Technische Universität Hamburg-Harburg

Robust Relative Pose Estimation of two Cameras by Decomposing Epipolar Geometry

Vom Promotionsausschuss der
Technischen Universität Hamburg-Harburg
zur Erlangung des akademischen Grades
Doktor Ingenieur
genehmigte Dissertation

von
Fabian Wenzel

aus Hamburg
2007

1. Gutachter: Prof. Dr.-Ing. Rolf-Rainer Grigat
2. Gutachter: Prof. Dr.-Ing. Reinhard Koch

Tag der mündlichen Prüfung: 23.05.2007

Fabian Wenzel

Robust Relative Pose Estimation of two Cameras by Decomposing Epipolar Geometry

Hamburg 2006

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über `http://dnb.d-nb.de` abrufbar.

Herstellung und Verlag: Books on Demand GmbH, Norderstedt
ISBN: 978-3833499340

Acknowledgements

First of all, I would like to thank Prof. Dr. Rolf-Rainer Grigat for giving me the opportunity of doing research in the fascinating field of computer vision and for supporting my work. I am also greatful to Prof. Dr. Reinhard Koch for being the second referee of my thesis and to Prof. Dr. Siegfried Rump for chairing my graduation process.

The members of the Vision Systems group made this research very enjoyable, and I would like to thank them all for many technical discussions. Special thanks go to Marco Grimm for creative conversations on our ideas, to Philipp Urban who helped me identify mathematical pitfalls, and to Dipl.-Ing. Ralph Kricke for always increasing my mood with his excellent humor.

Many thanks go to my friends Thilo Heinrichson, Marko Soltau, Rüdiger Bölcke and Matthias Dehn who listened to me when not everything went fine and who helped me free my mind in life outside the university.

I would like to thank my parents deeply for their continuous support.

Finally, Sandra, I would like to thank you for your love, for your patience and for always keeping me grounded.

Hamburg, May 2007

Contents

List of Symbols

α	Real part of complex filter response γ, page 83
α_x, α_y	Scaling factors for intrinsic projection, page 11
β	Imaginary part of complex filter response γ, page 83
$\mathbf{C}$	Optical center of a camera, page 9
C	Cross-ratio, page 8
D	Filter kernel width, page 81
d_2	Euclidean distance, page 23
$d_{\mathtt{C}}$	Mahalanobis distance, page 23
Δe	Angular error between two homogeneous vectors, page 30
$\mathbf{e}, \mathbf{e}'$	Left and right epipole of a stereo camera system, page 14
$\mathtt{E}$	Essential matrix, page 15
$\hat{\mathbf{e}}, \hat{\mathbf{e}}'$	Left and right normalized epipole of a stereo camera system, page 14
$\mathbb{R}^n$	Euclidean space, page 5
$\hat{\mathbf{e}}, \hat{\mathbf{e}}'$	Canonic epipoles on the z-axis in transformed space, page 51
$\mathtt{F}$	Fundamental matrix, page 16
f	Focal length, page 11
g	Filter kernel width for Gaussian smoothing, page 44
γ	Complex filter response of quadrature filtering, page 83
$\mathtt{H}$	Homography, page 7
H	Height of an image, page 100

$\mathtt{H}_d$ Differential Homography, page 85

$\mathtt{H}_E$ Euclidean transformation matrix, page 7

$\mathtt{H}_H$ Householder transformation, page 50

$\mathtt{H}_N$ Prenormalization matrix the 8-point algorithm, page 22

$\mathtt{H}_P$ Singular homography, projection, page 9

$\mathtt{H}_A$ Affine transformation matrix, page 7

$\mathtt{H}_S$ Similarity transformation matrix, page 7

$\mathtt{I}$ Identity Matrix, page 9

I Image intensity, page 21

k Minimum number of samples for a model, page 27

$\mathtt{K}$ Internal camera matrix, page 11

κ_A Condition of a matrix, page 127

k_x, k_y Pixel dimensions of a CCD sensor element, page 11

$\mathbf{l}$ Line in $\mathbb{P}^2$, page 6

$\mathtt{L}$ Epipolar line homography, page 29

$\mathcal{L}$ Ray through camera center $\mathbf{C}$, page 14

$\mathtt{M}$ Unnormalized relative orientation, part of a projection matrix $\mathtt{P}$, page 12

$\mathcal{N}$ Gaussian noise, page 30

N Number of correspondences, page 13

$O(n)$ Orthogonal group of matrices, page 129

$\mathtt{P}$ Projection matrix, page 12

$\mathtt{P}_0$ Canonic perspective projection matrix, page 11

ϕ Argument of complex filter response γ, page 83

φ Longitude of spherical coordinates, page 34

φ_E Rotation angle for normalized epipolar line homography, page 54

$\boldsymbol{\pi}$ Projective hyperplane, page 6

$\mathbb{P}^n$ Projective space, page 5

$\mathtt{R}$ Rotation matrix, in particular in $\mathbb{R}^3$, page 7

ρ Cost function, page 23

ρ_i Individual error term of cost function ρ, page 23

s Scale factor, page 7

S_j Random sample of data points, page 27

S^2 Gaussian sphere, unit sphere in $\mathbb{R}^2$, page 34

τ Shift for projection-based M-estimation, page 66

σ_L Singular value of a normalized fundamental matrix, page 53

s_k Shearing of coordinate axes, page 11

$SO(n)$ Special orthogonal group of matrices, page 129

$\mathbf{t}$ Translation vector $\in \mathbb{R}^3$, page 7

t Threshold parameter, page 27

$\boldsymbol{\theta}$ Orientation for projection-based M-estimation, page 66

ϑ Co-latitude of spherical coordinates, page 34

$\mathbf{u}$ Motion vector in $\mathbb{R}^2$, page 21

$\mathbf{v}$ Instantaneous image velocity of a moving point (optical flow) in $\mathbb{R}^2$, page 21

$\mathbf{w}$ Mirror vector for Householder reflections, page 51

$\mathtt{W}$ Givens rotation by 90° around optical axis, page 18

W Width of an image in pixels, page 43

$\mathbf{x}$ Projective point, specifically in $\mathbb{P}^2$, page 6

$\mathbf{X}_E$ Euclidean point in $\mathbb{R}^3$, page 9

x_0, y_0 Pixel coordinates of a camera's optical axis, page 11

$\mathbf{X}_C$ Homogeneous point given in camera coordinates, page 10

$\mathbf{x}_E$ Euclidean point in $\mathbb{R}^2$, page 9

$\hat{\mathbf{x}}$ Normalized point in $\mathbb{P}^2$, page 11

$\mathbf{x}'$ Point corresponding to $\mathbf{x}$, page 16

$\mathbf{x}^*$ True position of a point, page 23

$\tilde{\mathbf{x}}, \tilde{\mathbf{x}}'$ Corresponding points in transformed space, in which the z-axes of a stereo camera system coincide, page 52

$\mathbf{X}_W$ Homogeneous point given in world coordinates, page 10

$[\mathbf{x}]_\times$ Cross-product of vector $\mathbf{x}$ in matrix notation, page 6

$\mathbb{Z}$ Space of integers, page 87

Chapter 1

Introduction

1.1 Motivation

One of the most outstanding abilities of the human visual system is to perceive three-dimensional space, even though the eyes only capture two-dimensional projections of the world we are living in.

The perception of 3D space by a computer has been of interest for a wide range of research during the last decades, ranging from the field of artificial intelligence and machine learning to photogrammetry. There has been significant progress in the development of methods and algorithms for 3D computer vision in the last twenty years. Whereas before, calibration objects had to be used for 3D computations, they are not necessarily needed anymore in order to estimate the relative pose of two cameras or to reconstruct points of a 3D scene. Nowadays, many applications of 3D geometry can be found in robotics, medical imaging, image-based rendering, surveillance, model acquisition, or augmented reality, to name a few [TBM02, FCSK02, DBF98, Nie94, SWV+00].

Estimating 3D structure using a sequence of real images still contains open problems. Some are caused by the fact that real data is not perfect but contains different kinds of errors: Not only may point correspondences be affected by noise, they may also be wrong so that they do not belong to the underlying 3D geometry of the scene.

Therefore, a mandatory first step in most current techniques for estimating the relative pose of two cameras is the classification of point correspondences into *inliers* and *outliers*. Questions for which fully satisfying answers have not been found yet are the following: Is it possible to estimate relative pose without any additional information like the scale of noise? Can 3D parameters of the scene be determined with measurements including outliers? How can epipolar geometry contribute to develop robust algorithms for the estimation of two cameras' relative pose?

The present work discusses these problems. It is based on a new factorization of epipolar geometry that explicitly reveals two parts which describe the relative pose of two cameras. The objective of the present work is to examine if and how the proposed decomposition may contribute to the process of relative pose estimation by applying it to different techniques.

1.2 Outline of the thesis

This thesis is organized as follows: In the next chapter, the framework of projective geometry is reviewed, including the pinhole camera model. The mathematical basics for weak calibration, i. e. the estimation of two cameras' relative pose given corresponding image points, is described. The two main entities of epipolar geometry, the fundamental and the essential matrix, are laid out. This chapter also reviews the state of the art for calibration techniques, i. e. methods for estimating camera parameters if corresponding points are not exact.

Besides laying out mathematical fundamentals, epipolar geometry is approached from a geometric point of view. In particular, an interpretation of the fundamental and essential matrix is outlined which explicitly reveals the two underlying components, the epipoles and the epipolar line homography. Their stability in case of noisy point correspondences is analyzed. Subsequently, a factorization of epipolar geometry is motivated and its application for solving the relative pose problem is outlined.

In chapter 3, suitable parametrizations for factorizing epipolar geometry are introduced. Whereas the first part concentrates on finding minimal representations of an epipole, section 3.3 describes parametrizations of the fundamental and the essential matrix.

Chapter 4 discusses an approach for geometry-driven relative pose estimation using a sparse set of point correspondences. It utilizes a projection-based estimation scheme which so far has not been applied to the estimation of a fundamental matrix without losing some of its properties. In this section, an estimation technique for the intrinsically calibrated case is suggested which explicitly exploits the properties of an essential matrix during optimization.

In chapter 5, a dense approach is laid out in which the spatial coherence of corresponding image points is directly exploited. Local properties of correspondences at foreground/background discontinuities are analyzed and constraints with respect to gradients of a motion field are obtained by means of differential homographies. Eventually, an estimation scheme for epipoles by employing a Hough transform is proposed.

Chapter 6 compares the two techniques of the previous chapters with standard ones in terms of accuracy in real or synthetic environments.

In chapter 7, a conclusion and an outlook of this work is given.

Appendix A discusses the software framework which has been developed within the scope of this work. It is different from existing signal processing applications as it applies generative programming to the development of modular algorithms. This way, the introduced techniques could be implemented with minimal overhead.

1.2.1 Main Contributions of this Work

- A novel geometry-based decomposition of the fundamental and the essential matrix has been developed. It is able to minimally represent the epipoles as well as the epipolar line homography. In contrast to previously known techniques, it is a symmetric

decomposition that also considers the case of known internal camera parameters, and it is directly able to represent epipoles at infinity.

- Representations for homogeneous points have been evaluated in order to establish a minimal parametrization of epipoles that is particularly suitable for Hough transforms. In this context, mappings from cartography have been identified to yield superior properties than commonly used representations like spherical coordinates.
- A relation between local variations of a smooth dense motion field and their horizontal and vertical derivations has been established. It could be shown analytically that gradients of a continuous motion field cannot have identical orientations except for degenerate situations.
- An estimation scheme for the relative pose of two cameras using sparse point correspondences has been formulated that retains characteristics of an essential matrix during optimization.
- A dense estimation scheme for epipoles based on a Hough transform has been developed. As opposed to previous techniques, it is able to deal with arbitrary camera movements and has relaxed requirements with respect to the accuracy of single motion vectors.
- A software framework for image processing applications has been designed for the evaluation of the proposed methods. It utilizes generative techniques in order to reduce dependencies of newly implemented algorithms.

The relevance of some of the introduced techniques is not limited to the field of 3D computer vision. The minimal representation of epipoles can also be applied to other tasks like vanishing point detection [WG06a]. The software framework described in appendix A can be used in other signal processing domains as well [WG05a].

The structure and contributions of this thesis are summarized in figure 1.1.

1.3 Preliminaries

The main focus of this work is the analysis of image sequences captured by a single, moving camera. Its internal camera parameters are required to be known for most of the algorithms introduced in chapters 4 and 5. Section 2.3.2 shows how this is possible by using a calibration object, even though other techniques like self-calibration exist [HZ03]. Moreover, it is assumed that corresponding points or blocks in the two images have been identified in a prior step. Section 2.6.1 gives a brief overview about possible techniques.

1.4 Notation

The notation of symbols follows Hartley and Zisserman's book [HZ03].

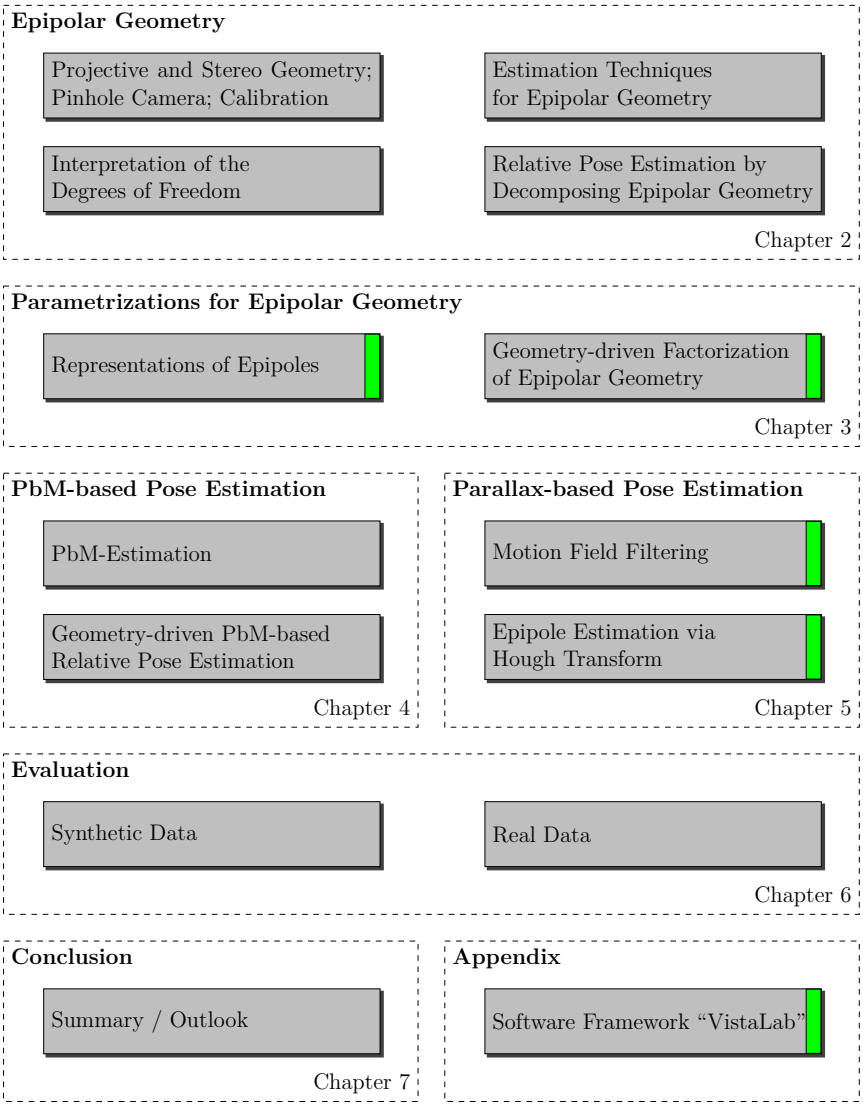


Figure 1.1: Structure of this thesis. A green mark indicates that the corresponding content has been published within the scope of this work.

Chapter 2

Epipolar Geometry

Most of the subsequently introduced methods draw on epipolar geometry. Hence, this chapter describes fundamental concepts necessary for following the rest of this work. After laying out mathematical basics of projective geometry, the pinhole camera model is introduced. One of its major advantages is that it can be completely represented by a single matrix, the *projection* matrix. Section 2.3 describes how to estimate a projection matrix from known 2D/3D point correspondences via camera calibration. Known 3D points may not be required if two cameras are used to look at a static scene. They can also be related by 2D/2D correspondences of points in the two views. In this case, it is possible to weakly calibrate the two cameras by using the framework of *epipolar geometry*. Its core components, the *fundamental* and the *essential* matrix, will be described in section 2.4 and it will be laid out how and in which cases the relative pose of two weakly calibrated cameras can be determined.

2.1 Projective Geometry

Euclidean geometry is able to model the geometry of the three-dimensional world we are living in. However, it fails to consistently relate 3D space to its images that are captured by our eyes or by cameras. As a simple example, two parallel lines do not intersect in the Euclidean world. But when taking a photo of e. g. a railway, the two rails on the image do intersect at the horizon. The reason for this phenomenon is that 2D images are a *projection* of the 3D world which does not preserve parallelity. *Projective geometry* is able to model the world along with their projections. It introduces the *projective space* $\mathbb{P}^n$, which is different from Euclidean space $\mathbb{R}^n$. For example, points in projective space $\mathbb{P}^n$ are represented by *homogeneous coordinates* that are invariant to scaling. Projective geometry is subject to many books, in which a lot of details and additional mathematical aspects can be found [HZ03, Fau93, FL01].

2.1.1 Projective Point

A point $\mathbf{x}$ in projective space $\mathbb{P}^n$ is represented by a *homogeneous* vector containing $n+1$ elements of which at least one must not be equal to zero:

$$\mathbf{x} = \begin{pmatrix} x_1, x_2, x_3, \ldots, x_n, x_{n+1} \end{pmatrix}^\top$$

If $x_{n+1} \neq 0$, $\mathbf{x}$ corresponds to the Euclidean point

$$\mathbf{x}_E = \frac{1}{x_{n+1}} \begin{pmatrix} x_1, x_2, \ldots, x_n \end{pmatrix}^\top .$$

If $x_{n+1} = 0$, then $\mathbf{x}$ does not have a Euclidean representation. It is located at infinity and describes an *ideal* point at the Euclidean *direction* $\begin{pmatrix} x_1, x_2, \ldots, x_n \end{pmatrix}^\top$.

Two projective points $\mathbf{x}_1$ and a scaled version $\mathbf{x}_2 = \lambda \mathbf{x}_1, \lambda \neq 0$ are equivalent as they represent the same Euclidean point. This fact is denoted with $\mathbf{x}_2 \sim \mathbf{x}_1$ in this work. Scaling equivalence is not limited to points but applies to all elements and transformations of projective space. Consequently, an entity of projective space with n elements has at most $n-1$ degrees of freedom.

2.1.2 Projective Hyperplane

The entity dual to a projective point in $\mathbb{P}^n$ is a projective hyperplane. A hyperplane $\boldsymbol{\pi} \in \mathbb{P}^n$ can be represented as a vector containing $n+1$ elements as well:

$$\boldsymbol{\pi} = \begin{pmatrix} \pi_1, \pi_2, \pi_3, \ldots, \pi_n, \pi_{n+1} \end{pmatrix}^\top$$

A projective hyperplane $\boldsymbol{\pi}$ contains a point $\mathbf{x}$ if and only if

$$\mathbf{x}^\top \boldsymbol{\pi} = 0$$

In $\mathbb{P}^2$, which is important for this work, hyperplanes correspond to lines. The line $\mathbf{l}$ joining two points $\mathbf{x}_1$ and $\mathbf{x}_2 \in \mathbb{P}^2$ can be constructed with

$$\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2, \tag{2.1}$$

as the cross-product $\times$ ensures

$$\mathbf{x}_1^\top \mathbf{l} = 0 \quad \text{and} \quad \mathbf{x}_2^\top \mathbf{l} = 0.$$

If the cross product is rewritten as matrix-vector multiplication, equation (2.1) yields

$$\mathbf{l} = [\mathbf{x}_1]_\times \mathbf{x}_2 \quad \text{with} \quad [\mathbf{x}_1]_\times = \begin{bmatrix} 0 & -w_1 & y_1 \\ w_1 & 0 & -x_1 \\ -y_1 & x_1 & 0 \end{bmatrix} . \tag{2.2}$$

2.1.3 Homographies

An advantage of projective space, besides being able to represent points at infinity, is the fact that all collineations, i. e. all line-preserving transformations, can be expressed by matrix-vector multiplications of the form

$$\mathbf{x}' = \mathtt{H}\mathbf{x}. \tag{2.3}$$

In this expression, $\mathtt{H}$ represents a linear transformation $\mathbb{P}^n \to \mathbb{P}^n$ and is called *homography*. The following sections describe a hierarchy of transformations of $\mathbb{P}^n$, each revealing different invariants.

Euclidean Transformations

Euclidean transformations consist of two components: rotation and translation. The corresponding homography has the following form:

$$\mathtt{H}_E = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

Here, $\mathtt{R}$ is a rotation matrix (see also appendix C) and $\mathbf{t}$ is a translation vector. Metric entities like lengths and areas are invariant under a Euclidean transformation.

Similarity Transformations

If isotropic scaling is allowed besides rotation and translation, the resulting transformation is a similarity transformation with a homography

$$\mathtt{H}_S = \begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix},$$

in which s represents a global scaling factor. A similarity transformation retains *ratios* of metric units as well as angles, but it changes Euclidean properties.

Affine Transformations

Affine transformations are a further generalization by including non-isotropic scaling and shearing:

$$\mathtt{H}_A = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

The submatrix $\mathtt{A}$ may contain arbitrary elements, but it must be regular, so that $\det(\mathtt{A}) \neq 0$ (see also section 2.1.4). Parallelity is invariant under an affine transformation. Consequently, as all parallel lines intersect at infinity, the hyperplane at infinity is an invariant subspace.

Projective Transformations

The most general case of a linear transformation $\mathtt{H} : \mathbb{P}^n \rightarrow \mathbb{P}^n$ is represented by a homography

$$\mathtt{H} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{h}^\top & h \end{bmatrix}$$

containing arbitrary elements. Besides collinearity and tangency, the only invariant property is the cross-ratio C of four points $\mathbf{x}_{1,2,3,4}$ of a linear subspace $\mathbb{P}^1$

$$C(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) = \frac{|\mathbf{x}_1\mathbf{x}_2||\mathbf{x}_3\mathbf{x}_4|}{|\mathbf{x}_1\mathbf{x}_3||\mathbf{x}_2\mathbf{x}_4|} \quad \text{, with}$$
$$|\mathbf{x}_i\mathbf{x}_j| = \det \begin{bmatrix} x_{i1} & x_{j1} \\ x_{i2} & x_{j2} \end{bmatrix}. \tag{2.4}$$

Table 2.1 summarizes different groups of transformations.

