathbf{M}}_i$ respectively. In this case the normal equations can be rewritten as

$$\begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^\top & \mathbf{V} \end{bmatrix} \begin{bmatrix} \Delta(\mathbf{P}) \\ \Delta(\mathbf{M}) \end{bmatrix} = \begin{bmatrix} \epsilon(\mathbf{P}) \\ \epsilon(\mathbf{M}) \end{bmatrix} \tag{A.10}$$

where the matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}, \Delta(\mathbf{P}), \Delta(\mathbf{M}), \epsilon(\mathbf{P})$ and $\epsilon(\mathbf{M})$ are composed of the blocks defined previously. Assuming $\mathbf{V}$ is invertible both sides of equation A.10 multiplied on the left with

$$\begin{bmatrix} \mathbf{I} & -\mathbf{W}\mathbf{V}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

to obtain

$$\begin{bmatrix} \mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^\top & \mathbf{0} \\ \mathbf{W}^\top & \mathbf{V} \end{bmatrix} \begin{bmatrix} \Delta(\mathbf{P}) \\ \Delta(\mathbf{M}) \end{bmatrix} = \begin{bmatrix} \epsilon(\mathbf{P}) - \mathbf{W}\mathbf{V}^{-1}\epsilon(\mathbf{M}) \\ \epsilon(\mathbf{M}) \end{bmatrix} \tag{A.11}$$

This can be separated in two groups of equations. The first one is

$$\left(\mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^\top\right) \Delta(\mathbf{P}) = \epsilon(\mathbf{P}) - \mathbf{W}\mathbf{V}^{-1}\epsilon(\mathbf{M}) \tag{A.12}$$

and can be used to solve for $\Delta(\mathbf{P})$. The solution can be substituted in the other group of equations:

$$\Delta(\mathbf{M}) = \mathbf{V}^{-1}\left(\epsilon(\mathbf{M}) - \mathbf{W}^\top\Delta(\mathbf{P})\right) \tag{A.13}$$

Note that due to the sparse block structure of $\mathbf{V}$ its inverse can be computed very efficiently.

The only computationally expensive step consist of solving equation A.12. This is however much smaller than the original problem. For 20 views and 2000 points the problem is reduced from solving 6000 unknowns concurrently to more or less 200 unknowns. To simplify the notations the normal equations were used in this presentation. It is however simple to extend this to the augmented normal equations.

Bibliography

[1] H. Akaike, "A new look at the statistical model identification", *IEEE Trans. on Automatic Control*, 19-6, pp 716-723, 1974.

[2] M. Armstrong, A. Zisserman and R. Hartley, "Euclidean Reconstruction from Image Triplets", *Computer Vision - ECCV'96*, Lecture Notes in Computer Science, Vol. 1064, Springer-Verlag, pp. 3-16, 1996.

[3] C. Ashcraft and R. Grimes. "SPOOLES: An object-oriented sparse matrix library". In Proceedings of the 9th SIAM Conference on Parallel Processing for Scientific Computing, San-Antonio, Texas, 1999. 10 pages on CD-ROM.

[4] N. Ayache and C. Hansen, "Rectification of images for binocular and trinocular stereovision", *Proc. Intern. Conf. on Pattern Recognition*, pp. 11-16, 1988.

[5] A. Azerbayejani, B. Horowitz and A. Pentland, "Recursive estimation of structure from motion using relative orientation constraints", *Proceedings of the International Conference of Computer Vision and Pattern Recognition*, IEEE Computer Society Press, pp.294-299, June 1993.

[6] H. Baker and T. Binford, "Depth from Edge and Intensity Based Stereo", *Int. Joint Conf. on Artificial Intelligence*, Vancouver, Canada, pp. 631-636, 1981.

[7] P. Beardsley, A. Zisserman and D. Murray, "Sequential Updating of Projective and Affine Structure from Motion", *International Journal of Computer Vision* (23), No. 3, Jun-Jul 1997, pp. 235-259.

[8] D. Bondyfalat and S. Bougnoux, "Imposing Euclidean Constraints During Self-Calibration Processes", *Proc. SMILE Workshop (post-ECCV'98)*, Lecture Notes in Computer Science, Vol. 1506, Springer-Verlag, pp.224-235, 1998.

[9] B. Boufama, R. Mohr and F. Veillon, "Euclidian Constraints for Uncalibrated Reconstruction", *Proc. International Conference on Computer Vision*, pp. 466-470, 1993.

[10] J.-Y. Bouguet and P. Perona, "3D Photography on your Desk". Proc. 6th Int. Conf. Computer Vision, Bombay, India, pages 43-50, January 1998.

[11] D. Brown, "Close-range camera calibration", Photogrammetric Engineering, 37(8):855-866, 1971.

[12] D. Brown, "The bundle adjustment - progress and prospect", *XIII Congress of the ISPRS*, Helsinki, 1976.