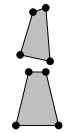
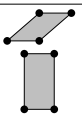
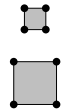
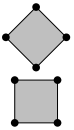
Group	Matrix	Distortion	Invariants
Projective	$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{h}^\top & h \end{bmatrix}$		tangency, collinearity, cross-ratio
Affine	$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$		parallelism, hyperplane at infinity
Similarity	$\begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$		relative metric entities, angles
Euclidean	$\begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$		metric entities (lengths, areas,volumes)

Table 2.1: Summary of different classes of projective transformations, illustrated with distortions of a square in $\mathbb{R}^2$. Each group includes invariants of the ones above.

2.1.4 Projections

For every regular homography $\mathtt{H}, \det(\mathtt{H}) \neq 0$, there is a linear inverse transformation $\mathtt{H}' = \mathtt{H}^{-1}$ as $\mathtt{H}\mathtt{H}' = \mathtt{I}$. If, on the other hand, $\mathtt{H}$ is singular so that $\det(\mathtt{H}) = 0$, it cannot be inverted and represents a projection. In this case, its singular value decomposition (SVD, see appendix B) $\mathtt{H} = \mathtt{U}\Sigma\mathtt{V}^\top$ yields vanishing diagonal elements of Σ. As an example, a projection $\mathtt{H}_P$ in Euclidean space $\mathbb{R}^3$ to a two-dimensional subspace can be decomposed as

$$\mathtt{H}_P = \mathtt{U}\Sigma\mathtt{V}^\top = \mathtt{U}\,\mathrm{diag}(\sigma_1, \sigma_2, 0)\mathtt{V}^\top \tag{2.5}$$

In this example, the third column $\mathbf{v}_3$ of $\mathtt{V} = \begin{bmatrix}\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\end{bmatrix}$ denotes the direction of projection. It is the component of the orthonormal basis of $\mathtt{V}$ which will be suppressed by $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, 0)$.

Therefore, columns of $\mathtt{U}$ corresponding to vanishing singular values may be omitted when performing a projection, reducing the dimension of the range space. Consequently, a singular homography may be written as a linear transformation $\mathbb{P}^n \to \mathbb{P}^m$, $n > m$, if a suitable basis is chosen. This may be advantageous as projection matrices (see section 2.2) can be directly indentified by their shape without computing their rank.

2.2 Pinhole Camera

Camera models describe the imaging process of a camera system mathematically. Therefore, they are able to relate points $\mathbf{X}_E = \begin{pmatrix}\mathrm{X}, \mathrm{Y}, \mathrm{Z}\end{pmatrix}^\top$ in an arbitrary 3D world coordinate system to pixels $\mathbf{x}_E = \begin{pmatrix}x, y\end{pmatrix}^\top$ in a digital image. It is possible to distinguish between two classes of models:

- **Linear camera models**: A linear camera model preserves collinearity so that lines in 3D space are mapped onto lines in an image. The *pinhole camera* is an example of a linear camera model. It has the advantage that it can be completely represented by a single matrix in projective space.
- **Non-linear camera models**: Many effects introduce non-linearities in a projection system, e. g. radial distortion of wide-angle lenses or non-planar mirrors of omnidirectional cameras, among others [Tsa87, GD01]. As this work does not focus on camera models, non-linear effects will not be considered. It is worth mentioning though that in many cases removing non-linear effects of a camera is possible in an independent step by image distortion [TM04]. Afterwards, an equivalent linear camera model may be established.

The pinhole camera model is introduced next. Figure 2.1 shows an illustration. Three Euclidean coordinate systems can be identified: A 3D *world coordinate system* $\begin{pmatrix}\mathrm{X}_W, \mathrm{Y}_W, \mathrm{Z}_W\end{pmatrix}$ in which locations of points in space are described, a 3D *camera coordinate system* $\begin{pmatrix}\mathrm{X}_C, \mathrm{Y}_C, \mathrm{Z}_C\end{pmatrix}$ having its origin at the optical center $\mathbf{C}$ of a camera, and a 2D *image coordinate system* $\begin{pmatrix}x, y\end{pmatrix}$ describing points or pixels in an image.

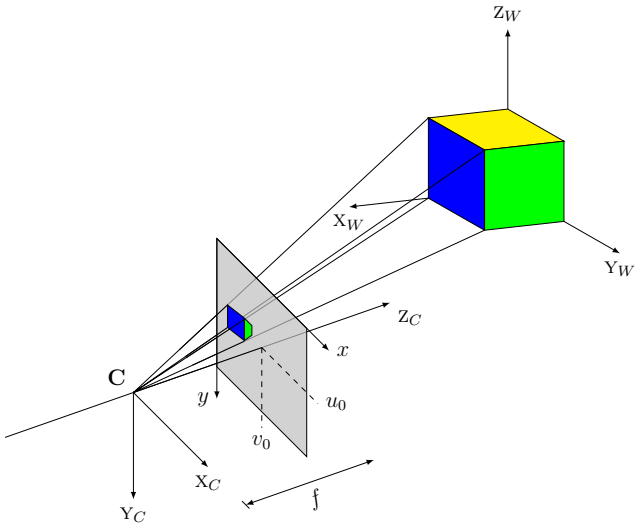


Figure 2.1: Projection with a pinhole camera. In real camera systems the image plane is located behind the optical center **C**. The shown setup is mathematically equivalent.

The imaging process using a pinhole camera can be decomposed into three steps:

1. **External transformation $\mathbb{P}^3 \rightarrow \mathbb{P}^3$:** In the first step, world coordinates $\mathbf{X}_W$ are transferred into camera coordinates $\mathbf{X}_C$. Using homogeneous coordinates, this procedure can be written as a Euclidean transformation:

$$\mathbf{X}_C = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}_W \tag{2.6}$$

   An external transformation only depends on the orientation and location of a camera, also called *pose*, with respect to the world coordinate system. It does not take internal properties of the camera into account. Therefore, the relevant parameters $\mathtt{R}$ und $\mathbf{t}$ are called *external* camera parameters. An external transformation has 6 degrees of freedom: Both $\mathtt{R}$ and $\mathbf{t}$ have three degrees of freedom each.

2. **Canonic perspective projection $\mathbb{P}^3 \rightarrow \mathbb{P}^2$:** Canonic projection follows an external transformation. In this step, Euclidean coordinates in 3D space are considered as

homogeneous 2D-coordinates of the image plane as illustrated by the rays through $\mathbf{C}$ in figure 2.1.

$$\hat{\mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}_C = \mathtt{P}_0 \mathbf{X}_C \tag{2.7}$$

This step is generic, it is common to all perspective projections and does not require any parameters. The projected points $\hat{\mathbf{x}}$ are called *normalized* image points as they originate from a projection, but they do not depend on intrinsic camera properties. Hence, their corresponding Euclidean coordinates still quantify the same units as $\mathbf{X}_W$ and $\mathbf{X}_C$ do in 3D space.

3. **Internal transformation** $\mathbb{P}^2 \to \mathbb{P}^2$: The last step maps normalized coordinates onto image or pixel coordinates. This transformation exclusively depends on internal camera properties. Necessary parameters are two scaling factors α_x and α_y for the x- and y-axis as well as pixel coordinates $\left(x_0, y_0\right)^\top$ describing the intersection of the camera's optical axis and the image plane. Following [HZ03], α_x and α_y can be calculated for a CCD camera given its focal length f and the dimensions $k_x \times k_y$ of a CCD sensor element with $\alpha_x = \frac{f}{k_x}$, $\alpha_y = \frac{f}{k_y}$. In $\mathbb{P}^2$ an internal transformation can be written by using an affine homography $\mathtt{K}$

$$\mathbf{x} = \begin{bmatrix} \alpha_x & s_k & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{x}} = \mathtt{K}\hat{\mathbf{x}} \tag{2.8}$$

In this notation, an additional component s_k accounts for shearing of the image's coordinate axes. However, CCD sensor elements are usually aligned in a rectangular grid structure so that $s_k = 0$ [HZ03]. It can be seen that $\mathtt{K}$ is an upper triangular matrix.

An internal transformation has 5 degrees of freedom.

It is possible to combine the three introduced steps to a single transformation matrix $\mathtt{P}$ that contains external and internal camera parameters:

$$\mathtt{P} = \begin{bmatrix} \alpha_x & s_k & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} = \mathtt{K}\left[\mathtt{R} \mid \mathbf{t}\right] \tag{2.9}$$

$$= \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \tag{2.10}$$

$$= \left[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\right] = \left[\mathtt{M} \mid \mathbf{p}_4\right] \tag{2.11}$$

$\mathtt{P}$ is called *projection matrix*, whereas $\mathtt{K}$ and $\left[\mathtt{R} \mid \mathbf{t}\right]$ are called *external* and *internal camera matrix* accordingly. $\mathtt{P}$ describes a mapping $\mathbb{P}^3 \rightarrow \mathbb{P}^2$ and has 12 elements. However, due to scaling invariance, it has 11 degrees of freedom. This is consistent with the 6 external and 5 internal degrees of freedom of $\left[\mathtt{R} \mid \mathbf{t}\right]$ and $\mathtt{K}$.

2.3 Camera Calibration

The projection matrix $\mathtt{P}$ of a camera system is generally unknown. However, it can be determined by a set of given 2D/3D-point correspondences. This process is called camera calibration. Once a camera is calibrated, it is possible to determine its pose with respect to the world coordinate system or to other calibrated cameras (see section 2.3.2). For this procedure, a calibration object with known 3D points is needed and their corresponding projections in an image have to be identified.

2.3.1 Direct Linear Transformation (DLT)

A single 3D-point $\mathbf{X}$ and its corresponding 2D-position $\mathbf{x}$ in an image yield the following projection equation, including an arbitrary scaling factor $\lambda \neq 0$ explicitly denoting the overall scale ambiguity

$$\mathbf{x} \sim \mathtt{P}\mathbf{X} \Leftrightarrow \begin{pmatrix} \lambda x \\ \lambda y \\ \lambda \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} \tag{2.12}$$

Rewriting (2.12) and substituting the unknown parameter λ results in two equations

$$\begin{aligned} (\mathrm{X}p_{31} + \mathrm{Y}p_{32} + \mathrm{Z}p_{33} + p_{34})x &= \mathrm{X}p_{11} + \mathrm{Y}p_{12} + \mathrm{Z}p_{13} + p_{14} \\ (\mathrm{X}p_{31} + \mathrm{Y}p_{32} + \mathrm{Z}p_{33} + p_{34})y &= \mathrm{X}p_{21} + \mathrm{Y}p_{22} + \mathrm{Z}p_{23} + p_{24} \end{aligned}$$

By introducing a vector $\mathbf{p} = \left(p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, \ldots, p_{33}, p_{34}\right)^\top$ containing all elements of $\mathtt{P}$, a block of a linear, homogeneous equation system

$$\mathtt{A}\mathbf{p} = \mathbf{0} \tag{2.13}$$

can be given:

$$\begin{bmatrix} -\mathrm{X} & -\mathrm{Y} & -\mathrm{Z} & -1 & 0 & 0 & 0 & 0 & \mathrm{X}x & \mathrm{Y}x & \mathrm{Z}x & x \\ 0 & 0 & 0 & 0 & -\mathrm{X} & -\mathrm{Y} & -\mathrm{Z} & -1 & \mathrm{X}y & \mathrm{Y}y & \mathrm{Z}y & y \end{bmatrix} \begin{pmatrix} p_{11} \\ p_{12} \\ p_{13} \\ \vdots \\ p_{33} \\ p_{34} \end{pmatrix} = \mathbf{0} \tag{2.14}$$

Blocks for several 2D/3D point correspondences of the form (2.14) may be combined by concatenating rows of $\mathtt{A}$. For at least $N = 6$ correspondences, the resulting equation system

$$\begin{bmatrix} -\mathrm{X}_1 & -\mathrm{Y}_1 & -\mathrm{Z}_1 & -1 & 0 & 0 & 0 & 0 & \mathrm{X}_1x & \mathrm{Y}_1x_1 & \mathrm{Z}_1x_1 & x_1 \\ 0 & 0 & 0 & 0 & -\mathrm{X}_1 & -\mathrm{Y}_1 & -\mathrm{Z}_1 & -1 & \mathrm{X}_1y & \mathrm{Y}_1y_1 & \mathrm{Z}_1y_1 & y_1 \\ -\mathrm{X}_2 & -\mathrm{Y}_2 & -\mathrm{Z}_2 & -1 & 0 & 0 & 0 & 0 & \mathrm{X}_2x & \mathrm{Y}_2x_2 & \mathrm{Z}_2x_2 & x_2 \\ 0 & 0 & 0 & 0 & -\mathrm{X}_2 & -\mathrm{Y}_2 & -\mathrm{Z}_2 & -1 & \mathrm{X}_2y & \mathrm{Y}_2y_2 & \mathrm{Z}_2y_2 & y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathrm{X}_N & -\mathrm{Y}_N & -\mathrm{Z}_N & -1 & 0 & 0 & 0 & 0 & \mathrm{X}_Nx & \mathrm{Y}_Nx_N & \mathrm{Z}_Nx_N & x_N \\ 0 & 0 & 0 & 0 & -\mathrm{X}_N & -\mathrm{Y}_N & -\mathrm{Z}_N & -1 & \mathrm{X}_Ny & \mathrm{Y}_Ny_N & \mathrm{Z}_Ny_N & y_N \end{bmatrix} \begin{pmatrix} p_{11} \\ p_{12} \\ p_{13} \\ \vdots \\ p_{33} \\ p_{34} \end{pmatrix} = \mathbf{0} \tag{2.15}$$

is not underdetermined anymore as the number of equations exceeds the degrees of freedom. However, as equation (2.15) is homogeneous, there is a continuum of solutions. A unique solution with $\|\mathbf{p}\| = 1$ can be found as the last column of $\mathtt{V}$ of a singular value decomposition of $\mathtt{A}$ (see equation (B.1)). This technique is called *direct linear transformation* (DLT) and will also be used for obtaining other projective entities (see section 2.5).

2.3.2 Extraction of Camera Parameters after Calibration

If a projection matrix $\mathtt{P} = \begin{bmatrix} \mathtt{M} \mid \mathbf{p}_4 \end{bmatrix}$ is known for a camera, internal and external camera parameters, i. e. $\mathtt{K}$ and $\begin{bmatrix} \mathtt{R} \mid \mathbf{t} \end{bmatrix}$ may be recovered. This is possible by RQ-decomposition of $\mathtt{M}$ due to the fact that $\mathtt{R}$ is orthogonal and $\mathtt{K}$ is an upper triangular matrix [HZ03]. Decomposing a projection matrix $\mathtt{P}$ into external and internal parameters can particularly be used to solve two problems:

1. **Determine the internal camera matrix $\mathtt{K}$ for a complete sequence of images (intrinsic calibration)**: If a single camera moves in 3D space, its internal parameters, the elements of $\mathtt{K}$, remain constant. Therefore, it is possible to estimate $\mathtt{K}$ with a calibration object so that for subsequent images the external parameters $\begin{bmatrix} \mathtt{R} \mid \mathbf{t} \end{bmatrix}$ are the only remaining unknowns.

2. **Determine the relative pose of two cameras**: If projection matrices $\mathtt{P}$ and $\mathtt{P}'$ have been estimated for two cameras looking at the same scene, the orientation and position of the second with respect to the first can be computed with a Euclidean transformation

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

so that

$$\begin{aligned} \mathtt{R}_{1\to 2} &= \mathtt{R}'\mathtt{R}^\top \quad \text{und} \\ \mathbf{t}_{1\to 2} &= -\mathtt{R}'\mathtt{R}^\top\mathbf{t} + \mathbf{t}' \end{aligned}$$

If the world coordinate system coincides with the camera coordinate system of the left camera, then $\mathtt{R} = \mathtt{I}$ and $\mathbf{t} = \mathbf{0}$ so that the left and right projection matrix can be given as

$$\mathtt{P} = \mathtt{K}\begin{bmatrix}\mathtt{I} \mid \mathbf{0}\end{bmatrix} \quad \text{and} \quad \mathtt{P}' = \mathtt{K}'\begin{bmatrix}\mathtt{R}' \mid \mathbf{t}'\end{bmatrix} \tag{2.16}$$

2.4 Stereo Geometry

This section describes relative pose estimation of two cameras *without* known 3D positions, i. e. by looking at an *unknown* scene. In this case, 3D/2D correspondences are not available and only 2D/2D correspondences between image points can be gathered.

2.4.1 Known Internal Camera Parameters

Section 2.3.2 has shown how to determine internal camera matrices $\mathtt{K}$, $\mathtt{K}'$ of two cameras by using a calibration object. If this has been done for the used cameras, subsequent calculations can be done in normalized coordinates so that the projection matrices contain external camera parameters only. As the scene is unknown, the left camera coordinate system can be chosen as world frame (see equation 2.16) so that

$$\hat{\mathtt{P}} = \begin{bmatrix}\mathtt{I} \mid \mathbf{0}\end{bmatrix} \quad \text{and} \quad \hat{\mathtt{P}}' = \begin{bmatrix}\mathtt{R}' \mid \mathbf{t}'\end{bmatrix}.$$

Figure 2.2 shows a setup of two cameras, also called stereo setup. It can be seen how two views of a scene are related geometrically: An image point $\hat{\mathbf{x}} = \hat{\mathtt{P}}\mathbf{X}$ in the left image originates from an unknown 3D point $\mathbf{X}$. According to the pinhole camera model, $\mathbf{X}$ must be on a ray $\mathcal{L}$ through $\hat{\mathbf{x}}$ and the left camera center $\mathbf{C}$. In the right camera, $\mathcal{L}$ is projected onto the *epipolar line* $\hat{\mathbf{l}}'$. Hence, the point correspondence of $\hat{\mathbf{x}}$ in the right image, $\hat{\mathbf{x}}' = \hat{\mathtt{P}}'\mathbf{X}$, must be on $\hat{\mathbf{l}}'$ as well.

As all rays of the left camera contain its optical center $\mathbf{C}$, all epipolar lines in the right image contain the projection of $\mathbf{C}$, the *epipole* $\hat{\mathbf{e}}' = \hat{\mathtt{P}}'\mathbf{C} = \hat{\mathtt{P}}'\left(0,0,0,1\right)^\top$.

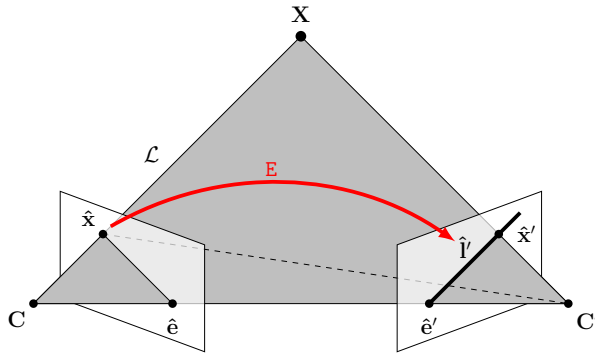


Figure 2.2: Epipolar geometry of a stereo camera system

The epipolar line $\hat{\mathbf{l}}'$ of a normalized point $\hat{\mathbf{x}}$ can be constructed given the cameras' relative pose as $\mathtt{R}'$ and $\mathbf{t}'$. According to equation (2.1) two points on $\hat{\mathbf{l}}'$ are sufficient. They can be chosen as the epipole $\hat{\mathbf{e}}'$ and the projection of an arbitrary second point on $\mathcal{L}$, e. g. the location $\hat{\mathbf{X}} = \left(\hat{\mathbf{x}}, 1\right)^\top$ of $\hat{\mathbf{x}}$ in 3D space. Hence, $\hat{\mathbf{l}}'$ can be written as

$$\hat{\mathbf{l}}' = \hat{\mathbf{e}}' \times \hat{\mathtt{P}}'\hat{\mathbf{X}} = \hat{\mathtt{P}}' \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} \times \hat{\mathtt{P}}' \begin{pmatrix} \hat{\mathbf{x}} \\ 1 \end{pmatrix} = \mathbf{t}' \times (\mathtt{R}'\hat{\mathbf{x}} + \mathbf{t}') = \mathbf{t}' \times \mathtt{R}'\hat{\mathbf{x}} = [\mathbf{t}']_\times \mathtt{R}'\hat{\mathbf{x}} \tag{2.17}$$

As the epipolar line $\hat{\mathbf{l}}'$ contains $\hat{\mathbf{x}}'$, the following equation holds:

$$\hat{\mathbf{x}}'^\top \hat{\mathbf{l}}' = 0 \quad \Leftrightarrow \quad \hat{\mathbf{x}}'^\top \mathtt{E} \hat{\mathbf{x}} = 0 \tag{2.18}$$

$$\text{where} \quad \mathtt{E} = [\mathbf{t}']_\times \mathtt{R}' \tag{2.19}$$

Equation (2.18), also called *correspondence condition*, is of fundamental importance for the rest of this work. Here, $\mathtt{E}$ is called the *essential matrix*. It can be constructed given $\mathtt{R}'$ and $\mathbf{t}'$, both having three degrees of freedom. However, as $\mathtt{E}$ is invariant to scaling, it only has five degrees of freedom. An essential matrix determines the relative pose of two cameras up to a global scaling factor, thus, it does not account for the magnitude $\|\mathbf{t}'\|$. Finally, it can be seen that $\mathtt{E}$ is singular, since $[\mathbf{t}']_\times$ is. In particular, its singular value decomposition (SVD, see appendix B) yields

$$\mathtt{E} = \mathtt{U}\Sigma\mathtt{V}^\top \sim \mathtt{U}\,\mathrm{diag}(1, 1, 0)\mathtt{V}^\top \tag{2.20}$$

Hence, an essential matrix has two identical and one vanishing singular value. According

to [May93], this constraint can also be written as a cubic equation

$$\mathtt{E}\mathtt{E}^\top\mathtt{E} - \frac{1}{2}\,\mathrm{trace}(\mathtt{E}\mathtt{E}^\top)\mathtt{E} = \mathtt{0} \tag{2.21}$$

The SVD of an essential matrix can also be used to determine epipoles. Faugeras has shown [FLS92] that the left and right null-space of an essential matrix $\mathtt{E}$, i. e. the third columns $\mathbf{u}_3$, $\mathbf{v}_3$ of $\mathtt{U}$ and $\mathtt{V}$, represent epipoles due to

$$\mathtt{E}\hat{\mathbf{e}} = \mathbf{0} \quad \text{and} \quad \mathtt{E}^\top\hat{\mathbf{e}}' = \mathbf{0}. \tag{2.22}$$

Equation (2.22) states that the epipolar line corresponding to an epipole is undefined. This is obvious as, according to (2.17), an epipolar line $\hat{\mathbf{l}}'$ is determined by *two different* points on $\hat{\mathbf{l}}'$, the epipole $\hat{\mathbf{e}}'$ and a second one.

As an essential matrix is singular, it can be considered as a projection according to section 2.1.4. More insight into this aspect will be given in sections 3.1 and 3.3.

2.4.2 Unknown Internal Camera Parameters

The correspondence condition (2.18) can also be transferred to an uncalibrated setup, i. e. to a system with unknown intrinsic camera matrices $\mathtt{K}$, $\mathtt{K}'$. In this case, normalized point coordinates are not available. Due to equation (2.8), the correspondence condition for unnormalized corresponding image points $\mathbf{x}$ and $\mathbf{x}'$ can be rewritten as

$$\mathbf{x}'^\top\mathtt{K}'^{-\top}\mathtt{E}\mathtt{K}^{-1}\mathbf{x} = \mathbf{x}'^\top\mathtt{F}\mathbf{x} = 0 \tag{2.23}$$

$$\text{with} \qquad \mathtt{F} = \mathtt{K}'^{-\top}\mathtt{E}\mathtt{K}^{-1} \tag{2.24}$$

$\mathtt{F}$ is called *fundamental matrix* and is equivalent to the essential matrix in an uncalibrated setting. Hence, equation (2.23) directly relates corresponding pixels. Besides being invariant to scaling, a fundamental matrix is of rank 2 as it contains $\mathtt{E}$ as a factor:

$$\det(\mathtt{F}) = 0 \tag{2.25}$$

Its singular value decomposition is of the form

$$\mathtt{F} = \mathtt{U}\mathtt{\Sigma}\mathtt{V}^\top = \mathtt{U}\,\mathrm{diag}(\sigma_1, \sigma_2, 0)\mathtt{V}^\top \tag{2.26}$$

Similarly to equation (2.22), epipoles can be determined as the left and right null-space of a fundamental matrix:

$$\mathtt{F}\mathbf{e} = \mathbf{0} \quad \text{and} \quad \mathtt{F}^\top\mathbf{e}' = \mathbf{0}. \tag{2.27}$$

Obviously, an essential matrix can be considered as a special fundamental matrix for $\mathtt{K} = \mathtt{K}' = \mathtt{I}$.

2.5 Weak Calibration and Relative Pose Estimation

This section describes how the matrices F and E can be computed given a set of corresponding 2D/2D points. It will be shown that this step corresponds to the estimation of P when using a calibration object. If no further information besides corresponding points is available, this procedure is called *weak* calibration, as the relative pose of the two cameras is estimated up to an arbitrary projective transformation. If internal camera parameters are known, the essential matrix may be estimated instead. It will be shown below that the true relative pose of the two cameras may be identified up to scale.

2.5.1 Fundamental Matrix Estimation

Analogous to equation (2.14), the correspondence condition (2.23) can be written as

$$f_{11}x'x + f_{12}x'y + f_{13}x'z + f_{21}y'x + f_{22}y'y + f_{23}y'z + f_{31}z'x + f_{32}z'y + f_{33}z'z = 0 \quad (2.28)$$

It can be seen that, in contrast to section 2.3, there is *one* equation per correspondence. Hence, given at least $N = 8$ point correspondences, the DLT method can be applied to an equation system $\mathtt{A}\mathbf{f} = \mathbf{0}$, in which $\mathbf{f} = \left(f_{11}, f_{12}, f_{13}, f_{21}, \ldots, f_{33}\right)^\top$ contains the elements of F. This algorithm is called *eight-point* algorithm. It has been introduced by Longuet-Higgins [LH81] and defended by Hartley [Har97].

A linear solution yields an estimate of a fundamental matrix that does not necessarily satisfy the singularity constraint (2.25). It is possible to obtain a closest rank-2 approximation. If the Frobenius norm is chosen, such a solution can be computed by setting the smallest singular value of the linear estimate of F to zero [HZ03]. However, this solution may not be optimal and can be refined (see section 2.6.2).

For $N \geq 8$, a unique solution via DLT cannot be found if A is of rank $q < 8$, i. e. if the rows of A are linearly dependent. It has been shown that this is the case if all point correspondences can be found on a ruled quadric passing to both camera centers [TZ00]. Degenerate configurations of point correspondences will not be considered in this work.

Other approaches use a minimum of 7 point correspondences by utilizing the additional constraint $\det(\mathtt{F}) = 0$, but there may generally be up to three solutions as they are obtained by finding roots of a cubic equation [Har94].

2.5.2 Essential Matrix Estimation

If internal camera parameters are known, the essential matrix can be estimated using the eight-point algorithm as well. However, the same aspects of section 2.5.1 apply: A linear estimate will not necessarily satisfy additional non-linear constraints such as the cubic equation (2.21). Again, either an approximation has to be found, e. g. by enforcing the singular values of eq. (2.20) [HHLM04], or by using fewer point correspondences.

A linear solution based on six correspondences has been introduced by Philip [Phi98] and reviewed in [Nis04]. In this case, the DLT method yields three vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$

spanning the null-space of $\mathtt{A_e}\mathbf{e}_{1,2,3} = 0$. Hence, any matrix of the form $\mathtt{E}_{xyz} = x\mathtt{E}_1 + y\mathtt{E}_2 + z\mathtt{E}_3$ will satisfy the correspondence condition (2.18).

By inserting $\mathtt{E}_{xyz}$ into the cubic constraint (2.21), a second system of the form (2.15) can be established:

$$\mathtt{A}_\mathtt{E}\left(y^3, y^2z, yz^2, z^3, y^2x, yzx, z^2x, yx^2, zx^2, x^3\right)^\top = \mathbf{0} \tag{2.29}$$

The solution via DLT yields the null-space of $\mathtt{A}_\mathtt{E}$ so that x, y, z can be determined.

A unique solution for the minimum number of correspondences may not be obtained. Kruppa has shown that 11 possible solutions exist for $N = 5$ [NS04].

2.5.3 Extraction of Camera Parameters

This section describes the derivation of camera parameters in a weakly calibrated camera setup. It has been mentioned in section 2.4.1 that in the case of an unknown scene content, the left camera coordinate system can be chosen as the world frame so that

$$\mathtt{R} = \mathtt{I} \quad \text{and} \quad \mathbf{t} = \mathbf{0}. \tag{2.30}$$

Therefore, the problem of obtaining camera parameters of a weakly calibrated stereo setup can be described as follows:

- Intrinsically calibrated case: Given the internal camera matrices $\mathtt{K}$, $\mathtt{K}'$ and an essential matrix $\mathtt{E}$, find the rotation $\mathtt{R}'$ and translation $\mathbf{t}'$ of the second camera with respect to the first.
- Intrinsically uncalibrated case: Given a fundamental matrix $\mathtt{F}$, find the rotation $\mathtt{R}'$ and translation $\mathbf{t}'$ as well as the intrinsic camera matrices $\mathtt{K}$, $\mathtt{K}'$.

It will be demonstrated in this section that a unique solution for the latter case may not be obtained.

Known Internal Camera Parameters

Following [HZ03], the singular value decomposition of an essential matrix (2.20) yields four solutions for the external camera parameters

$$\mathtt{R}'_1 = \mathtt{U}\mathtt{W}\mathtt{V}^\top, \qquad \mathtt{R}'_2 = \mathtt{U}\mathtt{W}^\top\mathtt{V}^\top \tag{2.31}$$

$$\mathbf{t}'_1 = \mathbf{u}_3, \qquad \mathbf{t}'_2 = -\mathbf{u}_3 \tag{2.32}$$

with a matrix

$$\mathtt{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.33}$$

that rotates the image plane by 90° around its optical axis. An illustration of the four solutions is shown in figure 2.3. The ambiguity with respect to $\mathtt{R}'_1$ and $\mathtt{R}'_2$ is called *twisted pair* (fig. 2.3(a),(c)), whereas the two solutions for the translation vector are denoted by *baseline reversal* (fig. 2.3(a),(b)). It can be seen that there is only one case (fig. 2.3(a)) in which a 3D point is located in front of the two cameras. Hence, the correct set of external camera parameters $\mathtt{R}'$, $\mathbf{t}'$ may be identified via triangulation of a single arbitrary point correspondence [HZ03].

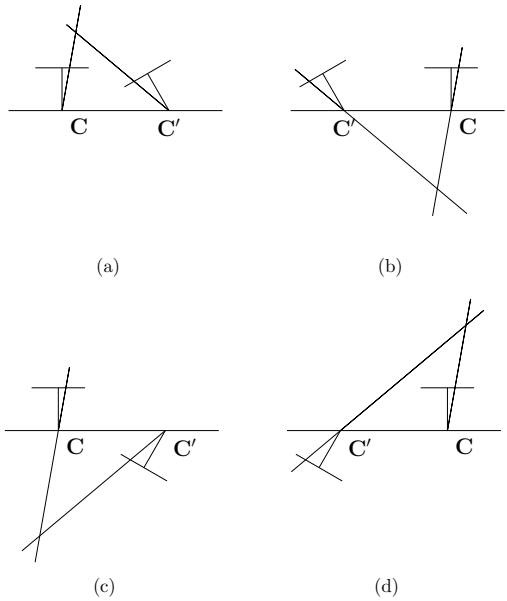


Figure 2.3: Multiplicity of solutions for a weakly calibrated stereo setup with known internal camera parameters: Only situation (a) yields world points in front of both cameras.

A different decomposition of the essential matrix will be introduced in section 3.3.4.

Unknown Internal Camera Parameters

In case of an uncalibrated setup, the seven degrees of freedom of a fundamental matrix $\mathtt{F}$ cannot describe the sixteen unknown parameters of $\mathtt{K}$, $\mathtt{K}'$ and $\left[\mathtt{R}' \mid \mathbf{t}'\right]$. In particular, as shown in [HZ03], one possible solution of the two projection matrices $\mathtt{P}$ and $\mathtt{P}'$ can be given

as

$$\mathtt{P} = \left[\mathtt{I} \mid \mathbf{0}\right] \quad \text{and} \quad \mathtt{P}' = \left[\left[\mathbf{e}'\right]_\times \mathtt{F} \mid \mathbf{e}'\right] \tag{2.34}$$

Other camera matrices may be obtained from $\mathtt{P}$ and $\mathtt{P}'$ by multiplication with a common homography $\mathtt{H} : \mathbb{P}^3 \to \mathbb{P}^3$. $\mathtt{HP}$ and $\mathtt{HP}'$ still describe the same relative geometry, hence they are compliant with a given fundamental matrix $\mathtt{F}$ as well.

2.6 Estimation Techniques for Epipolar Geometry

Corresponding points in two images are not known a priori. They can be estimated in different ways. However, the accuracy of point correspondences is limited by two aspects. Firstly, estimation is not possible with infinite precision so that correctly matched correspondences suffer from *noise*. Secondly, point matches may be *outliers*, i. e. they may be incorrectly matched. These two types of errors have to be approached differently. This section first lays out effects of noisy data for weak calibration. Subsection 2.6.3 describes the case in which correspondences have to be separated into inliers and outliers and discusses *robust* estimation techniques that have to be used in this case.

2.6.1 Estimating Point Correspondences

This work assumes point correspondences to be estimated in a prior step. In this section, a brief overview of possible techniques will be given. Surveys and more details can be found in the literature [Smi97, Tek95, BFBB92].

Sparse Methods

It is possible to reduce the problem of estimating point correspondences to pixels or regions in the two images that provide special features like high intensity gradients or textures. Finding a sparse set of point correspondences is usually implemented in a two-stage process [HZ03, Pol99]:

1. Detect and localize a set of feature points in both images such as *corners* by using Harris' approach [HS88] or the SUSAN algorithm by Smith [SB95].

2. Find matches by correlating small image regions around feature points meeting displacement constraints.

If more than two successive frames of a video sequence are analyzed, the search for point correspondences may additionally include temporal aspects like tracking [ST94].

On the one hand, sparse methods for estimating point correspondences have the advantage of reducing the amount of data to be processed in later steps. Also, areas with no structure are not considered as they do not provide features. On the other hand, sparse

methods generally do not evaluate the spatial coherence of adjacent correspondences so that false matches may occur more easily.

For the method introduced in chapter 4, the approach of Shi and Tomasi is used without tracking for establishing sparse point correspondences [ST94].

Dense Methods

Point correspondences may also be estimated for every pixel or block in an image. This task is identical to finding motion vectors describing the spatial difference between correspondences. Algorithms for estimating dense motion fields can be categorized into two different approaches:

1. *Gradient-based, time-differential motion estimation*: A reasonable assumption for corresponding points is the brightness constancy constraint equation (BCCE): It states that the intensity $I(\mathbf{x})$ of a projected 3D point stays constant accross images. The BCCE has first been analyzed by Horn and Schunk [Hor86]. A first order approximation has been used to find the *optical flow* of an image, the dense 2D velocity field

$$\mathbf{v}(\mathbf{x}) = \frac{d\mathbf{x}_E}{dt}$$

by

$$\mathbf{v}(\mathbf{x}) = -\frac{\partial I}{\partial t}\frac{\nabla I}{\|\nabla I\|^2}, \quad \nabla I = \begin{pmatrix} \partial I/\partial x \\ \partial I/\partial y \end{pmatrix} \tag{2.35}$$

Obviously, instantaneous velocities are identical to discrete motion vectors only in the case of *constant motion*. Also, due to equation (2.30), $\mathbf{v}$ is identical to the projected 3D velocity $\mathbf{V}_E$ of a point $\mathbf{X}_E$ in camera coordinates

$$\mathbf{V}_E = \frac{\partial \mathbf{X}_E}{\partial t}. \tag{2.36}$$

Without modifications, the quality of the estimated velocity field is poor due to two main effects: Firstly, the dominating presence of noise in areas without high intensity gradients, and secondly, the insufficiency of the linear flow model (2.35) for highly varying intensity gradients. Hence, optical flow algorithms usually include smoothness terms [BBPW04]. This way, the constraint of constant motion is relaxed.

2. *Correlation-based, time-discrete motion estimation*: Discrete motion vectors

$$\mathbf{u}(\mathbf{x}) = \mathbf{x}' - \mathbf{x}$$

for two corresponding points $\mathbf{x}$ and $\mathbf{x}'$ can for example be estimated by correlating block-sized regions. While being fast, block matching algorithms generally imply a motion model for blocks that only accounts for translation. Smoothness terms may also be included in an estimation algorithm to account for spatial coherence so that the problem of estimating motion vectors in regions without image features is better conditioned [dH93].

For this work, a block-based motion estimation algorithm has been developed [vdHWG06]. It will be used for the estimation of dense motion fields in chapter 5.

2.6.2 Unexact correspondences

It has been mentioned above that establishing *perfect* point correspondences is not possible. There are many reasons for it:

- **Finite resolution:** All digital images have finite resolution and, due to Shannon's theorem, contain insufficient information to obtain exact positions [Sha48].
- **Physical aspects:** Every imaging device suffers from various sources of noise [Jan01, Kam97]. As a consequence, intensities in an image do not represent true values. Also, lighting aspects like specular lights, non-lambertian surfaces of captured objects as well as shadows may lead to erroneous point matches [Tek95].
- **Accuracy of correspondence estimators:** Many correspondence or motion estimation algorithms favor processing speed er accuracy and therefore limit the space of motion vectors to e. g. integer values [dH93].

False correspondences, i. e. outliers, may occur due to similar structures in the two images.

Section 2.3 and 2.5 have presented the DLT algorithm for obtaining a linear estimate of the projection, the fundamental, or the essential matrix by solving an equation system of the form (2.15) via SVD. If point correspondences are subject to noise, `A` will generally not be singular anymore so that equation (2.15) cannot be solved. The linear estimate remains an approximation.

The following paragraphs revisit weak camera calibration in the context of real, i. e. noisy and false correspondences.

Optimal linear Estimation

The residual error of a linear estimate is subject to the values of `A`, which contains linear and bilinear combinations of point correspondences. In order to improve the quality of a linear solution, point correspondences can be *prenormalized* by transforming the set of point correspondences prior to weak calibration [Har97]. In this approach, affine transformations

$\mathbf{x}_{ni} = \mathtt{H}_N\mathbf{x}_i$ and $\mathbf{x}'_{ni} = \mathtt{H}'_N\mathbf{x}'_i$ for both images are defined such that the centroid of both sets of points $\{\mathbf{x}_{ni}\}$ and $\{\mathbf{x}'_{ni}\}$ coincides with the origin in transformed coordinates. Also, point positions are scaled such that their mean Euclidean distance to the origin equals $\sqrt{2}$. In the following, point correspondences $\{\mathbf{x}_{ni}, \mathbf{x}'_{ni}\}$ are called *prenormalized*. After weak calibration, the fundamental matrix $\mathtt{F}$ can be computed with

$$\mathtt{F} = \mathtt{H}'^{\top}_N \mathtt{F}_N \mathtt{H}_N, \tag{2.37}$$

where $\mathtt{F}_N$ is the linear estimate using the set of prenormalized correspondences. Appendix B gives some more details about the necessity for prenormalization.

Non-linear Optimization

A linear solution may be refined by non-linear optimization techniques due to the fact that the DLT-algorithm minimizes an *algebraic* error rather than a geometric or statistic one.

Non-linear optimization techniques iteratively modify a set of parameters such that an error of a given cost function

$$\rho = f(\rho_i) \stackrel{!}{=} \min \quad , \; i = 1 \ldots N \tag{2.38}$$

is minimized. It consists of individual error terms ρ_i for each of the N correspondences. A least-squares error for example can be written as

$$\rho_{\mathrm{LS}} = \sum_{i=1}^{N} \rho_i^2 \tag{2.39}$$

For the calibration techniques introduced in section 2.3 and 2.5, the set of parameters which is optimized must be able to represent the projection, the essential, or the fundamental matrix with respect to their constraints (2.21) and (2.25). Section 3.3 discusses possible existing and new parametrizations for weak calibration.

Suitable cost functions ρ for calibration techniques are based on *measurement* errors, i. e. they include the distance d of a measured position $\mathbf{x}_i$ to an optimal position $\mathbf{x}^*_i$. Therefore, the Euclidean distance

$$d_2(\mathbf{x}_i, \mathbf{x}^*_i) = \|\mathbf{x}_{iE} - \mathbf{x}^*_{iE}\|_2$$

can be used. If a normally distributed error model is applicable for measurements, a statistically optimal distance term can be given as the Mahalanobis distance

$$d_{\mathtt{C}}(\mathbf{x}_{iE}, \mathbf{x}^*_{iE}) = \sqrt{(\mathbf{x}_{iE} - \mathbf{x}^*_{iE})^{\top} \mathtt{C}^{-1} (\mathbf{x}_{iE} - \mathbf{x}^*_{iE})},$$

in which $\mathtt{C}$ is the error's covariance matrix [HZ03]. Another error term may be the Euclidean distance of a point $\mathbf{x}'_i$ to a corresponding epipolar line $\mathbf{l}_i = \left(l_{x,i}, l_{y,i}, l_{z,i}\right)^\top = \mathtt{F}\mathbf{x}_i$

$$d(\mathbf{x}'_i, \mathbf{l}_i) = \left|\mathbf{x}_i'^\top \mathbf{l}_i\right| / \left(\left|\left(l_{x,i}, l_{y,i}\right)\right| |w|\right) \tag{2.40}$$

Table 2.2 summarizes possible error terms as well as the entities to be optimized.

Objective	Error term for one correspondence	Entities to be optimized
Camera calibration		
Calibration error	$\rho_i = d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i^*)$	$\mathtt{P}$
Weak calibration - unknown internal camera matrices $\mathtt{K}, \mathtt{K}'$		
Geometric Error	$\rho_i = d(\mathbf{x}_i, \mathtt{F}^\top\mathbf{x}'_i) + d(\mathbf{x}'_i, \mathtt{F}\mathbf{x}_i)$	$\mathtt{F}$
Reprojection Error	$\rho_i = d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i) + d(\mathbf{x}'_i, \mathtt{P}'\mathbf{X}_i)$	$\mathtt{F}$, $\{\mathbf{X}_i\}$, see also eq. (2.34)
Pose estimation - known internal camera matrices $\mathtt{K}, \mathtt{K}'$		
Geometric Error	$\rho_i = d(\mathbf{x}_i, \mathtt{F}^\top\mathbf{x}'_i) + d(\mathbf{x}'_i, \mathtt{F}\mathbf{x}_i)$	$\mathtt{E}$, see also eq. (2.23)
Reprojection Error	$\rho_i = d(\mathbf{x}_i, \mathtt{P}\mathbf{X}'_i) + d(\mathbf{x}'_i, \mathtt{P}\mathbf{X}_i)$	$\mathtt{E}$, $\{\mathbf{X}_i\}$, see also eq. (2.34)

Table 2.2: Error terms and optimization parameters for different calibration techniques

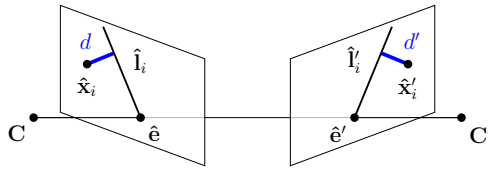
Two aspects are worth mentioning:

- In contrast to camera calibration, where known 3D points $\{\mathbf{X}_i^*\}$ are defined by a calibration object, the 3D locations of point correspondences $\{\mathbf{X}_i\}$ are unknown in the case of weak calibration. If they are not considered during optimization, the distance of a point to its corresponding epipolar line (2.40) may be used as a geometric error term. However, $\{\mathbf{X}_i\}$ may be included in optimization, leading to a Gold Standard error function [HZ03]. In this case, ρ is called *reprojection error*. Figure 2.4 illustrates both error terms.