[13] Y. Chen and G. Medioni, "Object Modeling by Registration of Multiple Range Images", *Proc. Int. Conf. on Robotics and Automation*, 1991.

[14] K. Cornelis, M. Pollefeys, M. Vergauwen and L. Van Gool, "Augmented Reality from Uncalibrated Video Sequences", In M. Pollefeys, L. Van Gool, A. Zisserman, A. Fitzgibbon (Eds.), *3D Structure from Images - SMILE 2000*, Lecture Notes in Computer Science, Vol. 2018, pp.150-167, Springer-Verlag, 2001.

[15] P. Courtney, N. Thacker and C. Brown, "A hardware architecture for image rectification and ground plane obstacle detection", *Proc. Intern. Conf. on Pattern Recognition*, pp. 23-26, 1992.

[16] I. Cox, S. Hingorani and S. Rao, "A Maximum Likelihood Stereo Algorithm", *Computer Vision and Image Understanding*, Vol. 63, No. 3, May 1996.

[17] N. Cui, J. Weng and P. Cohen, "Extended Structure and Motion Analysis from Monocular Image Sequences", *Proc. International Conference on Computer Vision*, pp. 222-229, Osaka, Japan, 1990.

[18] B. Curless. Better optical triangulation and volumetric reconstruction of complex models from range images. PhD thesis, Stanford University, 1996.

[19] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images" *Proc. SIGGRAPH '96*, 1996.

[20] L. de Agapito, R. Hartley and E. Hayman, "Linear calibration of a rotating and zooming camera", *Proc. CVPR*, pp. 15-20, 1999.

[21] R. Deriche and G. Giraudon, "A computational approach for corner and vertex detection", *International Journal of Computer Vision*, 1(2):167-187, 1993.

[22] P. Debevec, C. Taylor and J. Malik, "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach", *Siggraph*, 1996.

[23] P. Debevec, Y. Yu and G. Borshukov, "Efficient View-Dependent Image-Based Rendering with Projective Texture Mapping", *Proc. SIGGRAPH '98*, ACM Press, New York, 1998.

[24] U. Dhond and J. Aggarwal, "Structure from Stereo - A Review", IEEE Trans. Syst., Man and Cybern. 19, 1489-1510, 1989.

[25] L. Falkenhagen, "Depth Estimation from Stereoscopic Image Pairs assuming Piecewise Continuous Surfaces", *European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Productions*, Hamburg, Germany, 1994.

[26] L. Falkenhagen, "Hierarchical Block-Based Disparity Estimation Considering Neighbourhood Constraints". *Proc. International Workshop on SNHC and 3D Imaging*, Rhodes, Greece, 1997.

[27] O. Faugeras and G. Toscani, "Camera Calibration for 3D Computer Vision", *International Workshop on Machine Vision and Machine Intelligence*, pp. 240-247, Tokyo, 1987.

[28] O. Faugeras, L. Quan and P. Sturm, "Self-Calibration of a 1D Projective Camera and Its Application to the Self-Calibration of a 2D Projective Camera", *Computer Vision – ECCV'98*, vol.1, Lecture Notes in Computer Science, Vol. 1406, Springer-Verlag, pp.36-52, 1998.

[29] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*, MIT press, 1993.

[30] O. Faugeras, "Stratification of three-dimensional vision: projective, affine, and metric representations", Journal of the Optical Society of America A, pp. 465–483, Vol. 12, No.3, March 1995.

[31] O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig", *Computer Vision - ECCV'92*, Lecture Notes in Computer Science, Vol. 588, Springer-Verlag, pp. 563-578, 1992.

[32] O. Faugeras, Q.-T. Luong and S. Maybank. "Camera self-calibration: Theory and experiments", *Computer Vision - ECCV'92*, Lecture Notes in Computer Science, Vol. 588, Springer-Verlag, pp. 321-334, 1992.

[33] O. Faugeras and S. Maybank, "Motion from point matches: multiplicity of solutions", *International Journal of Computer Vision*, 4(3):225-246, June 1990.

[34] O. Faugeras and B. Mourrain, "on the geometry and algebra of point and line correspondences between *n* images", *Proc. International Conference on Computer Vision*, 1995, pp. 951-962.

[35] M. Fischler and R. Bolles, "RANdom SAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography", *Commun. Assoc. Comp. Mach.*, 24:381-95, 1981.

[36] A. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences", *Computer Vision – ECCV'98*, vol.1, Lecture Notes in Computer Science, Vol. 1406, Springer-Verlag, 1998. pp.311-326, 1998.

[37] W. Förstner, "A framework for low level feature extraction" *Computer Vision-ECCV'90*, Lecture Notes in Computer Science, Vol. 427, Springer-Verlag, pp.383-394, 1990.