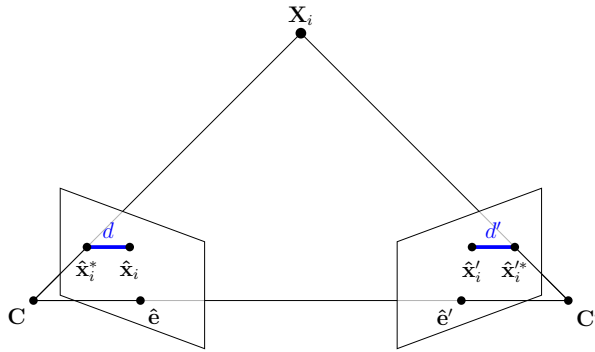
- If cameras have been intrinsically calibrated, computations can completely be done in normalized coordinates. However, measurement errors occur in the *images* so that a Gold Standard cost function has to operate with distance terms in pixel coordinates.

Non-linear optimization algorithms for solving equation (2.38) can be found in [PFTV86, BSS93]. As an example, the Levenberg-Marquardt algorithm is a least-squares optimizer with an error function (2.39) that is commonly used for 3D computer vision applications [HZ03].

A major drawback of non-linear optimization algorithms is that a globally optimal result may not be found. Instead, estimated parameters may converge to a local optimum only, so that they must be initialized with sufficiently good starting values, which this work aims at.



(a) Geometric error: An error term denotes the distance to epipolar lines



(b) Reprojection error: An error term denotes the distance to reprojected world points

Figure 2.4: Geometric and reprojection error

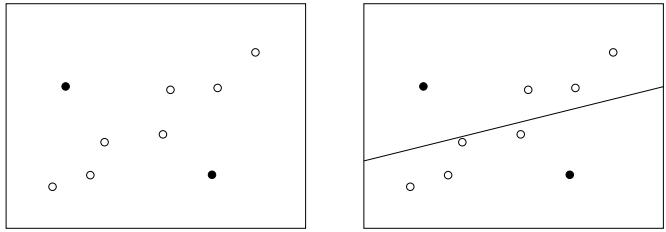
2.6.3 Robust Estimation

This section discusses estimation techniques in the presence of *outliers*. In this case, least squares error techniques fail as a single outlier yields a large error term that biases the optimal estimate. Figure 2.5 shows an example situation in which a line is fitted to a set of 2D points by minimizing a least-squares error function ρ_{LS}. An overview of robust estimation techniques for computer vision can be found in [Ste99] and [Mee04] and will be summarized below.

M-Estimation

Cost functions of least-squares estimation techniques may be modified such that an individual error term ρ_i is bounded [Hub64]. In this case, the influence of outliers can be limited. Example cost functions are listed in table 2.3.

It can be seen that M-estimators require additional tuning parameters that have to be provided externally.



(a) Point samples of a line model (white dots) including two outliers (black dots)

(b) Least-squares estimate of a line using all data

Figure 2.5: Influence of outliers on least-squares estimates

M-Estimator	Error Term
Beaton and Tukey	$\rho_i(d_i) = \begin{cases} \frac{a^2}{6}\left[1-\left(1-\left(\frac{d_i}{a}\right)^2\right)^3\right], & \text{if } \lvert d_i\rvert \le a \\ \frac{a^2}{6}, & \text{if } \lvert d_i\rvert > a \end{cases}$
Cauchy	$\rho_i(d_i) = \frac{b^2}{2}\log\left[1+\left(\frac{d_i}{b}\right)^2\right]$
Huber	$\rho_i(d_i) = \begin{cases} \frac{1}{2}d_i^2, & \text{if } \lvert d_i\rvert \le c \\ \frac{1}{2}c(2\lvert d_i\rvert - c), & \text{if } \lvert d_i\rvert > c \end{cases}$

Table 2.3: Different robust cost functions [Ste99]. Parameters a, b and c can be used for tuning; d_i is an individual distance term

Random Sampling

A second family of techniques for robust weak calibration given a measurement containing N correspondences is based on random sampling. The steps of a random sampling technique are shown in algorithm 1 on page 27. Its key idea is to iteratively find a best model built with a minimum number of data points which can subsequently be used for two purposes:

- Serving as initial value for subsequent refinement via M-estimation.
- Separating data into inliers and outliers based on residual errors with respect to the best model found so far. All inliers can then be used for *non*-robust estimation techniques of section 2.6.2.

Exhaustive random sampling requires all $n = \binom{N}{k} = \frac{N!}{k!(N-k)!}$ possible sets of k samples given N correspondences to be tested and is not suitable for real applications. Hence, n may be fixed externally to meet runtime requirements.

Algorithm 1 Random sampling

Repeat n times ($j = 0 \dots n-1$):

1. Pick a random sample S_j containing a minimum number k of correspondences
2. Compute an intermediate estimate of the model using S_j only
3. Compute error ρ_j according to equation (2.38) with respect to all N correcpondences
4. Update best model estimate if ρ_j is lower than ρ_k, $k = 0 \dots j-1$

Current techniques improve the framework of random sampling by introducing *guided matching* so that samples S contain data points that are likely to be inliers instead of picking them randomly [TM02].

There are two major random sampling approaches: Least median of squares (LMedS) [RL03] and random sampling consensus (RANSAC) [FB81].

Least Median of Squares The least median of squares (LMedS) technique chooses the median squared error

$$\rho_{\text{LMedS}} = \underset{i}{\text{med}}\, \rho_i^2 \tag{2.41}$$

as cost function. Hence, it imposes an upper bound of 50% on the ratio of outliers. Variations like the k-th order statistics (LkOS) do not use the median but the k-th largest error term [Mee04].

Random Sampling Consensus The RANSAC algorithm requires an additional threshold t for computing an error term ρ in a random sampling step like step 3 in algorithm 1 and distinguishes directly between inliers and outliers. The number of inliers is used as a score in RANSAC, so that a cost function (2.38) can be given as

$$\rho_{\text{RANSAC}} = \sum_{i=1}^{N} \rho_i \quad , \quad \rho_i = \begin{cases} 0 & \text{if } |d_i| < t \\ 1 & \text{otherwise} \end{cases} \tag{2.42}$$

Recent variations of RANSAC combine the random sampling algorithm with M-estimators (MSAC) or use different, statistically motivated scores as e. g. the maximum likelihood criterion (MLESAC) [TZ00]. The number of iterations n in RANSAC may be estimated given a confidence level, the outlier ratio and a noise model of inliers [HZ03].

Figure 2.6 summarizes the workflow of robust weak calibration for known internal camera parameters.

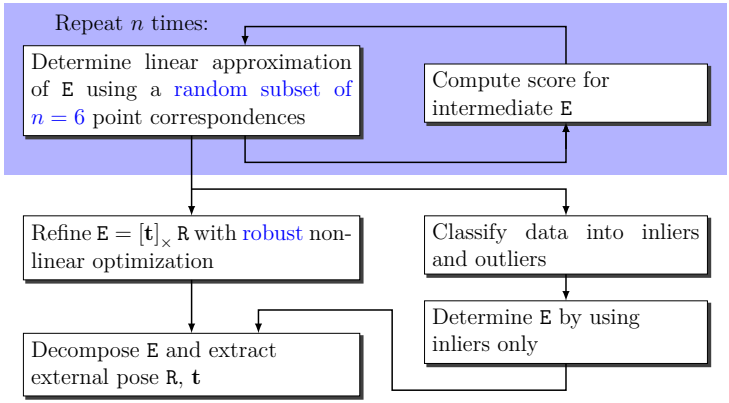


Figure 2.6: Workflow of robust weak calibration by random sampling

Hough Transform

A third robust estimation scheme frequently used in 2D computer vision is the Hough transform [Hou62]. It can be described as a voting technique operating with a *parameter space* that is discretized into *accumulator cells* and is called *Hough map*. It has two main flavors:

- **One-to-many transform**: In this approach, each single measurement votes for all accumulator cells in parameter space it may belong to. Afterwards, the accumulator cell having the highest number of votes is identified as parameter model.
- **Many-to-one transform**: Also being called *randomized Hough transform* [KKA99], a many-to-one transform can be closely related to random sampling. By picking a minimum number of necessary measurements for computing a model, its corresponding accumulator cell in parameter space is incremented. In contrast to RANSAC or least median of squares, intermediate estimates are not evaluated against all data. The parameter model is identified as the accumulator cell having the highest number of votes after n random trials. Therefore, a many-to-one Hough transform is suitable only if the probability of picking an outlier-free data set is high enough as both outlier-free models and models build from data including outliers are equally scored.

Obviously, the tuning parameter of a Hough transform is the size of an accumulator cell in parameter space, i. e. the resolution of a Hough map. Usually, a trade-off is necessary: If the resultion is increased, each cell represents a smaller interval in parameter space, hence decreases the discretization error. On the other hand, by increasing the resolution of the Hough map, more cells have to be visited for each measurement so that processing speed

decreases. Finally, Hough transforms are less sensitive to noise using a bigger cell size, as the accumulator cell representing the true model in parameter space may still cover the range of many noisy measurements.

A severe computational limitation of Hough transforms is the fact that the number of accumulator cells is exponential in the degrees of freedom to be estimated. As an example, a five-dimensional parameter space is at least needed for the robust estimation of an essential matrix. This makes the Hough transform unfeasible for relative pose estimation. In chapter 5, the Hough transform could be used for estimating epipoles given dense correspondences.

2.6.4 Conclusion

This section has reviewed robust estimation techniques for weak calibration. It has been shown that non-linear algorithms have to be used in order to optimize calibration results with respect to erroneous locations of image features. If measurements contain outliers, robust estimation algorithms have to be applied.

This work introduces geometry-driven techniques for relative pose estimation. Their objective is to improve the accuracy of initial values for a final optimization step.

Chapter 3

Parametrizations for Relative Pose Estimation

3.1 Epipoles and Relative Pose Estimation

3.1.1 The Degrees of Freedom in Epipolar Geometry

Section 2.4.1 mentioned that the 5 degrees of freedom of an essential matrix $\mathtt{E}$ can be identified as the relative orientation $\mathtt{R}'$ of two cameras and the direction of the translation vector $\mathbf{t}'$ between their optical centers $\mathbf{C}$ and $\mathbf{C}'$. However, the seven degrees of freedom of a fundamental matrix $\mathtt{F}$ have been motivated so far by counting the number of elements as well as the constraints on $\mathtt{F}$.

It is possible to find a common interpretation of the degrees of freedom for $\mathtt{E}$ as well as $\mathtt{F}$ as they contribute to two different aspects: The *epipoles* and the *epipolar transformation*[HG93, LF94, LF98]. Whereas the two epipoles are elements of $\mathbb{P}^2$ and may be defined by two parameters each (see section 3.2), the epipolar transformation relates epipolar lines in a first image to corresponding ones in the second by $\mathbf{l}' = \mathtt{L}^{-\top}\mathbf{l}$. Here, $\mathtt{L}$ is called *epipolar line homography*. Figure 3.1 shows an illustration. In the literature, an explicit notation for the relative pose of two cameras has not been mentioned so far.

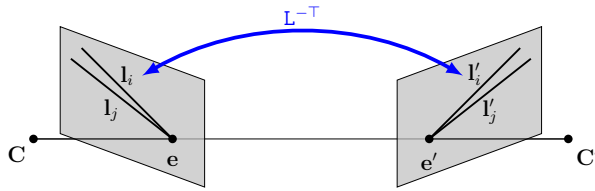


Figure 3.1: Epipoles $\mathbf{e}, \mathbf{e}'$ and the epipolar line homography $\mathtt{L}$

3.1.2 Stability Analysis

The stability of epipoles and the epipolar line homography using noisy measurements has been analyzed by Luong and Faugeras [LF94] even though an explicit notation has not been used. Gaussian noise $\mathcal{N}_2(\mathbf{0}, \sigma\mathtt{I})$,

$$\mathcal{N}_n(\boldsymbol{\mu}, \mathtt{C}) : \mathrm{prob}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \det(\mathtt{C})}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \mathtt{C}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{3.1}$$

has been added to pixel positions and the relative error of epipoles and the epipolar line homography has been computed. At a standard deviation of $\sigma = 1$ pixel, parameters of the epipolar transformation yield a relative error of approximately 0.1% whereas the relative error of the two epipoles was five times as high (0.5%). Similar observations have been made for other standard deviations. It has been concluded that the stability of a fundamental or essential matrix can primarily be characterized by the stability of epipoles.

In addition to the observations described in the last paragraph, the stability of epipoles has been further analyzed in this work. In an experiment, normalized epipoles have been chosen in spherical coordinates with co-latitudes $\vartheta \in [0°, 90°]$ (see section 3.2). According to this setup, synthetic random point correspondences containing additive zero-mean, uncorrelated Gaussian noise $\mathcal{N}_2(\mathbf{0}, \sigma\mathtt{I})$ have been computed. Then, the relative pose of two cameras has been reestimated. The mean error $\mu_\varphi, \mu_\vartheta$ and standard deviation $\sigma_\varphi, \sigma_\vartheta$ of epipole estimates in a random sampling pursuit of $n = 1000$ trials has been measured. Figure 3.2(a) shows that starting at a co-latitude $\vartheta = 20°$, the longitude φ of an epipole is more robust against noise than ϑ. It can be seen in figure 3.2(b) that the residual angular error Δe

$$\Delta e(\mathbf{e}, \mathbf{e}^*) = \arccos\left(\frac{\mathbf{e}^\top \mathbf{e}^*}{|\mathbf{e}||\mathbf{e}^*|}\right) \tag{3.2}$$

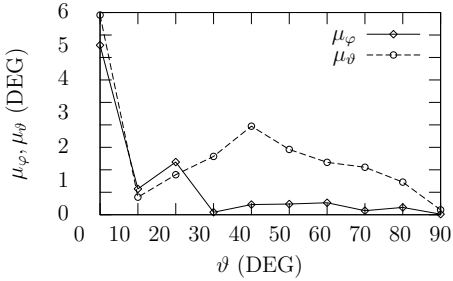
decreases with ϑ as well.

3.1.3 Relative Pose Estimation by Decomposition of Epipolar Geometry

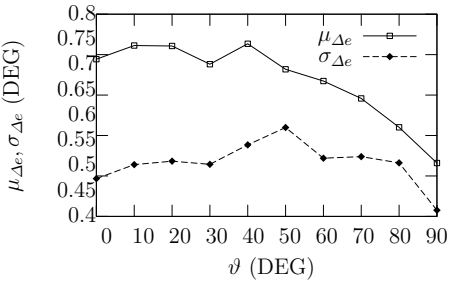
The last sections have demonstrated that epipoles deserve special attention in the process of weak calibration. Section 3.1.1 has demonstrated that they are an individual component of epipolar geometry. Section 3.1.2 has laid out stability aspects up to an epipole's components.

There are two possible approaches for focusing on epipoles when determining the relative pose of two cameras:

- Estimate epipoles by standard procedures for relative pose estimation, but separately optimize epipoles and the epipolar line homography.
- Determine the position of epipoles with techniques that are not necessarily based on epipolar geometry. Afterwards, estimate the residual epipolar line homography.



(a) Angular error at 1.2 pixels noise



(b) Mean error and standard deviation of spherical coordinates for an epipole at a noise level of 1.2 pixels

Figure 3.2: Robustness of an epipole's parameters with respect to noise

Both cases treat the estimation of epipoles as a *primary* or *outer* estimation problem. The derivation of a correct epipolar line homography is considered as a *secondary*, *inner* estimation step. The objective of both approaches mentioned above is to find accurate estimates of epipoles $\mathbf{e}$, $\mathbf{e}'$ in the primary estimation step such that a "good" residual epipolar line homography $\mathtt{L}$ can be found. This reasoning conforms to the situation shown in figure 3.1: Estimating $\mathtt{L}$ without any knowledge of the positions of $\mathbf{e}$ and $\mathbf{e}'$ cannot be done.

Figure 3.3 shows an illustration of how the described way of decomposing epipolar geometry may be applied to the relative pose problem by exposing differences with respect to the standard robust calibration procedure shown in fig. 2.6.

In the following chapters, two methods are analyzed with respect to the estimation of two cameras' relative pose. A method based on a sparse set of point correspondences using separate optimization techniques (chapter 4) and a method for the direct determination of epipoles using a dense motion field (see chapter 5) have been chosen.

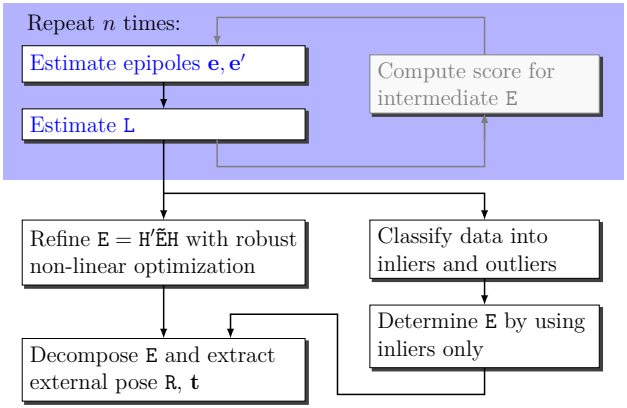


Figure 3.3: Geometry-driven weak calibration: Independent estimation of $\mathbf{e}$, $\mathbf{e}'$ and $\mathtt{L}$

However, in order to evaluate the two possibilities, the two epipoles and the epipolar transformation need to be identified in a fundamental or essential matrix in the following way:

$$\mathtt{F} = \mathtt{A}'(\mathbf{e}') \cdot \mathtt{L} \cdot \mathtt{A}(\mathbf{e}) \tag{3.3}$$

Such a factorization for weak calibration is necessary for considering the determination of epipoles and the epipolar line homography as an outer and inner estimation problem. In equation (3.3), the two matrices $\mathtt{A}'(\mathbf{e}')$ and $\mathtt{A}(\mathbf{e})$ exclusively contain information about the two epipoles and may be estimated in the outer estimation step, whereas $\mathtt{L}$ denotes a residual epipolar transformation to be estimated in the inner estimation step that can be done as soon as estimates of the two epipoles, and therefore $\mathtt{A}'(\mathbf{e}')$ and $\mathtt{A}(\mathbf{e})$, are available.

Section 3.3 discusses known parametrizations and suggests a new representation of the fundamental and the essential matrix. As a prerequisite, particularly for the Hough transform described in section 5.3, the next section focuses on a 2D representation of epipoles.

The rest of this chapter focuses on representations of entities in the scope of epipolar geometry. Besides *representations*, the term *parametrization* will be used likewise. The objective of the next subsections is to find a *factorization* (3.3) of a fundamental matrix $\mathtt{F}$ or essential matrix $\mathtt{E}$, i. e. parametrizations of epipoles and the epipolar transformation that may be used as decomposing factors of $\mathtt{E}$ and $\mathtt{F}$. In this case, the task of relative pose estimation can be split into an inner and outer optimization step.

3.2 Representations of Epipoles

3.2.1 Motivation

It has been mentioned in chapter 2 that the projective space $\mathbb{P}^2$ is different from the Euclidean space $\mathbb{R}^2$ as it is able to represent points at infinity. Consequently, Euclidean coordinates fail at representing epipoles if the camera translates sideways. However, homogeneous vectors are not a minimal representation as they need $n+1$ elements for describing n degrees of freedom, as they contain their magnitude as redundant information.

Non-minimal parametrizations may introduce *gauge* during numerical optimization. This term denotes the fact that changes in the set of parameters may have no effect on the value of the cost function. It has been mentioned that gauge may introduce ambiguous optima and lead to slower convergence [HZ03, Tri98]. The Hough transform is another technique that requires minimal parametrizations due to its computational complexity (see section 2.6.3).

The objective of this section is to find a suitable, minimal representation of an epipole that will be used in subsequent chapters, in particular in chapter 5.

When removing redundancy by fixing the scale of a homogeneous point in $\mathbb{P}^2$ such that $\|\mathbf{x}\| = 1$, the representation problem is identical to parameterizing unit vectors in 3D or, in other terms, the surface of the unit sphere S^2. Figure 3.4 shows an illustration of S^2.

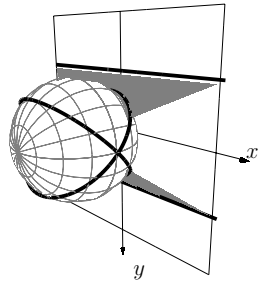


Figure 3.4: Unit sphere S^2 for representing the projective plane $\mathbb{P}^2$. Here, S^2 is oriented such that the north pole $\mathbf{x}_E = (0, 0, 1)^\top$ coincides with the optical axis. In this illustration, two parallel lines on the image plane at $z = 1$ can be found as a great circle on S^2. Their intersection does not have a Euclidean representation, but it can be identified on the equator of S^2.

Numerous parametrizations of points on S^2 exist with spherical coordinates (ϑ, φ) being the most obvious one. Here, $\vartheta \in [0°, 180°]$ represents the *co-latitude*, i. e. the angular deviance from the polar axis, whereas $\varphi \in [0°, 360°)$ is called *longitude*. It is well-known that parametrizations may either not be complete or contain singularities which makes

them unsuitable for specific situations. As an example, spherical coordinates (ϑ, φ) are singular at the poles as φ is not unique for $\vartheta = \{0°, 180°\}$. More details will be given below.

This section is organized as follows: After laying out some mathematical basics, an overview and comparison of existing representations of S^2 is presented. Subsequent experiments show that parametrizations originating from the field of cartography are suitable for representing epipoles.

3.2.2 Global and Local Parametrizations

It will first be shown that it is impossible to have a global one-to-one parametrization of S^2. The following details are closely based on [Stu64], in which a similar explanation for parameterizing the special orthogonal group $SO(3)$ (see appendix C) can be found.

The unit sphere S^2 topologically is a 2-dimensional compact manifold. A global 1-1 parametrization requires a homomorphism h from S^2 to the Euclidean space $\mathbb{R}^2$. A property of homomorphisms is that $h(\mathcal{U}_\mathbf{i})$ for an open neighborhood $\mathcal{U}_\mathbf{i}$ of a point $\mathbf{i}$ is open in $\mathbb{R}^2$. Hence, $h(\mathcal{I})$, being the union of all $h(\mathcal{U}_\mathbf{i})$ for $\mathbf{i} \in S^2$, would still be open. On the other hand, $h(\mathcal{I})$ describes a continuous map of a compact space, thus is still compact. As no Euclidean space contains an open compact subset, such a homomorphism cannot be found. As a consequence, there is no parametrization that preserves areas and lengths on S^2 simultaneously.

Parametrizations of S^2 fail at *singular points* that have infinitely many representations. However, a set of parameter patches, also called *atlas*, circumvents this limitation. An atlas may contain infinitely many parameter patches. In this case, for a point of interest $\mathbf{i}$, a unique parameter patch $h_\mathbf{i}$ can be chosen. This technique is also referred to as a *local parametrization* [HZ03]. A particular $h_\mathbf{i}$ usually is constructed such that $h_\mathbf{i}(\mathcal{U}_\mathbf{i})$, does not contain singular points and has advantageous numerical characteristics. It is interesting to note that using local parametrizations still may suffer from singularity-like situations. As an example, for parameterizing rotations, Hartley suggests to apply Householder transformations mapping a point of interest $\mathbf{i}$ to the origin $\mathbf{0}$ and choose a parametrization that "behaves well" in its vicinity [HZ03]. However, as it will be shown in section 3.3.2, a Householder transformation does not exist if $\mathbf{i}$ is identical to the origin and is numerically unstable if $\mathbf{i}$ is close to it.

Some robust estimation techniques like the Hough transform (see sections 2.6.3 and 5) explicitly require global parametrizations as a region of interest is not known in advance. Therefore, local parametrizations cannot be used in every situation.

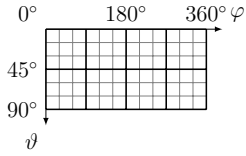
A second possibility to circumvent singularities is to use an atlas with a few parameter patches only [Fau93]. Such an atlas is suitable for the projective setting as representing the complete unit sphere S^2 is not necessary. As a point $\mathbf{x}$ and its antipodal point $-\mathbf{x} \sim \mathbf{x}$ on S^2 are equivalent, it is sufficient to parameterize a *hemisphere*. Parameter patches that cover a complete hemisphere exist and are given below. The following details focus on the *northern hemisphere* of S^2, i. e. to $\vartheta \in [0°, 90°], \varphi \in [0°, 360°)$, in which the north pole

$\mathbf{n} = \begin{pmatrix}0, 0, 1\end{pmatrix}^\top$ coincides with the optical axis, equivalently to figure 3.4.

3.2.3 Representations of Points on S^2

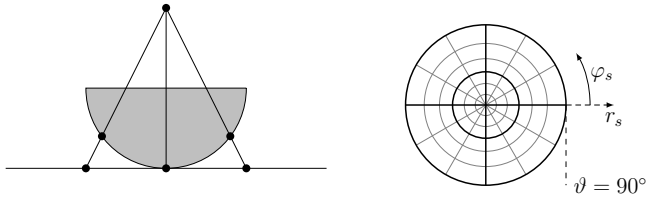
Many global parametrizations of S^2 are mentioned in the literature [Sny87]. Some of them can be found in the field of cartography. This section gives an overview and shows characteristics (see also table 3.1 and 3.2). Besides spherical coordinates, different *azimuthal projections* of a hemisphere are described. Other types are not considered as they have disadvantegous properties such as singular points, computational complexity or the lack of symmetry.

- Spherical coordinates (SPHERICAL):



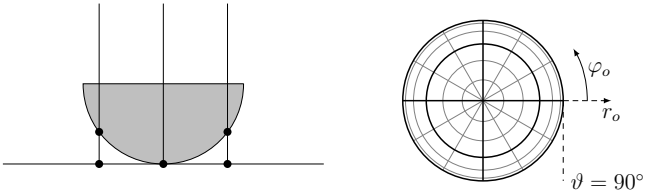
  Even though spherical coordinates yield a singular point at each of the two poles of S^2, they are used in many computer vision algorithms [Mee04]. Their advantage is their simple geometric interpretation.

- Stereographic projection (STEREO):



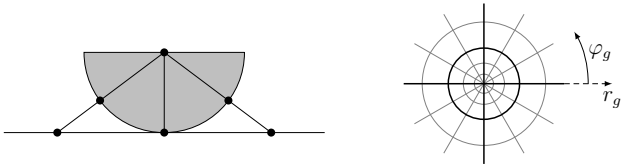
  A standard approach in topology for finding a parametrization of S^2 is stereographic projection. Here, the center of projection is located at the south pole $\mathbf{s}$, the Euclidean space $\mathbb{R}^2$ as the projection target is parallel to the equator. Corresponding points on $\mathbb{R}^2$ and S^2 can be found on the same ray through $\mathbf{s}$. It is obvious that $\mathbf{s}$ itself cannot be mapped onto $\mathbb{R}^2$. For computational purposes, a stereographic projection has the advantage of being a *rational* parametrization of points in $\mathbb{R}^3$. Hence, it does not involve trigonometric functions [Fau93].

- Orthographic projection (ORTHO):



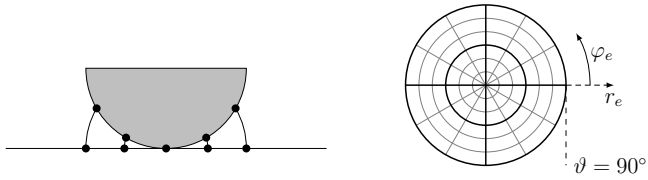
By omitting the z-coordinate of a point $\mathbf{X}_E$ in $\mathbb{R}^3$, its orthographic projection can be obtained. This parametrization does not contain singularities, but an atlas is needed for representing the northern and southern hemisphere of S^2 uniquely. Its advantage is its computational simplicity. It can be seen, however, that the resolution in ϑ decreases near the equator. This drawback is important in section 3.2.5 as it increases discretization errors in a Hough map.

- Gnomonic projection (GNOMONIC):



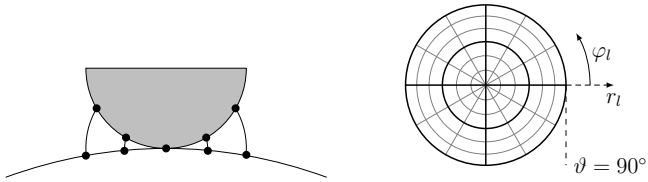
A gnomonic projection in cartography corresponds to the computation of a Euclidean representation given a homogeneous vector $\mathbf{x} \in \mathbb{P}^2$. Even though it is rational for points in $\mathbb{R}^3$, it is not unique for antipodal points and cannot represent the equator. A gnomonic projection inverts the process of mapping point coordinates from $\mathbb{R}^2$ to S^2 (see fig. 3.4). Therefore, it is suitable only for points of interest that are located in the camera's field of view. In this case, their homogeneous representation is not required anyway.

- Azimuthal equidistant projection (EQUI):



An azimuthal equidistant projection yields a 2D polar mapping of spherical coordinates. Euclidean distances on the map may be used as error terms for geodesics through the poles as their lengths are preserved.

- Azimuthal Lambertian projection (LAMBERT):



The last mapping considered in this paper is azimuthal Lambertian projection. It is area preserving, so regions on S^2 occupy the same area on the map. However, distances are not preserved.

Map	from S^2	to S^2	from SPHERICAL	to SPHERICAL
SPHERICAL	$\begin{pmatrix}\varphi\\ \vartheta\end{pmatrix} = \begin{pmatrix}\arctan\frac{\mathrm{Y}}{\mathrm{X}}\\ \arccos\mathrm{Z}\end{pmatrix}$	$\begin{pmatrix}\mathrm{X}\\ \mathrm{Y}\\ \mathrm{Z}\end{pmatrix} = \begin{pmatrix}\sin\vartheta\cos\varphi\\ \sin\vartheta\sin\varphi\\ \cos\vartheta\end{pmatrix}$	$\begin{pmatrix}\varphi\\ \vartheta\end{pmatrix} = \begin{pmatrix}\varphi\\ \vartheta\end{pmatrix}$	$\begin{pmatrix}\varphi\\ \vartheta\end{pmatrix} = \begin{pmatrix}\varphi\\ \vartheta\end{pmatrix}$
STEREO	$\begin{pmatrix}x_s\\ y_s\end{pmatrix} = \begin{pmatrix}\frac{\mathrm{X}}{1-\mathrm{Z}}\\ \frac{\mathrm{Y}}{1-\mathrm{Z}}\end{pmatrix}$	$\begin{pmatrix}\mathrm{X}\\ \mathrm{Y}\\ \mathrm{Z}\end{pmatrix} = \frac{1}{x_s^2+y_s^2+1}\begin{pmatrix}2x_s\\ 2y_s\\ x_s^2+y_s^2-1\end{pmatrix}$	$\begin{pmatrix}\varphi_s\\ r_s\end{pmatrix} = \begin{pmatrix}\varphi\\ \frac{\sin(\vartheta)}{1+\cos(\vartheta)}\end{pmatrix}$	$\begin{pmatrix}\varphi\\ \vartheta\end{pmatrix} = \begin{pmatrix}\varphi_s\\ \arccos(-\frac{r_s^2-1}{r_s^2+1})\end{pmatrix}$
ORTHO	$\begin{pmatrix}x_o\\ y_o\end{pmatrix} = \begin{pmatrix}\mathrm{X}\\ \mathrm{Y}\end{pmatrix}$	$\begin{pmatrix}\mathrm{X}\\ \mathrm{Y}\\ \mathrm{Z}\end{pmatrix} = \begin{pmatrix}x_o\\ y_o\\ \sqrt{1-x_o^2-y_o^2}\end{pmatrix}$	$\begin{pmatrix}\varphi_o\\ r_o\end{pmatrix} = \begin{pmatrix}\varphi\\ \sin(\vartheta)\end{pmatrix}$	$\begin{pmatrix}\varphi\\ \vartheta\end{pmatrix} = \begin{pmatrix}\varphi_o\\ \arcsin(r_o)\end{pmatrix}$
GNOMONIC	$\begin{pmatrix}x_g\\ y_g\end{pmatrix} = \begin{pmatrix}\frac{\mathrm{X}}{\mathrm{Z}}\\ \frac{\mathrm{Y}}{\mathrm{Z}}\end{pmatrix}$	$\begin{pmatrix}\mathrm{X}\\ \mathrm{Y}\\ \mathrm{Z}\end{pmatrix} = (x_g^2+y_g^2+1)^{-1/2}\begin{pmatrix}x_g\\ y_g\\ 1\end{pmatrix}$	$\begin{pmatrix}\varphi_g\\ r_g\end{pmatrix} = \begin{pmatrix}\varphi\\ \tan(\vartheta)\end{pmatrix}$	$\begin{pmatrix}\varphi\\ \vartheta\end{pmatrix} = \begin{pmatrix}\varphi_g\\ \arctan(r_g)\end{pmatrix}$
EQUI	$\begin{pmatrix}x_e\\ y_e\end{pmatrix} = \begin{pmatrix}\mathrm{X}\frac{\arcsin\sqrt{\mathrm{X}^2+\mathrm{Y}^2}}{\sqrt{\mathrm{X}^2+\mathrm{Y}^2}}\\ \mathrm{Y}\frac{\arcsin\sqrt{\mathrm{X}^2+\mathrm{Y}^2}}{\sqrt{\mathrm{X}^2+\mathrm{Y}^2}}\end{pmatrix}$	$\begin{pmatrix}\mathrm{X}\\ \mathrm{Y}\\ \mathrm{Z}\end{pmatrix} = \begin{pmatrix}x_e\,\mathrm{sinc}\sqrt{x_e^2+y_e^2}\\ y_e\,\mathrm{sinc}\sqrt{x_e^2+y_e^2}\\ \cos\sqrt{x_e^2+y_e^2}\end{pmatrix}$	$\begin{pmatrix}\varphi_e\\ r_e\end{pmatrix} = \begin{pmatrix}\varphi\\ \vartheta\end{pmatrix}$	$\begin{pmatrix}\varphi\\ \vartheta\end{pmatrix} = \begin{pmatrix}\varphi_e\\ r_e\end{pmatrix}$
LAMBERT	$\begin{pmatrix}x_l\\ y_l\end{pmatrix} = \begin{pmatrix}\frac{\mathrm{X}}{\sqrt{\mathrm{X}^2+\mathrm{Y}^2}}\sin\frac{\arccos\mathrm{Z}}{2}\\ \frac{\mathrm{Y}}{\sqrt{\mathrm{X}^2+\mathrm{Y}^2}}\sin\frac{\arccos\mathrm{Z}}{2}\end{pmatrix}$	$\begin{pmatrix}\mathrm{X}\\ \mathrm{Y}\\ \mathrm{Z}\end{pmatrix} = \begin{pmatrix}2x_l\sqrt{1-x_l^2-y_l^2}\\ 2y_l\sqrt{1-x_l^2-y_l^2}\\ 1-2x_l^2-2y_l^2\end{pmatrix}$	$\begin{pmatrix}\varphi_l\\ r_l\end{pmatrix} = \begin{pmatrix}\varphi\\ \frac{\sin(\vartheta/2)}{\sin(\pi/4)}\end{pmatrix}$	$\begin{pmatrix}\varphi\\ \vartheta\end{pmatrix} = \begin{pmatrix}\varphi_l\\ 2\arcsin(r_l\sin(\frac{\pi}{4}))\end{pmatrix}$

Table 3.1: Overview of different mappings for global parametrizations of points $(\mathrm{X},\mathrm{Y},\mathrm{Z})^\top \in S^2$. Transformation from and to spherical coordinates are given in polar coordinates $(x_\xi, y_\xi)^\top = (r_\xi\cos\varphi_\xi, r_\xi\sin\varphi_\xi)^\top$. For simplicity, mappings do not include scaling to and from a desired range. In this table, angles are given in radians.

Map	Singular points	Non-unique points	Non-representable points
SPHERICAL	$\{\mathbf{X}_E : \|\mathrm{z}\| = 1\}$	-	-
STEREO	$\{\mathbf{X}_E : \mathrm{z} = -1\}$	-	$\{\mathbf{X}_E : \mathrm{z} = -1\}$
ORTHO	-	$\{\mathbf{X}_E : \mathrm{z} \neq 0\}$	-
GNOMONIC	-	$\{\mathbf{X}_E : \mathrm{z} \neq 0\}$	$\{\mathbf{X}_E : \mathrm{z} = 0\}$
EQUI	$\{\mathbf{X}_E : \mathrm{z} = -1\}$	-	-
LAMBERT	$\{\mathbf{X}_E : \mathrm{z} = -1\}$	-	-

Table 3.2: Shortcomings of global parametrizations of the complete unit sphere S^2 for points $\mathbf{X}_E \in \mathbb{R}^3$.

3.2.4 Implementation

Section 3.1.3 has already introduced the two approaches derived in this work in which a minimal parametrization of epipoles will be used. Whereas in the first the relative pose of two cameras is estimated by a random sampling technique (chapter 4), the second is targeted at the direct detection of epipoles using a Hough transform (chapter 5). In the latter approach, epipoles are detected by intersecting epipolar lines on a Hough map.

For this application, as lines in an original image are mapped onto curves on the Hough map, a polar-recursive algorithm for azimuthal maps of S^2 has been used for accessing corresponding accumulator cells. It is summarized in algorithm 2.

Algorithm 2 Polar-recursive algorithm in pseudo-code for accessing a Hough map

```
// Recursive plot of a line in interval phi_start - phi_end
function recursiveCurve(line,phi_start,phi_end):

  newpoint = pointOnLine (line,phi_end)

  if(dist(newPoint,drawnPoint)>1):
    recursiveCurve(line,phi_start,(phi_start+phi_end)/2)
    recursiveCurve(line,(phi_start+phi_end)/2,phi_end)
  else:
    plotPoint(newPoint)
    drawnPoint=newPoint


// Draw curve
phi_start = arctan(line.x,line.y)
startPoint = polarPoint(pi/2.0,phi_start)
plotPoint(startPoint)
drawnPoint = startPoint
recursiveCurve(line,phi_start,phi_start+pi)
```

Here, `pointOnLine(line,phi)` determines the point on a projective line $\mathbf{l} \in \mathbb{P}^2$ given a longitude φ. Using spherical coordinates, this can be done with

$$\vartheta = \arctan \frac{l_z}{l_x \cos(\varphi) + l_y \sin(\varphi)} \tag{3.4}$$

Figure 3.5 illustrates the polar-recursive algorithm for navigating on an EQUI Hough map. For illustration purposes, a low resulution has been chosen; in this example, the complete map of a hemisphere of S^2 is discretized such that it fits into a grid of 8×8 accumulator cells. The polar-recursive algorithm 2 computes the discretization of a great circle on the Hough map. It starts at a point on the equator and first considers the interval

$\varphi = [0, 180°]$ up to its antipodal point on the opposite side of the equator. The interval is recursively halved until its end falls onto an adjacent accumulator cell which then can be accessed and incremented. The remaining interval is subsequently processed.

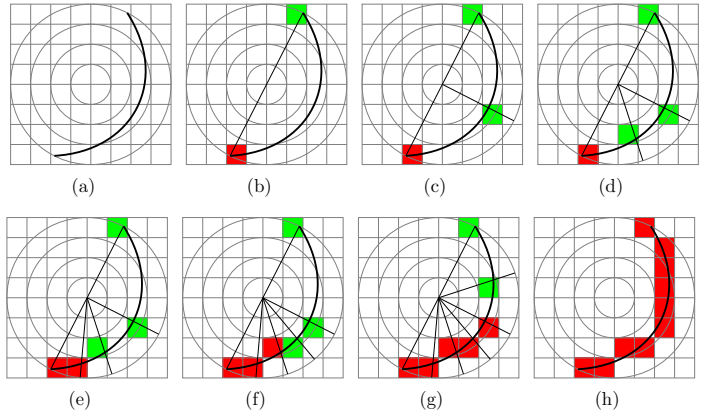


Figure 3.5: Steps for polar-recursive navigation on a Hough map given parameters of an epipolar line (a). Step (b): Determine longitudes at the equator. A red mark indicates a visited accumulator cell, a green mark indicates an intended visit. Steps (c) – (e): Split interval recursively until it ranges to an adjacent cell. (f) – (g): Continue likewise with intended visits. (h): Discretized epipolar line.

An implementation of algorithm 2 fails if curves pass through the north pole, so it has to be made sure that $\vartheta_{\min} = \arctan(l_z/\sqrt{l_x^2 + l_y^2}) > \varepsilon$. Otherwise, an equivalent radial-recursive algorithm has to be used instead.

3.2.5 Experimental results

The four parametrizations STEREO, ORTHO, EQUI and LAMBERT are considered below. The two remaining ones described in section 3.2.3 have already been classified not to be suitable for Hough maps: SPHERICAL yields singular points at the poles whereas GNOMONIC cannot represent a complete hemisphere.

Robust estimation of points via Hough transforms

It has been mentioned in section 2.6.3 that the size of a Hough cell affects the accuracy of estimates due to discretization of the parameter space. Based on the expressions listed in table 3.1, the resolution of a Hough map can be related to an upper bound of the angular

discretization error Δe in the following way. The map is assumed to be large enough that the longitudinal difference $|\varphi_1 - \varphi_2|$ of two adjacent points $\mathbf{x}_1$ and $\mathbf{x}_2$ does not exceed 90°.

$$\begin{aligned}
& \Delta e && = \arccos(\mathbf{x}_1^\top \mathbf{x}_2) \quad \text{with} \quad \|\mathbf{x}_1\| = \|\mathbf{x}_2\| = 1, \mathbf{x} \in \mathbb{P}^2 \\
\Leftrightarrow \quad & \cos(\Delta e) && = \sin\vartheta_1 \sin\vartheta_2 (\cos\varphi_1 \cos\varphi_2 + \sin\varphi_1 \sin\varphi_2) + \cos\vartheta_1 \cos\vartheta_2 \\
& && = \sin\vartheta_1 \sin\vartheta_2 \cos(\varphi_1 - \varphi_2) + \cos\vartheta_1 \cos\vartheta_2 \\
& && \leq \sin\vartheta_1 \sin\vartheta_2 + \cos\vartheta_1 \cos\vartheta_2 \quad (\text{as } \vartheta_1, \vartheta_2 \geq 0°,\ |\varphi_1 - \varphi_2| < 90°) \quad (3.5) \\
& && = \cos(\vartheta_1 - \vartheta_2) \\
\Leftrightarrow \quad & |\Delta e| && \leq |\vartheta_1 - \vartheta_2| = \Delta e_{\max}
\end{aligned}$$

Table 3.3 summarizes relations between the width W of a square Hough map in pixels and an upper bound $\Delta e_{\max}$ of the discretization error:

Map	Width W of map in pixels	Maximum angular error $\Delta e_{\max}$
STEREO	$W = \frac{2+\cos(\Delta e_{\max})}{\sin(\Delta e_{\max})}$	$\Delta e_{\max} = 90° - \arccos\left(-\frac{(1-1/W)^2-1}{(1-1/W)^2+1}\right)$
ORTHO	$W = \frac{1}{1-\cos(\Delta e_{\max})}$	$\Delta e_{\max} = \arccos(1 - 1/W)$
EQUI	$W = \frac{90°}{\Delta e_{\max}}$	$\Delta e_{\max} = 90°/W$
LAMBERT	$W = \frac{\sin(45°)}{\sin(45°)-\sin(45°-(\Delta e_{\max})/2)}$	$\Delta e_{\max} = 90° - 2\arcsin(\sin(45°)(1 - 1/W))$

Table 3.3: Relationship between the width W of a Hough map and the maximum angular error $\Delta e_{\max}$ caused by discretization of the parameter space

The relations described in table 3.3 are illustrated in figure 3.6. It can be seen that STEREO dominates other parametrizations, hence provides a lower maximum angular error than LAMBERT, EQUI and ORTHO, regardless of W. ORTHO provides the highest maximum angular error $\Delta e_{\max}$, whereas EQUI and LAMBERT both can be found close to STEREO. However, it will be shown below that a dominating parametrization does not account for noisy measurements.

A first experiment used synthetic data as input and is targeted at the estimation of points on a Hough map given a set of 64 intersecting lines. Due to symmetry, analysis has been reduced to 10 intersections on the x-axis, i. e. to $\varphi = 0°$, whereas co-latitudes have been chosen to be $\vartheta = 0°, 10°, \ldots, 90°$. For each of the 64 lines, corresponding pixels on the Hough map have been incremented using the polar-recursive algorithm described above. An example final Hough map for $\vartheta = 40°$ can be seen in figure 3.7. Indeed, all lines intersect on the x-axis at about $\vartheta = 40°$ on the Hough map. In the illustrated case, no noise has been added to the orientation of the lines, so that the remaining angular error Δe between the best accumulator cell of the Hough map and the true intersection can be considered as discretization error.