[38] G. Gimel'farb, "Symmetrical approach to the problem of automatic stereoscopic measurements in photogrammetry", Cybernetics, 1979, 15(20, 235-247; Consultants Bureau, N.Y.

[39] S. Gortler, R. Grzeszczuk, R. Szeliski and M. F. Cohen, "The Lumigraph", *Proc. SIGGRAPH '96*, pp 43–54, ACM Press, New York, 1996.

[40] W. Grimson, *From Images to Surfaces: A Computational Study of the Human Early Visual System*, MIT Press, Cambridge, Massachusetts, 1981.

[41] A. Gruen, "Accuracy, reliability and statistics in close-range photogrammmetry", *Proceedings of the Symposium of the ISP Commision V*, Stockholm, 1978.

[42] A. Gruen and H. Beyer, "System calibration through self-calibration", *Proceedings of the Workshop on Calibration and Orientation of Cameras in Computer Vision*, 1992.

[43] G. Golub and C. Van Loan, *Matrix Computations*, John Hopkins University Press, 1983.

[44] R. Haralick, C. Lee, K. Ottenberg, and M. Nolle, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", *International Journal of Computer Vision*, Vol.13, No.3, 1994, pp. 331-356.

[45] C. Harris and M. Stephens, "A combined corner and edge detector", *Fourth Alvey Vision Conference*, pp.147-151, 1988.

[46] R. Hartley, "Estimation of relative camera positions for uncalibrated cameras", *Computer Vision - ECCV'92*, Lecture Notes in Computer Science, Vol. 588, Springer-Verlag, pp. 579-587, 1992.

[47] R.Hartley, Chirality *International Journal of Computer Vision*, 26(1):41-61, January 1998.

[48] R. Hartley, "Euclidean reconstruction from uncalibrated views", in : J.L. Mundy, A. Zisserman, and D. Forsyth (eds.), *Applications of Invariance in Computer Vision*, Lecture Notes in Computer Science, Vol. 825, Springer-Verlag, pp. 237-256, 1994.

[49] R. Hartley, "Projective reconstruction from line correspondences", *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society Press, 1994.

[50] R. Hartley, "Self-calibration from multiple views with a rotating camera", Lecture Notes in Computer Science, Vol. 800-801, Springer-Verlag, pp. 471-478, 1994.

[51] R. Hartley, "A linear method for reconstruction from points and lines", *Proc. International Conference on Computer Vision*, pp. 882-887, 1995.

[52] R. Hartley, "In defense of the eight-point algorithm". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(6):580-593, June 1997.

[53] R. Hartley and P. Sturm, "Triangulation", *Computer Vision and Image Understanding*, 68(2):146-157, 1997.

[54] R. Hartley, "Computation of the Quadrifocal Tensor", *Computer Vision-ECCV'98*, Lecture Notes in Computer Science, Vol. 1406, Springer-Verlag, pp. 20-35, 1998.

[55] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[56] B. Heigl, R. Koch, M. Pollefeys, J. Denzler and L. Van Gool, "Plenoptic Modeling and Rendering from Image Sequences taken by Hand-held Camera", In *Proc. DAGM'99*, pp.94-101.

[57] A. Heyden and K. Åström, "Euclidean Reconstruction from Constant Intrinsic Parameters" *Proc. 13th International Conference on Pattern Recognition*, IEEE Computer Soc. Press, pp. 339-343, 1996.

[58] A. Heyden and K. Åström, "Euclidean Reconstruction from Image Sequences with Varying and Unknown Focal Length and Principal Point", *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, pp. 438-443, 1997.

[59] A. Heyden, *Geometry and Algebra of Multiple Projective Transformations*, Ph.D.thesis, Lund University, 1995.

[60] A. Heyden and K. Åström, "Minimal Conditions on Intrinsic Parameters for Euclidean Reconstruction", Asian Conference on Computer Vision, Hong Kong, 1998.

[61] M. Irani and S. Peleg, Super resolution from image sequences, *Proc. International Conference on Pattern Recognition*, Atlantic City, NJ, 1990.

[62] F. Kahl, "Critical Motions and Ambiuous Euclidean Reconstructions in Auto-Calibration", *Proc. ICCV*, pp.469-475, 1999.

[63] F. Kahl, B. Triggs, K. strm, "Critical Motions for Auto-Calibration When Some Intrinsic Parameters Can Vary", *Journal of Mathematical Imaging and Vision* 13,131-146,2000.

[64] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice* , Elsevier Science, Amsterdam, 1996.

[65] W. Karl, G. Verghese and A. Willsky, Reconstructing ellipsoids from projections. *CVGIP; Graphical Models and Image Processing*, 56(2):124-139, 1994.

[66] R. Koch, M. Pollefeys, L. Van Gool, "Realistic surface reconstruction of 3D scenes from uncalibrated image sequences", *Journal Visualization and Computer Animation*, Vol. 11, pp. 115-127, 2000.