In order to evaluate robustness, Gaussian noise $\mathcal{N}(0, \sigma)$ has been added to all orientations $\varphi_\mathbf{l}$ of a line. Figure 3.8 shows results for different noise levels at $\vartheta = 70°$. It can be

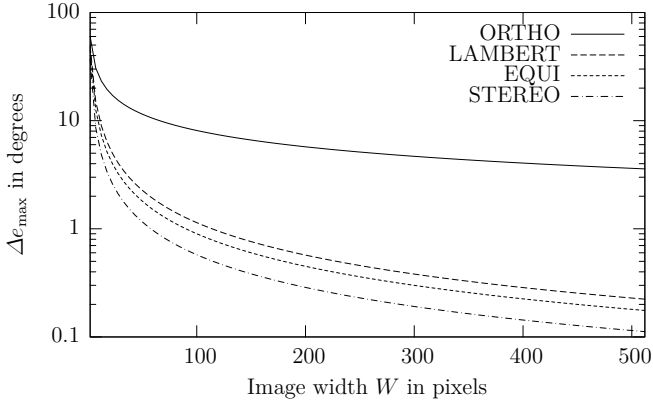


Figure 3.6: Maximum angular error caused by discretization vs. the width W of a Hough map

seen that for $\sigma \geq 4°$, intersections incorrectly tend to be attracted by the equator in case of ORTHO. This problem is illustrated in figure 3.9(a). As the resolution of ϑ decreases near the equator, discretization incorrectly yields a high number of votes due to the path of mapped lines. Other parametrizations, e. g. LAMBERT in figure 3.9(b), do not suffer from this phenomenon.

The accuracy of an estimated intersection of lines is also affected by the finite number of curves on the Hough map. Therefore, in another experiment, a Gaussian filter with kernel width g has additionally been applied to the Hough map. This approach is contrary to others in which special techniques like hierarchical [QM89] or irregular Hough maps are used for detecting vanishing points by intersecting lines passing through edges in an image [LMLK94]. A snapshot of a smoothed Hough map can be seen in figure 3.9(c). Evaluation results are shown in figures 3.10 and 3.11. It can be seen that the "attracting equator" of ORTHO could not be resolved by Gaussian smoothing. When using other parametrizations, maximum residual errors can approximately be halved at a moderate noise level (figure 3.10). Best results could be achieved with EQUI and LAMBERT.

Parametrization for non-linear optimization

Hornegger and Tomasi noted that for optimization, *fair* parametrizations are desirable as they enable sensitivity measures that are inherent to a problem itself and do no depend on its particular representation [HT99]. They showed that a fair parametrization $\mathbf{x}' = h(\mathbf{x})$ can be characterized by an orthogonal Jacobian $\mathtt{Q} = \partial h(\mathbf{x})/\partial \mathbf{x}^\top$. Thus, a fair parametrization is area- and length preserving and does not change the characteristics of an underlying error model. It has been mentioned in section 3.2.2 that a fair 2D representation of the unit

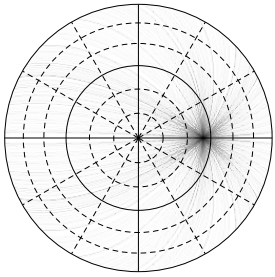


Figure 3.7: LAMBERT map for $\vartheta = 40°$

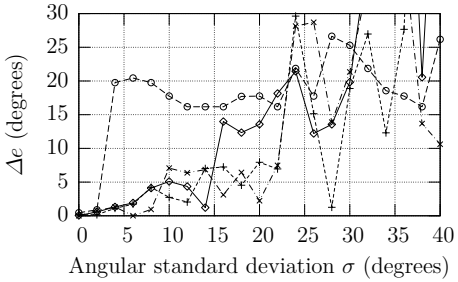
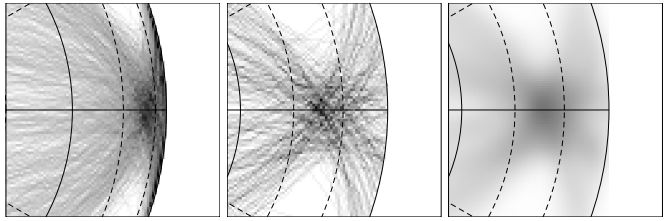


Figure 3.8: Angular error at different noise levels for an intersection at $\vartheta = 70°$. The line styles are identical to figure 3.6. ORTHO shows a dominating error of 20° which is caused by an attracting equator of the Hough map at $\vartheta = 90°$.

sphere S^2 does not exist. Therefore, different parametrizations also affect the performance of non-linear optimization techniques. Hence, the sensitivity with respect to noise has been evaluated as well. For this experiment, the same set of 64 lines has been used, and their intersection has been estimated by minimizing $\rho(\mathbf{x}) = \sum(\mathbf{x}^\top \mathbf{l})^2/(\mathbf{x}^\top \mathbf{x}\mathbf{l}^\top \mathbf{l})$. Figure 3.12 shows that the chosen representation affects the performance in the case of ORTHO. All other parametrizations need about the same number of iterations and do not depend on the co-latitude or noise level.



(a) ORTHO (b) LAMBERT (c) LAMBERT (after Gaussian smoothing)

Figure 3.9: Influence of noise on the stability of different parametrizations for $\vartheta = 70°$

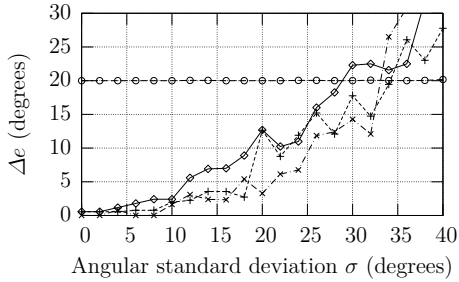


Figure 3.10: Effects of Gaussian smoothing ($g = 10$ pixels) for $\vartheta = 70°$. the case of ORTHO, smoothing intensifies the effect of an attracting equator so that a constant error of 20° can be observed even for noise-free input data.

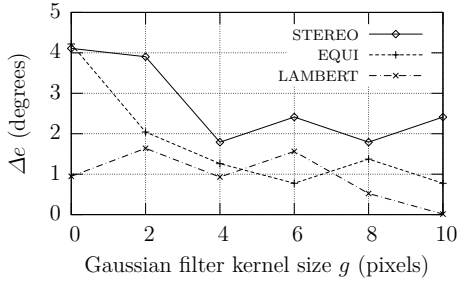
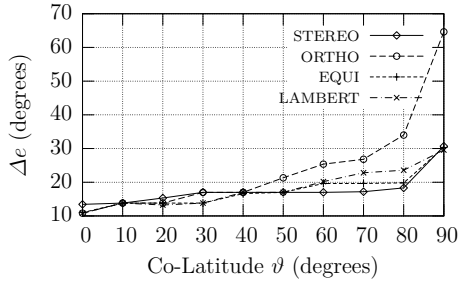
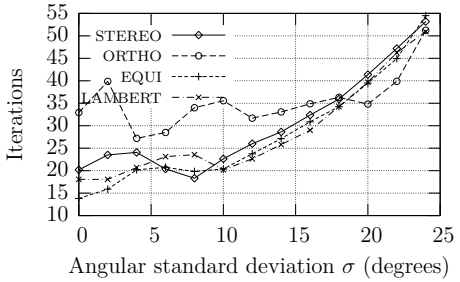


Figure 3.11: Angular errors for $\sigma = 8°$ and $\vartheta = 70°$



(a) Residual angular error for $\sigma = 8°$



(b) Number of iterations for $\vartheta = 80°$

Figure 3.12: Different parametrizations for non-linear optimization algorithms

3.3 Geometry-driven Factorization of Epipolar Geometry

Section 2.6.3 has introduced fundamentals of the robust estimation of two cameras' relative pose and emphasized the importance of epipoles in stereo geometry. The need for a factorization of the fundamental and the essential matrix revealing epipoles and the epipolar line homography has been motivated in section 3.1.3. Such a parametrization can also be beneficial if a minimal representation of $\mathtt{F}$ and $\mathtt{E}$ is needed, without an explicit representation of epipoles and the epipolar line homography (see section 2.6.2).

This section first reviews existing minimal and non-minimal representations of the fundamental and essential matrix. Subsequently, a new parametrization is introduced. It is based on Householder transformations and leads to a minimal, symmetric and geometrically interpretable factorization meeting the requirements mentioned above.

3.3.1 State of the Art

The structure of the essential and fundamental matrix has been intensively studied and parametrizations have been suggested that enforce the singularity of $\mathtt{F}$ as well as the cubic constraint (2.21) in case of the essential matrix. In this context, non-minimal and minimal parametrizations can be differentiated [Tri98].

Non-minimal parametrizations

If more parameters than degrees of freedom are used, it is still possible to retain characteristics of $\mathtt{E}$ and $\mathtt{F}$. In case of a fundamental matrix, such an over-parametrization can e. g. be written as

$$\mathtt{F} = [\mathbf{p_4}]_\times \mathtt{M}, \tag{3.6}$$

involving 12 elements of a projection matrix $\mathtt{P}$ as shown in equation (2.9) [HZ03]. It can be seen that in equation (3.6), $\mathtt{F}$ is singular, since $[\mathbf{p_4}]_\times$ is. If intrinsic camera parameters are known, computations can be done in normalized coordinates by using an essential matrix $\mathtt{E}$. In this case, $\mathtt{M}$ corresponds to the rotation matrix $\mathtt{R}$ (see equation 2.18), so it has additional properties that cannot be ensured by nine arbitrary elements (see appendix C). Hence, this parametrization fails for essential matrices.

In [IT00], Isgro and Verri introduced a parametrization of $\mathtt{F}$ involving eight parameters. It ensures the singularity constraint by finding a factorization

$$\mathtt{F} = \mathtt{H}'^{\top} \overline{\mathtt{F}} \mathtt{H} \quad \text{with} \tag{3.7}$$

$$\overline{\mathtt{F}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{3.8}$$

Here, $\mathtt{H}'$ is a rank-3 matrix that maps an epipole to the X-axis. $\mathtt{H}$ is then found such that expression (3.7) is valid.

It can be seen that $\overline{\mathtt{F}}$ is rank-2, so $\mathtt{F}$ is indeed singular. However, this approach has some conceptual drawbacks:

- Isgro and Verri have motivated their method by the need for parameterizing rank-2 matrices. However, simpler methods exist for this purpose (see above).
- The approach does not focus on the cubic constraint (2.21) of an essential matrix.
- The decomposition (3.7), especially the construction of $\mathtt{H}$, cannot completely be interpreted geometrically.
- Critical configurations in which the factorization fails are mentioned, but neglected.

Other non-minimal parametrizations are able to represent singular matrices by involving the locations of epipoles, as analyzed by Luong [LDFP93].

If the left epipole is given as $\mathbf{e} = \left(\alpha_e, \beta_e, -1\right)^\top$, a fundamental matrix may be written as

$$\mathtt{F} = \begin{bmatrix} a & b & \alpha_e a + \beta_e b \\ c & d & \alpha_e c + \beta_e d \\ e & f & \alpha_e e + \beta_e f \end{bmatrix} \tag{3.9}$$

The right epipole $\mathbf{e}' = \left(\alpha'_e, \beta'_e, -1\right)^\top$ may additionally be used for deriving a similar, symmetric representation

$$\mathtt{F} = \begin{bmatrix} a & b & \alpha_e a + \beta_e b \\ c & d & \alpha_e c + \beta_e d \\ \alpha'_e a + \beta'_e c & \alpha'_e b + \beta'_e d & \lambda \end{bmatrix} \tag{3.10}$$

with

$$\lambda = \alpha'_e \alpha_e a + \alpha'_e \beta_e b + \beta'_e \alpha_e c + \beta'_e \beta_e d. \tag{3.11}$$

This notation has been used in research e. g. in [PG98, Zha97, Zha98], because it ensures a singular fundamental matrix as the third row is a linear combination of the first and second — the same with columns. It has been mentioned in [LF94] that the parameters a, b, c, d implicitely represent the epipolar line homography $\mathtt{L}$. However, equation (3.10) and the elements of $\mathtt{L}$ cannot be used if one or both epipoles are located at infinity, e. g. if a camera translates sideways [WG06b]. Zhang takes care of this problem by using an atlas of 36 different parametrizations [Zha97]. In a later publication, he projectively transforms homogeneous coordinates in a preprocessing step such that mapped epipoles are finite [ZL01]. This way, parametrization (3.10) can be applied in any case. As a projective transformation does not retain metric properties (see section 2.1.3), error terms in transformed space like the ones listed in table 2.2 are not compliant with their intended

geometrical or statistical meaning. As a work-around, Zhang suggested to modify the cost function (2.38) such that an original error model is preserved in transformed space.

It can be concluded that non-minimal parametrizations for fundamental matrices exist in the literature. None of the introduced methods provides a factorization which can be used to treat the estimation of epipoles and the epipolar line homography as two different optimization problems. Moreover, properties of an essential matrix are not considered.

Minimal parametrizations

In [BS04], Bartoli et al. use the singular value decomposition (SVD) of a fundamental matrix for a minimal parametrization. In this approach, the orthonormal matrices $\mathtt{U}$ and $\mathtt{V}^\top$ are chosen such that they represent rotations (see section 3.3.6). As there is no global minimal parametrization of rotation matrices [Stu64], many representations suffer from a critical configuration known as the "gimbal lock" problem. It is named after an equivalent situation which can be found in gyroscopic devices [Gra98]. In Bartoli's approach, local parametrizations are used in order to circumvent this situation. Section 3.2.2 gives more details on global versus local parametrizations.

Even though in [BS04], Bartoli justified his approach with experimental results, it has a conceptual drawback as the proposed orthonormal representation is not unique for the special case of essential matrices: There is a continuum of possible singular value decompositions as for any Givens rotation $\mathtt{R}_Z$ around the $\mathtt{Z}$ axis,

$$\mathtt{E} \sim \mathtt{U}\,\mathrm{diag}(1,1,0)\mathtt{V}^\top = \mathtt{U}\mathtt{R}_Z\,\mathrm{diag}(1,1,0)\mathtt{R}_Z^\top\mathtt{V}^\top \tag{3.12}$$

This fact has been analyzed in Bartoli's proposal and two work-arounds have been mentioned. The first requires special optimization techniques whereas the second actually suggests not to use essential matrices at all by shifting the origin of the coordinate frame off the image center. This is contrary to the prenormalization step described in section 2.6.2 in which the origin is shifted near the optical axis for general stereo camera configurations.

An approach for the minimal parametrization of an *essential matrix* $\mathtt{E}$ may be found in equation (2.18) and corresponds to (3.6) in the intrinsically calibrated case. Here, the rotation matrix $\mathtt{R}'$ similarly has to be parameterized locally in order to avoid gimbal lock configurations. Finally, two parameters have to describe the direction of $\mathbf{t}'$.

Similarly to the last section, a factorization of the essential matrix decoupling epipoles from the epipolar line homography has not been proposed yet.

3.3.2 Householder-based Parametrization

This work introduces an approach based on Householder transformations $\mathbb{P}^2 \to \mathbb{P}^2$. A Householder matrix $\mathtt{H}_H$ (see also appendix C) is symmetric, has two degrees of freedom and represents a reflection by a hyperplane with normal $\mathbf{w}$. Hence, it is part of $O(3)\backslash SO(3)$ so that $\det(\mathtt{H}_H) = -1$. Given two different vectors $\mathbf{e}, \tilde{\mathbf{e}} \in \mathbb{R}^3$ of equal length, a Householder transformation $\mathtt{H}_H$ reflecting $\mathbf{e}$ to $\tilde{\mathbf{e}}$ is defined by

$$\mathtt{H}_H = \mathtt{I} - 2\frac{\mathbf{w}\mathbf{w}^\top}{\mathbf{w}^\top\mathbf{w}} \quad \text{, where} \tag{3.13}$$

$$\mathbf{w} = \mathbf{e} - \tilde{\mathbf{e}} \tag{3.14}$$

It can be easily seen that the transformation matrix $\mathtt{H}_H$ is orthogonal and does not depend on the magnitude of $\mathbf{w}$. This property is identical to the global scaling equivalence of homogeneous coordinates so that equation (3.13) is also valid for projective space $\mathbb{P}^2$. Section 3.2 has shown how a vector $\mathbf{e} \in \mathbb{P}^2$ may be described with two parameters. Hence, $\mathtt{H}_H$ may be minimally parameterized. An advantage of Householder transformations is that the transformation matrix $\mathtt{H}_H$ does not need to be computed explicitly.

It may be noticed that a Householder matrix may be represented by a single projective point $\mathbf{e}$ or $\mathbf{w} \in \mathbb{P}^2$. If $\mathbf{w}$ is used, equation (3.13) even may be simplified as

$$\mathtt{H}_H = \mathtt{I} - 2\mathbf{w}\mathbf{w}^\top \quad \text{if} \quad \|\mathbf{w}\| = 1. \tag{3.15}$$

In this case, the number of operations for a Householder transformation given $\mathbf{w}$ reduces to 6 multiplications and 5 additions/subtractions. This is less than the number of operations for a matrix-vector multiplication in $\mathbb{P}^2$ (9 multiplications and 6 additions).

If $\mathbf{e}$ is close or identical to $\tilde{\mathbf{e}}$, its negative version $-\tilde{\mathbf{e}}$ as a target vector avoids numeric instabilities in equation (3.13) and leads to an alternative mirror vector $\mathbf{w}^* = \mathbf{e} + \tilde{\mathbf{e}}$. More details on Householder transformations can be found in [GvL96].

Proposition 3.3.1 *For each rank-2 matrix* $\mathtt{F}$*, there exist orthogonal reflection matrices* $\mathtt{H}_H$ *and* $\mathtt{H}'_H \in O(3)\backslash SO(3)$ *such that*

$$\mathtt{F} = \mathtt{H}'_H\tilde{\mathtt{F}}\mathtt{H}_H \qquad \textit{with} \tag{3.16}$$

$$\tilde{\mathtt{F}} = \mathtt{H}'_H\mathtt{F}\mathtt{H}_H = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \tilde{\mathtt{F}}_L & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix} \tag{3.17}$$

Proof Let $\mathbf{e} = \left(e_x, e_y, e_w\right)^\top$ and $\mathbf{e}' = \left(e'_x, e'_y, e'_w\right)^\top$ be the left and right null-space of $\mathtt{F}$ and let $\mathtt{H}_H$ be the Householder matrix mapping $\mathbf{e}$ to a target vector $\tilde{\mathbf{e}} = \mathtt{H}_H\mathbf{e} = \|\mathbf{e}\|\left(0, 0, \pm 1\right)^\top$ on the $\tilde{\text{Z}}$-axis, $\mathtt{H}'_H$ likewise. Then equations (3.16), (3.17) hold as, in transformed space, $\tilde{\mathbf{e}} = \|\mathbf{e}\|\left(0, 0, \pm 1\right)^\top$ requires the bottom row of $\tilde{\mathtt{F}}$ to be zero. Consequently, $\tilde{\mathbf{e}}'$ requires the right column of $\tilde{\mathtt{F}}$ to be zero as well. ∎

Note that in this proposition, the signs of $\tilde{\mathbf{e}}, \tilde{\mathbf{e}}'$ can be chosen arbitrarily. However, section 3.3.6 will show that they may be fixed mutually if the determinant of $\tilde{\mathtt{F}}$ is required to be positive. In particular, a positive determinant will be assumed in proposition 3.3.2.

3.3.3 Unknown Internal Camera Parameters

Proposition 3.3.1 can be used for decomposing fundamental matrices as epipoles are the left and right null-space of a fundamental matrix $\mathtt{F}$. $\tilde{\mathtt{F}}$ has four elements, so it has three degrees of freedom. Moreover, it does not depend on epipoles.

The decomposition has the following properties:

- The two epipoles can be related by

$$\mathbf{e}' = \mathtt{R}_H \mathbf{e} \quad \text{with} \quad \mathtt{R}_H = \mathtt{H}'_H \mathtt{H}_H \tag{3.18}$$

  Here, $\mathtt{R}_H$ is a rotation matrix as it is the product of two reflections. Due to construction of $\mathtt{H}'_H$ and $\mathtt{H}_H$, the rotation axis of $\mathtt{R}_H$ depends on the constellation of both epipoles with respect to the $\tilde{\mathtt{z}}$-axis.

- The singular values of $\mathtt{F}$ and $\tilde{\mathtt{F}}$ are identical as shown in appendix B.

If epipoles are known a priori, at least three non-collinear point correspondences in transformed space $\tilde{\mathbf{x}}_i = \mathtt{H}\mathbf{x}_i$ and $\tilde{\mathbf{x}}'_i = \mathtt{H}'\mathbf{x}'_i$ are sufficient for determining $\tilde{\mathtt{F}}$. If they are not known, which is usually the case, the proposed decomposition can already be used as an alternative to equation (3.10). However, the scale of $\tilde{\mathtt{F}}$ is not fixed yet.

Minimal Parameterization of a Fundamental Matrix

This section describes how $\tilde{\mathtt{F}}$ may be represented by three parameters. A minimal parametrization is possible by decomposing $\tilde{\mathtt{F}}_L$ of equation (3.17) in a similar way.