[67] R. Koch, M. Pollefeys, B. Heigl, L. Van Gool and H. Niemann. "Calibration of Hand-held Camera Sequences for Plenoptic Modeling", *Proc.ICCV'99 (international Conference on Computer Vision)*, pp.585-591, Corfu (Greece), 1999.

[68] R. Koch, B. Heigl, M. Pollefeys, L. Van Gool and H. Niemann, "A Geometric Approach to Lightfield Calibration", *Proc. CAIP99*, LNCS 1689, Springer-Verlag, pp.596-603, 1999.

[69] R. Koch, *Automatische Oberflachenmodellierung starrer dreidimensionaler Objekte aus stereoskopischen Rundum-Ansichten*, *PhD thesis*, University of Hannover, Germany, 1996 also published as Fortschritte-Berichte VDI, Reihe 10, Nr.499, VDI Verlag, 1997.

[70] R. Koch, M. Pollefeys and L. Van Gool, Multi Viewpoint Stereo from Uncalibrated Video Sequences. *Proc. European Conference on Computer Vision*, pp.55-71. Freiburg, Germany, 1998.

[71] R. Koch, M. Pollefeys and L. Van Gool, Automatic 3D Model Acquisition from Uncalibrated Image Sequences, Proceedings Computer Graphics International, pp.597-604, Hannover, 1998.

[72] R. Koch: Surface Segmentation and Modeling of 3-D Polygonal Objects from Stereoscopic Image Pairs. Proc. ICPR'96, Vienna 1996.

[73] R. Koch, "3-D Surface Reconstruction from Stereoscopic Image Sequences", *Proc. Fifth International Conference on Computer Vision*, IEEE Computer Soc. Press, pp. 109-114, 1995.

[74] A. Koschan, "Eine Methodenbank zur Evaluierung von Stereo-Vision-Verfahren", Ph.D. Thesis, Technische Universität Berlin, Berlin, Germany, 1991.

[75] E. Kruppa, "Zur ermittlung eines objektes aus zwei perspektiven mit innerer orientierung", *Sitz.-Ber. Akad. Wiss., Wien, math. naturw. Abt. IIa*, 122:1939-1948, 1913.

[76] S. Laveau and O. Faugeras, "Oriented Projective Geometry for Computer Vision", in : B. Buxton and R. Cipolla (eds.), *Computer Vision - ECCV'96*, Lecture Notes in Computer Science, Vol. 1064, Springer-Verlag, pp. 147-156, 1996.

[77] R. Lenz and R. Tsai, "Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:713-720, 1988.

[78] M. Levoy and P. Hanrahan, "Lightfield Rendering", *Proc. SIGGRAPH '96*, pp 31–42, ACM Press, New York, 1996.

[79] D. Liebowitz and A. Zisserman, "Combining Scene and Auto-calibration Constraints", *Proc. ICCV*, pp.293-300, 1999.

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

[81] W.E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In Computer Graphics (SIGGRAPH '87 Proceedings), volume 21, pages 163-169, July 1987.

[82] Q.-T. Luong, *Matrice Fondamentale et Autocalibration en Vision par Ordinateur*, PhD thesis, Université de Paris-Sud, France, 1992.

[83] Q.-T. Luong and O. Faugeras, "The fundamental matrix: theory, algorithms, and stability analysis", *Internation Journal of Computer Vision*, 17(1):43-76, 1996.

[84] Q.-T. Luong and O. Faugeras, "Self Calibration of a moving camera from point correspondences and fundamental matrices", *Internation Journal of Computer Vision*, vol.22-3, 1997.

[85] Y. Ma, S. Soatto, J, Košecká and S. Sastry, "Euclidean Reconstruction and Reprojection Up to Subgroups", *Proc. ICCV*, pp.773-780, 1999.

[86] D. Marr and T. Poggio, "A Computational Theory of Human Stereo Vision", *Proc. Royal Society of London*, Vol. 204 of B, pp. 301-328, 1979.

[87] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system", *Proc. SIGGRAPH'95*, pp. 39-46, 1995.

[88] T. Moons, "A Guided Tour Through Multiview Relations", *Proc. SMILE Workshop (post-ECCV'98)*, Lecture Notes in Computer Science 1506, Springer-Verlag, pp.304-346, 1998.

[89] T, Moons, L. Van Gool, M. Proesmans and E. Pauwels, "Affine reconstruction from perspective image pairs with a relative object-camera translation in between", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no.1, pp. 77-83, Jan. 1996.

[90] H.P. Moravec. "Visual mapping by a robot rover". *Proc. of the 6th International Joint Conference on Artificial Intelligence*, pp.598-600, 1979.

[91] J. Mundy and A. Zisserman, "Machine Vision", in : J.L. Mundy, A. Zisserman, and D. Forsyth (eds.), *Applications of Invariance in Computer Vision*, Lecture Notes in Computer Science, Vol. 825, Springer-Verlag, 1994.

[92] K. Nuyts, J.P.Kruth, B. Lauwers, M. Pollefeys, L. Qiongyan, J. Schouteden, P. Smars, K. Van Balen, L. Van Gool, M. Vergauwen. "Vision on Conservation: VIRTERF" Proc. International Symposium on Virtual and Augmented Architecture (VAA01), LNCS, 2001 (in press).

[93] E. Ofek, E. Shilat, A. Rappopport and M. Werman, "Highlight and Reflection Independent Multires-
      olution Textures from Image Sequences", *IEEE Computer Graphics and Applications*, vol.17 (2),
      March-April 1997.

[94] Y. Ohta and T. Kanade, "Stereo by Intra- and Inter-scanline Search Using Dynamic Programming",
      IEEE Trans. on Pattern Analysis and Machine Intelligence 7(2), 139-154, 1985.

[95] M. Okutomi and T. Kanade, "A Locally Adaptive Window for Signal Processing", *International Jour-
      nal of Computer Vision*, 7, 143-162, 1992.

[96] D. Papadimitriou and T. Dennis, "Epipolar line estimation and rectification for stereo image pairs",
      IEEE Trans. Image Processing, 5(4):672-676, 1996.

[97] PhotoModeler, by Eos Systems Inc., `http://www.photomodeler.com/`.

[98] S. Pollard, J. Mayhew and J. Frisby, "PMF: A Stereo Correspondence Algorithm Using a Disparity
      Gradient Limit", *Perception* 14(4), 449-470, 1985.

[99] M. Pollefeys, *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*, PhD.
      thesis, K.U.Leuven, 1999.

[100] M. Pollefeys, F. Verbiest, L. Van Gool, "Surviving dominant planes in uncalibrated structure and
      motion recovery", *Proc. ECCV 2002*, LNCS.

[101] M. Pollefeys, R. Koch, M. Vergauwen, L. Van Gool. "Automated reconstruction of 3D scenes from
      sequences of images", *Isprs Journal Of Photogrammetry And Remote Sensing* (55)4 (2000), pp. 251-
      267.

[102] M. Pollefeys, M. Vergauwen, L. Van Gool, "Automatic 3D modeling from image sequences", *In-
      ternational Archive of Photogrammetry and Remote Sensing*, Vol. XXXIII, Part B5, pp. 619-626,
      2000.

[103] M. Pollefeys, R. Koch, M. Vergauwen, B. Deknuydt, L. Van Gool. "Three-dimensional scene re-
      construction from images", *proc. SPIE Electronic Imaging, Three-Dimensional Image Capture and
      Applications III*, SPIE Proceedings series Vol. 3958, pp.215-226, 2000.

[104] M. Pollefeys and L. Van Gool, "Stratified self-calibration with the modulus constraint", accepted for
      publication in *IEEE transactions on Pattern Analysis and Machine Intelligence*.

[105] M. Pollefeys, R. Koch and L. Van Gool. "Self-Calibration and Metric Reconstruction in spite of
      Varying and Unknown Internal Camera Parameters", *International Journal of Computer Vision*.

[106] M. Pollefeys, R. Koch and L. Van Gool, "A simple and efficient rectification method for general
      motion", *Proc.ICCV'99 (international Conference on Computer Vision)*, pp.496-501, Corfu (Greece),
      1999.

[107] M. Pollefeys, R. Koch, M. Vergauwen and L. Van Gool, "An Automatic Method for Acquiring
      3D models from Photographs: applications to an Archaeological Site", accepted for *Proc. ISPRS
      International Workshop on Photogrammetric Measurements, Object Modeling and Documentation in
      Architecture and Industry*, july 1999.

[108] M. Pollefeys, M. Proesmans, R. Koch, M. Vergauwen and L. Van Gool, "Detailed model acquisition
      for virtual reality", in J. Barcelo, M. Forte and D. Sanders (eds.), *Virtual Reality in Archaeology*, to
      appear, ArcheoPress, Oxford.

[109] M. Pollefeys, R. Koch, M. Vergauwen and L. Van Gool, "Automatic Generation of 3D Models from
      Photographs", *Proceedings Virtual Systems and MultiMedia*, 1998.

[110] M. Pollefeys, R. Koch, M. Vergauwen and L. Van Gool, "Virtualizing Archaeological Sites", *Pro-
      ceedings Virtual Systems and MultiMedia*, 1998.

[111] M. Pollefeys, R. Koch, M. Vergauwen and L. Van Gool, "Metric 3D Surface Reconstruction from Uncalibrated Image Sequences", *Proc. SMILE Workshop (post-ECCV'98)*, Lecture Notes in Computer Science, Vol. 1506, pp.138-153, Springer-Verlag, 1998.