Proposition 3.3.2 *For each regular matrix $\tilde{\mathtt{F}}_L \in \mathbb{R}^{2,2}, \det(\tilde{\mathtt{F}}_L) > 0$, there exist two reflections $\mathtt{H}_{HL}$, $\mathtt{H}'_{HL}$ and non-zero parameters $\sigma_1, \sigma_2 > 0$ such that $\tilde{\mathtt{F}}_L$ can be written as*

$$\tilde{\mathtt{F}}_L = \mathtt{H}'_{HL} \mathtt{D} \mathtt{H}_{HL} = \mathtt{H}'_{HL} \operatorname{diag}(\sigma_1, \sigma_2) \mathtt{H}_{HL} \tag{3.19}$$

Proof Let $\mathtt{A}_L = \mathtt{U}\Sigma\mathtt{V}^\top = \mathtt{U} \operatorname{diag}(\sigma_1, \sigma_2)\mathtt{V}^\top$ be the SVD of $\mathtt{A}_L$. Then, $\mathtt{U}$ and $\mathtt{V}$ are orthogonal, i. e. may represent reflections or rotations. If both represent reflections, then $\mathtt{H}'_{HL} = \mathtt{U}$, $\mathtt{H}'_{HL} = \mathtt{V}^\top$ and $\mathtt{D} = \Sigma$. If not, then both represent rotations, due to the condition that $\det(\mathtt{A}), \sigma_1, \sigma_2 > 0$, and the reflection property can be achieved by switching signs of e. g. the last columns of $\mathtt{U}$ and $\mathtt{V}$. ∎

It can easily be verified by looking at the proof above that the 2D reflection matrices have the form

$$\mathtt{H}_{HL} = \begin{bmatrix} \cos\alpha_L & \sin\alpha_L \\ \sin\alpha_L & -\cos\alpha_L \end{bmatrix}, \quad \mathtt{H}'_{HL} = \begin{bmatrix} \cos\alpha'_L & \sin\alpha'_L \\ \sin\alpha'_L & -\cos\alpha'_L \end{bmatrix}.$$

Hence, they can be described with single parameters α_L, α'_L.

Finally, as the overall scale does not matter, the determinant of $\tilde{\mathtt{F}}_L$ may be fixed to $\det(\tilde{\mathtt{F}}_L) = 1$ so that a single parameter σ_L can represent $\mathtt{D} \sim \mathrm{diag}(\sigma_L, \sigma_L^{-1})$.

Equations (3.16), (3.17), and (3.19) may be used to minimally parameterize a fundamental matrix $\mathtt{F}$ with seven parameters as

$$\mathtt{F} \sim \mathtt{H}'_H \begin{bmatrix} \mathtt{H}'_{HL} & 0 \\ 0 & 1 \end{bmatrix} \mathrm{diag}(\sigma_L, \sigma_L^{-1}, 0) \begin{bmatrix} \mathtt{H}_{HL} & 0 \\ 0 & 1 \end{bmatrix} \mathtt{H}_H. \tag{3.20}$$

In this equation, $\tilde{\mathtt{F}}$ can be represented by three parameters $(\alpha_L, \alpha'_L, \sigma_L)$ whereas four are sufficient for defining $\mathtt{H}_H$ and $\mathtt{H}'_H$. Recall that this factorization is of the form (3.3).

3.3.4 Known Internal Camera Parameters

The statements of the last section can be transferred to the case in which internal camera parameters are known. Proposition 3.3.1 is also valid for normalized coordinates and normalized epipoles $\hat{\mathbf{e}}$, $\hat{\mathbf{e}}'$. When looking at the relationship (2.23) between the fundamental and the essential matrix, it is obvious that normalized versions $\mathtt{K}^{-1}\mathtt{H}$ and $\mathtt{K}'^{-1}\mathtt{H}'$ in equation (3.16) generally are no longer unitary and symmetric. Hence, the two Householder matrices $\hat{\mathtt{H}}$ and $\hat{\mathtt{H}}'$ have to be determined in normalized space and yield a similar relationship:

$$\mathtt{E} = \hat{\mathtt{H}}'_H \tilde{\mathtt{E}} \hat{\mathtt{H}}_H \tag{3.21}$$

$$\tilde{\mathtt{E}} = \hat{\mathtt{H}}'_H \mathtt{E} \hat{\mathtt{H}}_H = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.22}$$

As reflections do not change singular values as shown in appendix B, $\tilde{\mathtt{E}}$ in equation (3.21) is still an essential matrix. It is rank-2 but must have one degree of freedom. A constructive explanation is given in the following.

Due to proposition 3.3.1 and the choice of the target vectors $\tilde{\mathbf{e}}$, $\tilde{\mathbf{e}}'$, the two transformed epipoles can be found on the $\tilde{\mathrm{z}}$-axis. If they do not differ in sign, then the relative pose between the two cameras in transformed space can be restricted: The translation $\tilde{\mathbf{t}}'$ can only be along the $\tilde{\mathrm{z}}$-axis, whereas the rotation $\tilde{\mathtt{R}}'$ is limited to a Givens rotation around it by an angle φ_E. Using these properties, equation (2.18) and the convenient normalization of $\left\|\tilde{\mathbf{t}}'\right\| = 1$, $\tilde{\mathtt{E}}$ can be constructed as

$$\tilde{\mathbf{t}}' = \begin{pmatrix}0,0,1\end{pmatrix}^\top \tag{3.23}$$

$$\tilde{\mathtt{R}}' = \begin{bmatrix}\cos\varphi_E & -\sin\varphi_E & 0\\ \sin\varphi_E & \cos\varphi_E & 0\\ 0 & 0 & 1\end{bmatrix} \tag{3.24}$$

$$\tilde{\mathtt{E}} = \left[\tilde{\mathbf{t}}\right]'_\times \tilde{\mathtt{R}}' = \begin{bmatrix}0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\cos\varphi_E & -\sin\varphi_E & 0\\ \sin\varphi_E & \cos\varphi_E & 0\\ 0 & 0 & 1\end{bmatrix} \tag{3.25}$$

$$= \begin{bmatrix}-\sin\varphi_E & -\cos\varphi_E & 0\\ \cos\varphi_E & -\sin\varphi_E & 0\\ 0 & 0 & 0\end{bmatrix} \tag{3.26}$$

If normalized epipoles are known, the problem of estimating an essential matrix $\mathtt{E}$ is univariate as, by rewriting the correspondence condition (2.18), a single correspondence in transformed space determines $\tilde{\mathtt{E}}$:

$$\varphi_E = \arctan\frac{\tilde{x}\tilde{y}' - \tilde{x}'\tilde{y}}{\tilde{x}\tilde{x}' + \tilde{y}\tilde{y}'} \tag{3.27}$$

If epipoles are unknown, the decomposition can be used as a minimal parametrization of an essential matrix consisting of the two epipoles $\hat{\mathbf{e}}$, $\hat{\mathbf{e}}'$ and an additional angle φ_E

$$\mathtt{E} = \hat{\mathtt{H}}'_H(\hat{\mathbf{e}}')\tilde{\mathtt{E}}(\varphi_E)\hat{\mathtt{H}}_H(\hat{\mathbf{e}}) \tag{3.28}$$

Equivalently to the observations for unknown intrinsic camera parameters, equation (3.28) is of the form (3.3). The relationship of $\tilde{\mathtt{E}}(\varphi_E)$ to the epipolar line homography will be underlined below.

3.3.5 Relationship to the Epipolar Line Homography

The proposed decomposition can be used to directly identify both components of epipolar geometry: The epipoles as well as the epipolar line homography. The following observations may be helpful:

- $\tilde{\mathtt{F}}$ can be written as a product of a singular diagonal matrix, an orthogonal matrix $\mathtt{W}$

defined in equation (2.33), and an affine homography $\mathtt{L}$ in the following way:

$$\tilde{\mathtt{F}} = \operatorname{diag}(1,1,0)\mathtt{WL} \tag{3.29}$$

$$= \begin{bmatrix}1&0&0\\0&1&0\\0&0&0\end{bmatrix}\begin{bmatrix}0&-1&0\\1&0&0\\0&0&1\end{bmatrix}\begin{bmatrix}b&d&0\\-a&-c&0\\0&0&1\end{bmatrix} \tag{3.30}$$

$$= \begin{bmatrix}0&-1&0\\1&0&0\\0&0&0\end{bmatrix}\begin{bmatrix}b&d&0\\-a&-c&0\\0&0&1\end{bmatrix} \tag{3.31}$$

$$= \begin{bmatrix}a&b&0\\c&d&0\\0&0&0\end{bmatrix}. \tag{3.32}$$

- $\tilde{\mathtt{E}}$ can be written as a product of a singular diagonal matrix, an orthogonal matrix $\mathtt{W}$ defined in equation (2.33), and a rotation $\tilde{\mathtt{R}}'$ in the following way:

$$\left[\tilde{\mathbf{t}}\right]'_\times = \operatorname{diag}(1,1,0)\mathtt{W} \tag{3.33}$$

$$\Rightarrow \quad \tilde{\mathtt{E}} = \operatorname{diag}(1,1,0)\mathtt{W}\tilde{\mathtt{R}}' \tag{3.34}$$

$$= \operatorname{diag}(1,1,0)\begin{bmatrix}0&-1&0\\1&0&0\\0&0&1\end{bmatrix}\begin{bmatrix}\cos\varphi_E&-\sin\varphi_E&0\\\sin\varphi_E&\cos\varphi_E&0\\0&0&1\end{bmatrix} \tag{3.35}$$

$$= \begin{bmatrix}-\sin\varphi_E&-\cos\varphi_E&0\\\cos\varphi_E&-\sin\varphi_E&0\\0&0&0\end{bmatrix}. \tag{3.36}$$

It can be verified that $\mathtt{L}$ in equation (3.29) is the affine epipolar line homography in transformed space and that it reduces to the rotation $\tilde{\mathtt{R}}'$ of equation (3.24) in the intrinsically calibrated case. A geometric interpretation will be given in the following section.

Geometric Interpretation

The proposed decomposition enables a fully geometric interpretation of the correspondence equation (2.23). First, the internally calibrated case is considered. Figure 3.13 shows a stereo setup again. Figure 3.14 illustrates the result of Householder transformations as the baseline is aligned with the positive $\tilde{\mathtt{Z}}$-axis. In 3.15, the remaining parameter φ_E of $\tilde{\mathtt{E}}$ can be seen. It is defined such that corresponding epipolar lines coincide after rotation. The illustrated steps are similar to the process of image rectification, even though in that case epipoles are usually mapped onto the X-axis so that epipolar lines in the image become horizontal [HZ03].

The decomposition of the fundamental matrix yields a similar interpretation. However, after aligning both coordinate systems with the baseline, the residual transformation is not a rotation around the origin but an affine one.

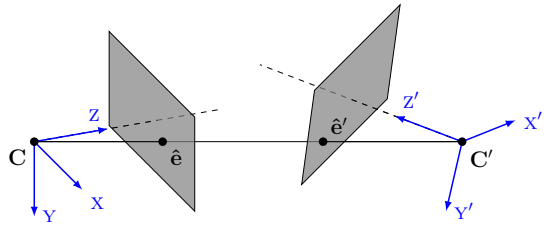


Figure 3.13: Geometric interpretation I: Stereo setup

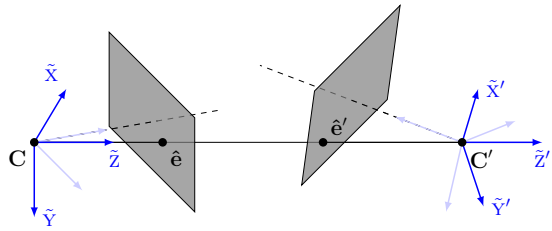


Figure 3.14: Geometric interpretation II: Setup after aligning z-axes with the baseline

Figure 3.16 shows a geometric interpretation of the uncalibrated correspondence condition.

It is worth noting that, according to section 2.1.4, a singular matrix represents a projection. As both the fundamental and the essential matrix are rank-2, they can be considered as projection matrices as well. In figure 3.15, this property can be identified: After both coordinate systems are aligned by φ_E, epipolar lines coincide if their $\tilde{\mathrm{Z}}$ components are ommitted, hence, if they are projected onto the $\tilde{\mathrm{X}}\tilde{\mathrm{Y}}$-planes. This is done with $\mathrm{diag}(1, 1, 0)$ in equations (3.29) and (3.33).

3.3.6 Relationship to the Singular Value Decomposition

Even though the introduced approach is motivated by parameterizing the fundamental and the essential matrix explicitly by using information about epipoles, it is closely related to their SVD [Har92, BS04].

In the singular value decomposition of a fundamental matrix (2.26), $\mathtt{U}$ and $\mathtt{V}$ are $O(3)$ matrices, hence may represent reflections or rotations. However, their signs may be switched as $\mathtt{F} = \mathtt{U}\Sigma\mathtt{V}^\top \sim -\mathtt{U}\Sigma\mathtt{V}^\top$. Thus it can be ensured that $\mathtt{U}$ and $\mathtt{V}$ are $SO(3)$ matrices [BS04]. Note that the last columns of $\mathtt{U}$ and $\mathtt{V}$ represent epipoles as described in section 2.4.1. Hence, if the rotation property is required, epipoles are mutually fixed with

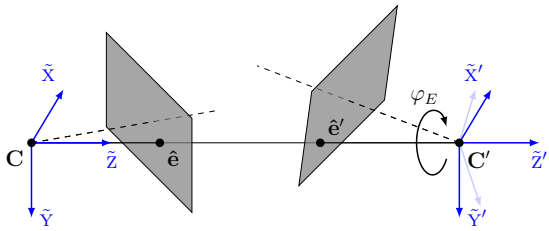
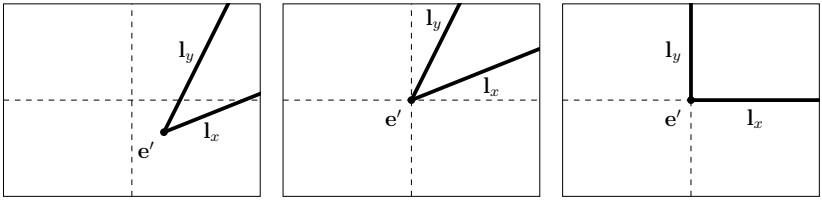


Figure 3.15: Geometric interpretation III: Aligining the two setups with the remaining degree of freedom φ_E



(a) View of an uncalibrated camera including an epipole and two epipolar lines

(b) View after transformation with $\mathtt{H}_H$ and $\mathtt{H}'_H$.

(c) Aligining epipolar lines with the affine epipolar line homography $\mathtt{L}$

Figure 3.16: Geometric interpretation of Householder-based decomposition: Unnormalized coordinates. For illustration purposes, the origin is located in the image center and not, as usual, in the upper left corner. The two shown epipolar lines are chosen to correspond to the coordinate axes.

respect to their signs.

In the decomposition of the fundamental matrix (3.20), it can be easily seen that $\mathtt{U} = \mathtt{H}'_H \begin{bmatrix} \mathtt{H}'_{HL} & 0 \\ 0 & 1 \end{bmatrix}$ and $\mathtt{V}^\top = \mathtt{H}_H \begin{bmatrix} \mathtt{H}_{HL} & 0 \\ 0 & 1 \end{bmatrix}$ and that $\mathtt{U}$ and $\mathtt{V}^\top$ are $SO(3)$ matrices.

For the intrinsically calibrated case, a similar relation to the singular value decomposition can be derived: The SVD of an essential matrix can be seen in eq. (2.20), whereas according to equations (3.26) and (3.28) a Householder-based decomposition can be written as

$$\mathtt{E} \sim \hat{\mathtt{H}}_H \begin{bmatrix} -\sin\varphi_E & -\cos\varphi_E & 0 \\ \cos\varphi_E & -\sin\varphi_E & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{\mathtt{H}}'_H \tag{3.37}$$

$$= \hat{\mathtt{H}}_H \quad [\tilde{\mathtt{t}}]_\times \tilde{\mathtt{R}} \quad \hat{\mathtt{H}}'_H \tag{3.38}$$

The two matrices $\mathtt{U}$ and $\mathtt{V}^\top$ thus can be chosen as

$$\mathtt{U} = \hat{\mathtt{H}}'_H \tag{3.39}$$

$$\mathtt{V}^\top = \begin{bmatrix} -\sin\varphi_E & -\cos\varphi_E & 0 \\ \cos\varphi_E & -\sin\varphi_E & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{\mathtt{H}}_H \tag{3.40}$$

In contrast to Bartoli's approach [BS04], the proposed parametrization yields a unique solution as the ambiguous rotation around the baseline (3.12) does not occur.

3.3.7 Limitations

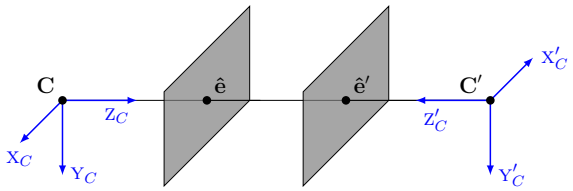
Ambiguous Solutions

Given a singular value decomposition of an essential matrix $\mathtt{E}$, external camera parameters can be extracted as described in section 2.5.3 up to four ambiguous solutions. According to the proposed approach, they can also be seen in figure 3.14: Both $\tilde{\mathtt{z}}$-axes point to the right. By changing their direction, the residual alignment remains identical. This pair of solutions can be identified as the "baseline reversal". The second pair can be identified in fig. 3.15. When rotating one of the two coordinate systems with the epipolar line homography $\tilde{\mathtt{R}}'(\varphi_E)$, there are actually two situations in which the coordinate systems are aligned: φ_E and $\varphi_E + 180°$. In one case, coordinate axes point in the opposite direction, but the correspondence condition (2.23) is still met. This pair of solutions is the "twisted pair" ambiguity. For non-linear optimization, φ_E can be limited e. g. to be in $[-90°, 90°)$. Finally, a direction of the $\mathtt{z}$-axis can be chosen according to the numerical stability of Householder transformations. Details will be given in the following section.

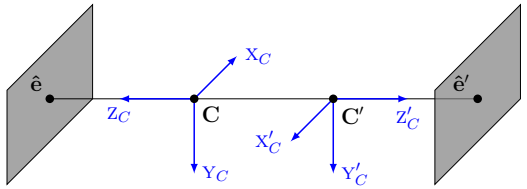
Critical Configurations

As the signs of $\tilde{\mathbf{e}}$ and $\tilde{\mathbf{e}}'$ are mutually fixed due to alignment (see figure 3.14), there is a specific critical configuration in which the proposed approach cannot be used directly. Figure 3.17(a) shows an illustration in which the two cameras are facing each other. Here, the two normalized epipoles can be written as $\hat{\mathbf{e}} = \hat{\mathbf{e}}' = \big(0, 0, 1\big)^\top$. In order for two coordinate systems to be aligned, *one* of the two epipoles has to be reflected to its antipodal version. In this case, however, $\mathbf{w} = \mathbf{0}$ according to section 3.3.2 and equations (3.13) and (3.14), i. e. a Householder matrix does not exist. Therefore, the alternative vector $\mathbf{w}^*$ has to be used for *both* epipoles so that $\tilde{\mathbf{e}} = \tilde{\mathbf{e}}' = \big(0, 0, -1\big)^\top$. Hence, the transformed coordinate systems are still not aligned. Consequently, Householder transformations cannot be used in this configuration. At first glance, this may be surprising as the coordinate systems in figure 3.17 already seem to be aligned except for the orientation of one optical axis.

An equivalent, second, "baseline reversed" case can be seen in figure 3.17(b). However, these two critical configurations do not delimit the proposed decomposition. There is a practical and a theoretical reason for it:



(a)



(b)

Figure 3.17: Critical configuration for Householder-based parametrizations

1. In real applications for three-dimensional computer vision, a setup of two cameras directly facing each other (fig. 3.17(a)) will not be of practical use: Generally, the surface of an object of interest that is located between the two cameras will be visible by one camera only so that no relevant point correspondences can be established. The second critical configuration is even less relevant in practice as in this case, the cameras' fields of view do not intersect at all. Finally, this work focuses on subsequent images of a single moving camera (see section 1.3) so that turnarounds of 180° between two frames will not occur.

2. In order to avoid numerical instabilities completely, a simple preprocessing step illustrated in figure 3.18 may be applied.

   This scheme applies a reflection $\mathtt{H}_\mathtt{z}$ if necessary. Afterwards, the left epipole $\hat{\mathbf{e}}$ is guaranteed to have a non-negative component $\hat{e}_w$, whereas $\hat{e}'_w$ will be non-positive.

Hence, numerically stable target vectors $\tilde{\mathbf{e}} = \begin{pmatrix}0,0,-1\end{pmatrix}^\top$ and $\tilde{\mathbf{e}}' = \begin{pmatrix}0,0,1\end{pmatrix}^\top$ can be used for building Householder transformations which align the coordinate systems of the left and right camera correctly. This step may also be applied in the case of unnormalized coordinates in order to avoid critical configurations for decomposing the fundamental matrix.

3.3.8 Experimental Results

Even though the representations of the last section were motivated by finding a factorization or epipolar geometry, which will be applied in later chapters, experiments using the proposed parametrization have been done with respect to accuracy and speed.

Performance

A first experiment evaluates the computation time by using two parametrizations for weak calibration given synthetic correspondences:

- FULL: Overparametrization (3.6): All 12 entries of $\mathtt{P}'$ for $\mathtt{F} = [\mathbf{p_4}]_\times \mathtt{M}$
- HOUSE: By applying the Householder-based decomposition of a fundamental matrix, the minimum number of parameters are optimized.

The linear solution of the normalized eight-point algorithm has been used to initialize parameters. The Levenberg-Marquardt algorithm with numeric differentiation served as an optimizer for the reprojection error. Execution time of an iteration step using the test system mentioned in section 6.1.3 has been measured for both methods. Results can be found in table 3.4.

Parametrization	Number of correspondences				
	50	100	500	1000	5000
FULL	4.88	4.89	4.84	4.80	4.69
HOUSE	5.23	4.70	4.72	4.65	4.78

Table 3.4: Execution times in μs of an iteration step given different number of correspondences

It can be seen that both methods are equally performant. Even though in HOUSE, fewer parameters have to be optimized, there is an overhead for building the fundamental matrix. As the Levenberg-Marquardt algorithm is a gradient-based optimization technique and operates with normal equations (see [HZ03], appendix A6), computations for both FULL and HOUSE could be accelerated further if the Jacobian is computed analytically [BS04].

Table 3.5 shows the number of iterations for the same scenario. In the case of HOUSE, the Levenberg-Marquardt algorithm always needs fewer iterations compared to the overparametrization FULL, as gauge is avoided.

Parametrization	Number of correspondences				
	50	100	500	1000	5000
FULL	332	387	305	286	306
HOUSE	153	169	161	154	209

Table 3.5: Number of iterations given different number of correspondences

Accuracy

The first two frames of the "corridor" sequence served as real data for evaluating accuracy. True correspondences were available as ground truth (see also chapter 6). In this experiment, mean-zero uncorrelated Gaussian noise (3.1) has beed added to pixel positions $\{\mathbf{x}, \mathbf{x}'\}$. The reprojection error has been minimized during optimization.