[112] M. Pollefeys, R. Koch, M. Vergauwen and L. Van Gool, "Flexible acquisition of 3D structure from motion", *Proceedings IEEE workshop on Image and Multidimensional Digital Signal Processing*, pp.195-198, Alpbach, 1998.

[113] M. Pollefeys, R. Koch, M. Vergauwen and L. Van Gool, "Flexible 3D Acquisition with a Monocular Camera", *Proceedings IEEE International Conference on Robotics and Automation*, Vol.4, pp.2771-2776, Leuven, 1998.

[114] M. Pollefeys, R. Koch and L. Van Gool, "Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters", *Proc. International Conference on Computer Vision*, Narosa Publishing House, pp.90-95, 1998.

[115] M. Pollefeys, L. Van Gool and M. Proesmans, "Euclidean 3D Reconstruction from Image Sequences with Variable Focal Lengths", *Computer Vision - ECCV'96*, Lecture Notes in Computer Science, Vol. 1064, Springer-Verlag, pp. 31-42, 1996.

[116] M. Pollefeys, L. Van Gool and A. Oosterlinck, "The Modulus Constraint: A New Constraint for Self-Calibration", *Proc. 13th International Conference on Pattern Recognition*, IEEE Computer Soc. Press, pp. 349-353, 1996.

[117] M. Pollefeys and L. Van Gool, "A stratified approach to self-calibration", *Proc. 1997 Conference on Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, pp. 407-412, 1997.

[118] M. Pollefeys and L. Van Gool, "Self-calibration from the absolute conic on the plane at infinity", *Proc. Computer Analysis of Images and Patterns*, Lecture Notes in Computer Science, Vol. 1296, Springer-Verlag, pp. 175-182, 1997.

[119] M. Pollefeys, L. Van Gool and T. Moons. "Euclidean 3D reconstruction from stereo sequences with variable focal lengths", *Recent Developments in Computer Vision*, Lecture Notes in Computer Science, Vol.1035, Springer-Verlag, pp. 405-414, 1996.

[120] M. Pollefeys, L. Van Gool and T. Moons. "Euclidean 3D reconstruction from stereo sequences with variable focal lengths", *Proc.Asian Conference on Computer Vision*, Vol.2, pp.6-10, Singapore, 1995

[121] M. Pollefeys, L. Van Gool and A. Oosterlinck, "Euclidean self-calibration via the modulus constraint", in F.Dillen, L.Vrancken, L.Verstraelen, and I. Van de Woestijne (eds.), *Geometry and topology of submanifolds, VIII*, World Scientific, Singapore, New Jersey, London, Hong Kong, pp.283-291, 1997.

[122] W. Press, S. Teukolsky and W. Vetterling, *Numerical recipes in C: the art of scientific computing*, Cambridge university press, 1992.

[123] P. Pritchett and A. Zisserman, "Wide Baseline Stereo Matching", *Proc. International Conference on Computer Vision*, Narosa Publishing House, pp. 754-760, 1998.

[124] P. Pritchett and A. Zisserman, "Matching and Reconstruction from Widely Separate Views", *Proc. SMILE Workshop (post-ECCV'98)*, Lecture Notes in Computer Science, Vol. 1506, Springer-Verlag, pp.78-92, 1998.

[125] M. Proesmans, L. Van Gool and A. Oosterlinck, "Determination of optical flow and its discontinuities using non-linear diffusion", *Computer Vision - ECCV'94*, Lecture Notes in Computer Science, Vol. 801, Springer-Verlag, pp. 295-304, 1994.

[126] M. Proesmans, L. Van Gool, F. Defoort, "Reading between the lines - a method for extracting dynamic 3D with texture", *Sixth international conference on computer vision*, pp. 1081-1086, January 4-7, 1998.

[127] P. Rousseeuw, *Robust Regression and Outlier Detection*, Wiley, New York, 1987.

[128] S. Roy, J. Meunier and I. Cox, "Cylindrical Rectification to Minimize Epipolar Distortion", *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.393-399, 1997.

[129] F. Schaffalitzky and A. Zisserman, "Geometric Grouping of Repeated Elements within Images", *Proc. 9th British Machine Vision Conference*, pp 13-22, 1998.

[130] C. Schmid and R. Mohr, "Local Greyvalue Invariants for Image Retrieval", *IEEE transactions on Pattern Analysis and Machine Intelligence*, Vol.19, no.5, pp 872-877, may 1997.

[131] C. Schmid, R. Mohr and C. Bauckhage, "Comparing and Evaluating Interest Points", *Proc. International Conference on Computer Vision*, Narosa Publishing House, pp. 230-235, 1998.

[132] J.G. Semple and G.T. Kneebone, *Algebraic Projective Geometry*, Oxford University Press, 1952.

[133] ShapeSnatcher, by Eyetronics, `http://www.eyetronics.com/`.

[134] A. Shashua, "Omni-Rig Sensors: What Can be Done With a Non-Rigid Vision Platform?" *Proc. of the Workshop on Applications of Computer Vision (WACV)*, Princeton, Oct. 1998.

[135] A. Shashua, "Trilinearity in visual recognition by alignment", *Computer Vision - ECCV'94*, Lecture Notes in Computer Science, Vol. 801, Springer-Verlag, pp. 479-484, 1994.

[136] J. Shi and C. Tomasi, "Good Features to Track", *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pp. 593 - 600, 1994.

[137] C. Slama, *Manual of Photogrammetry*, American Society of Photogrammetry, Falls Church, VA, USA, 4th edition, 1980.

[138] M. Soucy and D. Laurendeau. "A general surface approach to the integration of a set of range views". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):344-358, April 1995.

[139] S.M. Smith and J.M. Brady. "SUSAN - a new approach to low level image processing". Int. Journal of Computer Vision, Vol.23, Nr.1, pp.45-78, 1997.

[140] M. Spetsakis and J. Aloimonos, "Structure from motion using line correspondences", *International Journal of Computer Vision*, 4(3):171-183, 1990.

[141] G. Stein, "Lens Distortion Calibration Using Point Correspondences", *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, pp 602-608, 1997.

[142] P. Sturm, *Vision 3D non-calibrée: contributions à la reconstruction projective et études des mouvements critiques pour l'auto-calibrage*, Ph.D. Thesis, INP de Grenoble, France , 1997.

[143] P. Sturm and L. Quang, "Affine stereo calibration", *Proceedings Computer Analysis of Images and Patterns*, Lecture Notes in Computer Science, Vol. 970, Springer-Verlag, pp. 838-843, 1995.

[144] P. Sturm, "Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction", *Proc. 1997 Conference on Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, pp. 1100-1105, 1997.

[145] P. Sturm, "Critical motion sequences and conjugacy of ambiguous Euclidean reconstructions", *Proc. SCIA - 10th Scandinavian Conference on Image Analysis*, Lappeenranta, Finland, pp. 439-446, 1997.

[146] P. Sturm, "Critical Motion Sequences for the Self-Calibration of Cameras and Stereo Systems with Variable Focal Length". *Proc. BMVC - 10th British Machine Vision Conference*, pp. 63-72, 1999.

[147] R. Szeliski and S. B. Kang, "Recovering 3D shape and motion from image streams using non-linear least-squares", Journal of Visual Communication and Image Representation, 5(1):10–28, March 1994.

[148] C. Taylor, P. Debevec and J. Malik, "Reconstructing Polyhedral Models of Architectural Scenes from Photographs", *Computer Vision - ECCV'96*, Lecture Notes in Computer Science, Vol. 1065, vol.II, pp 659-668, 1996.

[149] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization approach", *International Journal of Computer Vision*, 9(2):137-154, 1992.

[150] P. Torr. "An assessment of information criteria for motion model selection". In *CVPR97*, pages 47–53, 1997.

[151] P. Torr and A. Zisserman, "Robust parametrization and computation of the trifocal tensor", *Image and Vision Computing*, 15(1997) 591-605.

[152] P. Torr and A. Zisserman, "Robust Computation and Parameterization of Multiple View Relations", *Proc. International Conference on Computer Vision*,Narosa Publishing house, pp 727-732, 1998.

[153] P. Torr, P. Beardsley and D. Murray, "Robust Vision", *Proc. British Machine Vision Conference*, 1994.

[154] P. Torr, *Motion Segmentation and Outlier Detection*, PhD Thesis, Dept. of Engineering Science, University of Oxford, 1995.

[155] P. Torr, A. Fitzgibbon and A. Zisserman, "Maintaining Multiple Motion Model Hypotheses Over Many Views to Recover Matching and Structure", *Proc. International Conference on Computer Vision*, Narosa Publishing house, pp 485-491, 1998.

[156] B. Triggs, "The geometry of projective reconstruction I: Matching constraints and the joint image", *Proc. International Conference on Computer Vision*, IEEE Computer Soc. Press, pp. 338-343, 1995.

[157] B. Triggs, "The Absolute Quadric", *Proc. 1997 Conference on Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, pp. 609-614, 1997.

[158] B. Triggs, P. McLauchlan, R. Hartley, A. Fiztgibbon, "Bundle Adjustment – A Modern Synthesis", In B. Triggs, A. Zisserman, R. Szeliski (Eds.), *Vision Algorithms: Theory and Practice*, LNCS Vol.1883, pp.298-372, Springer-Verlag, 2000.

[159] R. Tsai and T. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces", *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol.6, pp.13-27, Jan. 1984.

[160] R. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision", *Proc. Computer Vision and Pattern Recognition*, 1986.

[161] R. Tsai. "A versatile camera calibration technique for high-accuracy 3D machine vision using off-the-shelf TV cameras and lenses". *IEEE Journal of Robotics and Automation*, RA-3(4):323-331, August 1987.

[162] G. Turk and M. Levoy "Zippered Polygon Meshes from Range Images" Proceedings of SIGGRAPH '94 pp. 311-318.

[163] T. Tuytelaars. M. Vergauwen, M. Pollefeys and L. Van Gool, "Image Matching for Wide baseline Stereo", Proc. International Conference on Forensic Human Identification, 1999.

[164] T. Tuytelaars and L. Van Gool "Wide Baseline Stereo based on Local, Affinely invariant Regions" British Machine Vision Conference, pp. 412-422, 2000.

[165] L. Van Gool, F. Defoort, R. Koch, M. Pollefeys, M. Proesmans and M. Vergauwen, "3D modeling for communications", *Proceedings Computer Graphics International*, pp.482-487, Hannover, 1998.

[166] L. Van Gool, T. Moons, D. Ungureanu, "Affine/photometric invariants for planar intensity patterns" , Proceedings 4th European Conference on Computer Vision, ECCV'96, Lecture Notes in Computer Science, vol. 1064, pp.642-651, 1996.

[167] M. Vergauwen, M. Pollefeys, L. Van Gool, "A stereo vision system for support of planetary surface exploration", *Proc. International Conference on Vision Systems*, LNCS, 2001.

[168] M. Vergauwen, M. Pollefeys, R. Moreas, F. Xu, G. Visentin, L. Van Gool and H. Van Brussel. "Calibration, Terrain Reconstruction and Path Planning for a Planetary Exploration System", *Proc. i-SAIRAS 2001*.

[169] M. Vergauwen, M. Pollefeys, L. Van Gool, Calibration and 3D measurements from Martian Terrain Images, Proc. International Conference on Robotics and Automation, IEEE Computer Society Press, 2001.

[170] J. Weng, P. Cohen and M. Herniou, "Camera calibration with distortion models and accuracy evaluation", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(10):965-980, 1992.

[171] R. Willson and S. Shafer, "A Perspective Projection Camera Model for Zoom Lenses", *Proceedings Second Conference on Optical 3-D Measurement Techniques*, Zurich Switzerland, October 1993.

[172] R. Willson, "Modeling and Calibration of Automated Zoom Lenses" *Proceedings of the SPIE 2350:Videometrics III*, Boston MA, October 1994, pp.170-186.

[173] R. Willson, *Modeling and Calibration of Automated Zoom Lenses*, Ph.D. thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, January 1994.

[174] R. Willson and S. Shafer, "What is the Center of the Image?", *Journal of the Optical Society of America A*, Vol. 11, No. 11, pp.2946-2955, November 1994.

[175] G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press Monograph, ISBN 0-8186-8944-7, 1990.

[176] C. Zeller, *Calibration projective, affine et Euclidienne en vision par ordinateur et application a la perception tridimensionnelle*, Ph.D. Thesis, Ecole Polytechnique, France, 1996.

[177] Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry", *Artificial Intelligence Journal*, Vol.78, pp.87-119, October 1995.

[178] Z. Zhang, "On the Epipolar Geometry Between Two Images with Lens Distortion", *Proc. International Conference on Pattern Recognition*, IEEE Computer Soc. Press, A80.13, 1996.

[179] C. Loop and Z. Zhang. "Computing Rectifying Homographies for Stereo Vision". IEEE Conf. Computer Vision and Pattern Recognition (CVPR'99), Colorado, June 1999.

[180] A. Zisserman, D. Liebowitz and M. Armstrong, "Resolving ambiguities in auto-calibration", *Phil. Trans. R. Soc. Lond.*, A(1998) 356, 1193-1211.

Curriculum Vitae

Marc Pollefeys is an assistant professor at the computer science department of the University of North Carolina – Chapel Hill since July 2002. Before this he was a post-doctoral researcher of the Funds for Scientific Research - Flanders affiliated to the Center for Processing of Speech and Images of the K.U.Leuven. His PhD dissertation on "Self-calibration and Metric 3D Reconstruction from Uncalibrated Image Sequences" was awarded the Scientific Prize BARCO 1999. His current research focuses on 3D modeling from images, multi-view geometry, image-based rendering, virtual and augmented reality and applications. He is currently involved in projects ranging from digital archaeology, over immersive video and 3D television, to planetary rover control. Marc Pollefeys has published over 60 technical papers and won several awards, amongst which the prestigious Marr Prize at the International Conference on Computer Vision in 1998. He is the chair of the working group on Image Sequences of the International Society for Photogrammetry and Remote Sensing (ISPRS). He has organized the Second European Workshop on Structure from Multiple Images of Large-scale Environments in Dublin in 2000 and several courses and tutorials at major vision and graphics conferences. He has served on the program commitee of several international conferences and workshops and is a regular reviewer for most major computer vision, computer graphics and photogrammetry journals and conferences.