The following sets of parameters have been optimized, starting with an initial linear solution INIT:

- FREE: Overparametrization (3.6): all 12 entries of $\mathtt{P}'$
- RTE: 5 parameters of $\mathtt{E} = [\mathbf{t}]'_\times \mathtt{R}'$.
- HHE: 5 parameters of $\mathtt{E} = \hat{\mathtt{H}}'\tilde{\mathtt{E}}\hat{\mathtt{H}}$.

Results can be seen in figure 3.19. Optimized epipoles have been compared to their true values. All parameter sets yield almost identical results. In other experiments, similar observations have been made for different parametrizations [BS04, VT98]. An explanation is that by using the reprojection error as a common cost function, parametrizations do not influence the overall quality.

3.3.9 Conclusion

This section presented a symmetric decomposition of the essential and the fundamental matrix. It can be used as a minimal parametrization in non-linear optimization algorithms and does not suffer from critical configurations due to a simple preprocessing step. Moreover, it explicitly reveals the two components of epipolar geometry: the epipoles and the epipolar line homography. Besides geometric insight of epipolar geometry, experiments have demonstrated that minimal parametrizations can indeed be used to avoid gauge and reduce the number of iterations, and therefore computation time.

Sections 3.3.3 and 3.3.4 mentioned that the proposed decomposition of the essential and fundamental matrix is of the form (3.3). Thus, it is suitable for decomposing the relative pose problem into the combination of an outer estimation problem targeted at the determination of epipoles and a residual inner estimation problem considering the residual epipolar transformation.

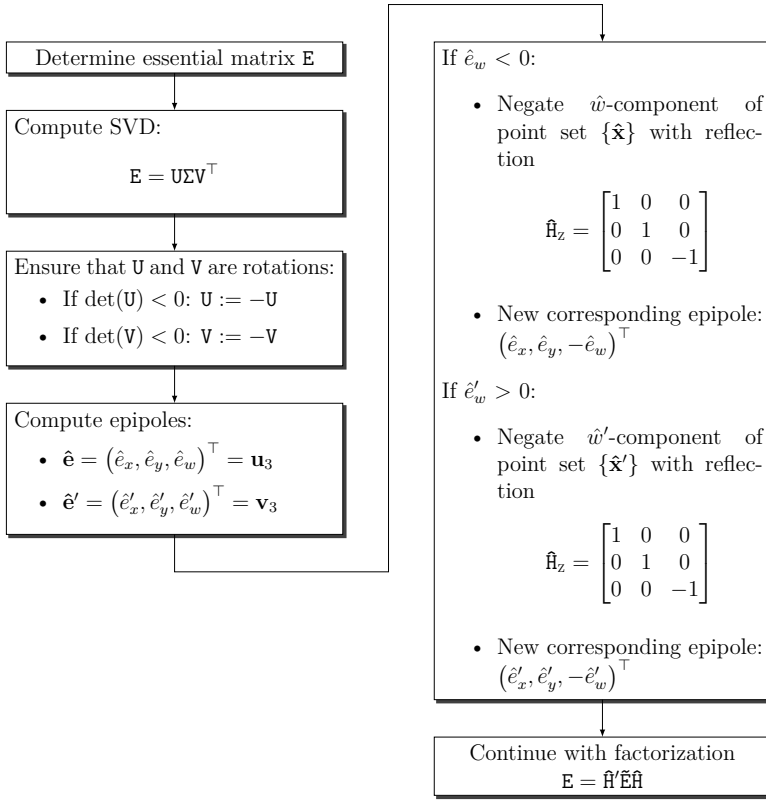


Figure 3.18: Avoiding numerical instabilities for Householder-based parametrizations of epipolar geometry in a preprocessing step

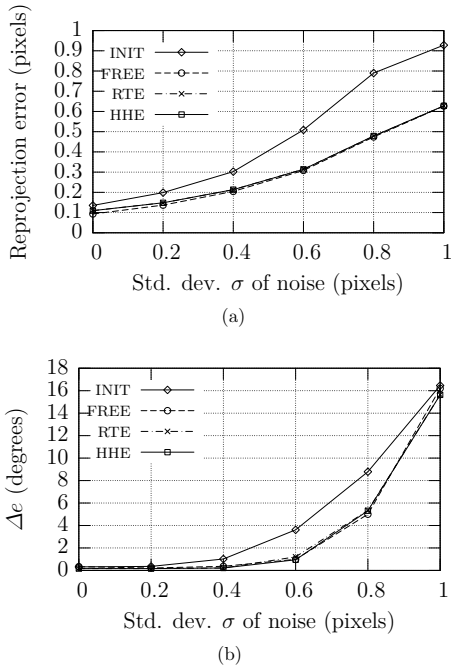


Figure 3.19: (a) Reprojection error, (b) Mean angular error of two epipoles

Chapter 4

PbM-based Relative Pose Estimation

This chapter discusses the first approach which has been motivated in section 3.1.3. It introduces the computation of epipoles in the context of projection-based M-estimation. As opposed to the normalized eight-point algorithm described in section 2.6.2, the estimation of epipoles can be treated as an outer optimization problem, whereas the determination of the epipolar line homography forms an inner optimization problem.

After reviewing basics of projection-based M-estimators (pbM-estimators) in section 4.2, the estimation of essential matrices with pbM-based estimation is laid out. It will be shown that the Householder-based decomposition of an essential matrix is a well-suited parametrization. Section 4.4 describes additional aspects for angular parameters.

4.1 Motivation

A general drawback of the robust estimation techniques discussed in section 2.6.3 is their need for additional information like the scale of inlier noise. It is used as a threshold in order to generate an inlier/outlier dichotomy, i. e. to discriminate inliers from outliers. In most situations, however, the scale of noise is not known, so that estimation is based on assumptions and may lead to erroneous results. Section 2.6.3 mentioned that other methods like least median of squares or a least k-th order statistics implicitly require a certain percentage of data being inliers.

Recently, techniques for autonomous robust estimation have been of increasing interest. Projection-based M-estimators explore a univariate distribution of a residual parameter given an estimate of the others during optimization [Mee04]. It is motivated by a linear errors-in-variables (EIV) model and therefore cannot be applied to many non-linear problems in computer vision, including essential or fundamental matrix estimation. In this case, it has still been used for weak calibration to generate an inlier/outlier dichotomy for postprocessing [Mee04]. In this work, it will be used for applying the outer and inner optimization problem mentioned in section 3.1.3.

4.2 Projection-Based M-Estimation

This section reviews some fundamentals of projection-based M-estimation. Further details can be found in [Mee04], [SM05] or [CM03].

PbM-estimators are based on a linear error-in-variables (EIV) model of the form

$$\mathbf{y}_{i0}^\top \boldsymbol{\theta} - \tau = 0 \qquad i = 1 \dots N \tag{4.1}$$

$$\mathbf{y}_i = \mathbf{y}_{i0} + \delta \mathbf{y}_i \qquad \delta \mathbf{y}_i \sim \mathcal{N}_p(\mathbf{0}, \sigma^2 \mathbf{I}) \tag{4.2}$$

in which $\mathbf{y}_{i0}$ is the true value for a p-dimensional data point $\mathbf{y}_i$ containing additive Gaussian noise with unknown variance σ^2. $\{\boldsymbol{\theta}, \tau\}$ is the partitioned parameter set to be estimated. In the following, $\boldsymbol{\theta}$ is also called *orientation* whereas τ is referred to as *shift*. The scale ambiguity of (4.1) is usually removed by adding a normalization constraint

$$\|\boldsymbol{\theta}\| = 1. \tag{4.3}$$

It is interesting to note that $\left(\boldsymbol{\theta}, \tau\right)^\top$ represents a hyperplane containing all $\left(\mathbf{y}_{i0}, -1\right)^\top, i = 1 \dots N$. Using the linear EIV model for M-estimation yields the following optimization problem:

$$\{\hat{\boldsymbol{\theta}}, \hat{\tau}\} = \underset{\boldsymbol{\theta}, \tau}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \rho \left(\frac{\mathbf{y}_i^\top \boldsymbol{\theta} - \tau}{s} \right) \tag{4.4}$$

In this equation, $\rho(x)$ is a robust error term (see section 2.6.3) whereas s represents the scale of noise. An identical *maximization* problem can be achieved by rewriting equation (4.4)

$$\{\hat{\boldsymbol{\theta}}, \hat{\tau}\} = \underset{\boldsymbol{\theta}, \tau}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} \kappa \left(\frac{\mathbf{y}_i^\top \boldsymbol{\theta} - \tau}{h} \right) \quad \text{, where} \tag{4.5}$$

$$\kappa(x) = c \cdot (1 - \rho(x)) \tag{4.6}$$

On the one hand, equation 4.5 is just a simple rewrite of (4.4). On the other hand, it is an optimization problem that can originally be found in the field of *kernel density estimation* [Sco92]. Here, a bandwidth factor h substitutes the scale parameter s in equation. κ is called kernel function and is required to have the following properties [CM02a]:

$$\kappa(x) = \kappa(-x) \geq 0 \tag{4.7}$$

$$\kappa(x) = 0 \text{ for } \|x\| > 1 \tag{4.8}$$

$$\int_{-\infty}^{\infty} \kappa(x) d\kappa = 1 \tag{4.9}$$

$$\kappa(0) \geq \kappa(x) \text{ for } x \neq 0 \tag{4.10}$$

There are many choices for possible kernel functions. Table 4.1 lists some of them.

Name	Kernel
Uniform	$\kappa(x) = c$
Triangle	$\kappa(x) = c \cdot (1 - \lvert x \rvert)$
Epanechnikov	$\kappa(x) = c \cdot (1 - x^2)$
Biweight	$\kappa(x) = c \cdot (1 - x^2)^2$
Triweight	$\kappa(x) = c \cdot (1 - x^2)^3$
Normal	$\kappa(x) = c \cdot \mathcal{N}(0, 1)$
Cosine arch	$\kappa(x) = c \cdot \cos(\frac{\pi}{2} x)$

Table 4.1: Possible kernels for density estimation [Sco92]. All kernels $\kappa(x)$ are supported on $[-1, 1]$ and are zero otherwise.

Scott observed that the quality of a kernel density estimate only marginally depends on the choice of the kernel itself [Sco92]. Therefore, as in other approaches, the kernel corresponding to the biweight loss function is chosen for following computations [CM02a].

Equation (4.4) can be separated into two subproblems: an outer optimization problem for finding $\hat{\boldsymbol{\theta}}$ and an inner optimization problem that detects an optimal $\tau_{\boldsymbol{\theta}}$ given an orientation $\boldsymbol{\theta}$ and a bandwidth estimate $h_{\boldsymbol{\theta}}$:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} \kappa \left(\frac{\mathbf{y}_i^\top \boldsymbol{\theta} - \tau_{\boldsymbol{\theta}}}{h_{\boldsymbol{\theta}}} \right) \tag{4.11}$$

$$\hat{\tau}_{\boldsymbol{\theta}} = \underset{\tau}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} \kappa \left(\frac{\mathbf{y}_i^\top \boldsymbol{\theta} - \tau}{h_{\boldsymbol{\theta}}} \right) = \underset{\tau}{\operatorname{argmax}} f(\tau) \tag{4.12}$$

An optimal $\hat{\tau}_{\boldsymbol{\theta}}$ maximizes $f(\hat{\tau}_{\boldsymbol{\theta}})$. Hence, the solution of (4.5) can be given as $\{\hat{\boldsymbol{\theta}}, \hat{\tau}_{\hat{\boldsymbol{\theta}}}\}$. This technique is referred to as *projection-based* M-estimation. This name has been chosen due to the nature of the term $\mathbf{y}_i^\top \boldsymbol{\theta}$ that projects data points $\{\mathbf{y}_i\}$ onto a hyperplane of defined by $\boldsymbol{\theta}$ in a first step before finding a mode $\hat{\tau}_{\boldsymbol{\theta}}$.

The two subproblems (4.11) and (4.12) can be approached with different techniques which will be described below. Again, especially if data points $\{\mathbf{y}_i\}$ contain outliers, optimization may yield *local* optima only. Thus, a random sampling framework similar to variations of RANSAC [TZ00] can be used to find initial estimates near a *global* optimum. This technique has been introduced by Chen and Meer as the PBMSAC algorithm [CM03] and is summarized in algorithm 3.

If the parameter set obeys a linear EIV model (4.1), the first step of algorithm 3 would be sufficient as the mode $\hat{\tau}_{\hat{\boldsymbol{\theta}}}$ minimizes (4.4). Otherwise, $\hat{\tau}_{\hat{\boldsymbol{\theta}}}$ does not necessarily have to be optimal. In this case, Chen analyzed the residual distribution $f(\hat{\tau}_{\hat{\boldsymbol{\theta}}})$ and determined a *basin of attraction* around $\hat{\tau}_{\hat{\boldsymbol{\theta}}}$ [CM03]. It is assumed that it contains inliers only. Subsequently, the parameter set can further be optimized with respect to an appropriate error function,

e. g. the reprojection error in case of camera calibration. However, as mentioned in section 2.6.3, robust estimation techniques like M-estimation may be used as step 2 and 3 instead.

Algorithm 3 PBMSAC estimation

1. Repeat n times:
   - Find an initial estimate $\{\boldsymbol{\theta}, \tau\}$ using a subset of measured data
   - Perform non-linear optimization of $\boldsymbol{\theta}$ by solving (4.12)
   - Keep result if leading to a higher density $f(\hat{\tau}_{\hat{\boldsymbol{\theta}}})$
2. Find left and right local minima of the density around the mode $\hat{\tau}_{\hat{\boldsymbol{\theta}}}$ of step (1) to build an inlier/outlier dichotomy
3. Postprocess inliers to find final estimates $\{\hat{\boldsymbol{\theta}}, \tau_{\hat{\boldsymbol{\theta}}}\}$

This work does not follow Chen's approach completely. In particular, a basin of attraction is not computed for discriminating inliers from outliers.

4.2.1 Inner Optimization: Mode finding via Kernel Density Estimation

This section discusses the inner optimization problem, i. e. it focuses on how to find $\hat{\tau}_{\boldsymbol{\theta}}$ given an orientation $\boldsymbol{\theta}$. In the context of kernel density estimation, equation (4.12) is considered as finding the *mode* maximizing the *density* $f(\tau)$.

Bandwidth selection

Equation (4.12) shows that the kernel density depends on an additional bandwidth parameter $h_{\boldsymbol{\theta}}$. It has been mentioned in the literature that a correct $h_{\boldsymbol{\theta}}$ is crucial for the quality of kernel density estimaties. Figure 4.1 shows an illustration of a density plot using two different bandwidth parameters on the same measurements. In this example, samples of two Gaussian distributions with $\mu_1 = 4.8$, $\sigma_1 = 0.5$ and $\mu_2 = 5.5$, $\sigma_2 = 0.3$ have been drawn.

Unfortunately, the bandwidth parameter is not known a priori for real measurements.

An optimal bandwidth $\hat{h}_{\boldsymbol{\theta}}$ minimizes the asymptotic mean integrated square error (AMISE) between the estimated and the true density if an Epanechnikov kernel is used [Sco92]. Under certain conditions, $\hat{h}_{\boldsymbol{\theta}}$ can be iteratively determined [RD05]. A brief survey of iterative bandwidth estimation techniques can be found in [WJ95]. For this work, accuracy requirements are relaxed as a single mode is considered. In figure 4.1, it can be identified even if the bandwidth is not optimal.

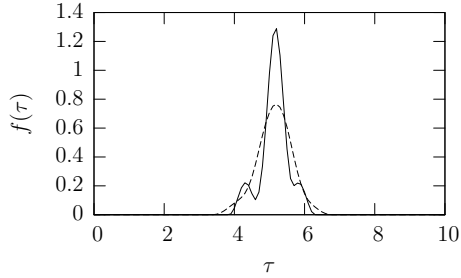


Figure 4.1: Influence of bandwidth parameters for kernel density estimation. It can be seen that a large bandwidth parameter yields a single mode at $\mu_2 = 5.5$ only. In this example, the second at $\mu_1 = 4.8$ can only be identified for $h = 0.5$.

In the following, the bandwidth $h_{\boldsymbol{\theta}}$ for a given orientation $\boldsymbol{\theta}$ is based on the *median of absolute deviations* (MAD) [CM03]

$$h_{\boldsymbol{\theta}} = n^{-1/5} \operatorname*{med}_j |\mathbf{y}_j^\top \boldsymbol{\theta} - \operatorname*{med}_i \mathbf{y}_i^\top \boldsymbol{\theta}|. \tag{4.13}$$

Mode Search

In order to solve equation (4.12), i. e. to find the mode $\hat{\tau}_{\boldsymbol{\theta}}$ given an orientation $\boldsymbol{\theta}$, two strategies are introduced.

Adaptive Sampling Strategy

The adaptive sampling approach introduced by Chen and Meer [CM03] consists of a course and a fine step in which the density $f(\tau)$ is sampled. The course step estimates the density at ten locations defined by equidistant indices of the ordered samples $\mathbf{y}_i^\top \boldsymbol{\theta}$. Hence, dense regions are visited more frequently. The fine step finds an approximation of the mode by sampling a narrow band around the best course candidate. Computation time can be kept constant this way. On the other hand, the distance between two samples determines the accuracy of a mode's location.

Adaptive Sampling is also used in this work.

Mean Shift

For completeness, the mean shift technique is mentioned that finds an optimal mode for a linear EIV model. It has not been used in this work due to the fact that, according to section 4.4, weak calibration will not meet the linear EIV model. Moreover, section 3.1.3 mentioned that an optimal calibration result may be obtained in a following step by minimizing the reprojection error.

A necessary condition for an optimal mode $\hat{\tau}$ is a zero gradient of the true density. The gradient $\nabla f(\tau)$ can be used as an approximation as the true density is unknown. The mean shift algorithm utilizes properties of the density gradient. It has been analyzed by Fukunaga and Hostetler [FH75], and a steepest descent-like algorithm has been introduced to the field of computer vision by Cheng, Comaniciu and Meer [Che95, CM02b].

4.2.2 Outer Optimization

As the bandwidth estimate (4.13) depends on two median values $m_1 = \mathrm{med}_i \, \mathbf{y}_i^\top \boldsymbol{\theta}$ and $m_2 = \mathrm{med}_j \, |\mathbf{y}_j^\top \boldsymbol{\theta} - m_1|$, it will change during optimization and lead to a non-continuous density function $f(\tau_{\boldsymbol{\theta}})$ (4.12). Consequently, derivative-based optimization algorithms may not be applied and other optimization techniques like the Nelder-Mead algorithm have to be used for the outer optimization step.

4.3 PbM-based Weak Calibration

This section summarized the PBMSAC algorithm for weak calibration, i. e. for estimating the fundamental matrix $\mathtt{F}$. A partitioning of $\mathtt{F}$ equivalent to (4.1) has been suggested by Chen [CM03]: As f_{33} is independent of the point correspondences' Euclidean components, it can be used as shift τ_F. In this case, the remaining elements define $\boldsymbol{\theta}_{\mathtt{F}} = \left(f_{11}, f_{12}, f_{13}, f_{21}, \ldots, f_{32}\right)^\top$, whereas the data vector can be written as

$$\mathbf{y}_i = \left(x'x, x'y, x', y'x, y'y, y', x, y\right)^\top .$$

An initial estimate of $\mathtt{F}$ may be scaled such that equation (4.3) is met during optimization. However, in order to avoid gauge, the normalization constraint has to be retained during optimization. Chen suggested to represent the 8 elements of $\boldsymbol{\theta}_{\mathtt{F}}$ by a vector of polar angles $\boldsymbol{\beta} = \left(\beta_1, \beta_2, \ldots, \beta_{n-1}\right)^\top$ such that for $\boldsymbol{\theta} = \left(\theta_1, \theta_2, \ldots, \theta_n\right)^\top$

$$\begin{aligned}
\theta_1 &= \sin\beta_1 \sin\beta_2 \ldots \sin\beta_{n-2} \sin\beta_{n-1} \\
\theta_2 &= \sin\beta_1 \sin\beta_2 \ldots \sin\beta_{n-2} \cos\beta_{n-1} \\
\theta_2 &= \sin\beta_1 \sin\beta_2 \ldots \cos\beta_{n-2} \\
&\vdots \\
\theta_{n-1} &= \sin\beta_1 \cos\beta_2 \\
\theta_n &= \cos\beta_1
\end{aligned} \tag{4.14}$$

Hence, the fundamental matrix is described by a set of seven polar angles $\{\beta_1, \beta_2, \ldots, \beta_7\}_{\mathtt{F}}$ plus a shift $\tau_F = f_{33}$ [CM03]. It can easily be seen that this representation is not optimal:

- Neither does the parametrization of a fundamental matrix accounts for the singularity condition $\det(\mathtt{F}) = 0$, nor for the additional cubic constraint (2.21) in case of an essential matrix $\mathtt{E}$.

- Section 3.2 mentioned that spherical coordinates yield singularities at the poles. As the parametrization (4.14) actually describes points on a hypersphere S^{n-1}, there are $2(n-2)$ singular points $\beta_{1..(n-2)} = \{0, 180°\}$.

4.4 PbM-based Relative Pose Estimation

This section solves the drawbacks described in section 4.3 for the intrinsically calibrated setting. In this case, the Householder-based parametrization of an essential matrix $\mathtt{E}$ can be used for robust projection-based M-estimation. The epipoles $\mathbf{e}$ and $\mathbf{e}'$ or, equivalently, the directions $\mathbf{w}$ and $\mathbf{w}'$ of section 3.3.2, may be represented by four parameters. Their estimation corresponds to the outer optimization problem (4.11). Moreover, there is a residual angle φ_E describing the epipolar line homography which can be estimated via mode finding in an inner optimization step (4.12).

Section 3.3.5 pointed out the explicit presentation of the projective character in epipolar geometry when the proposed Householder-based factorization (3.20), (3.28) is used. However, the linear EIV model (4.1) is not valid for this kind of minimal parametrization as well due to the fact that angles or their 2D-mappings describe an essential matrix. Both are used for trigonometric, non-linear calculations in epipolar geometry.

This is not a drawback: It has been mentioned in section 4.3 that a linear EIV model cannot describe the bilinear character of the correspondence condition anyway.

4.4.1 Global search

The second modification of the projection-based M-estimation technique is an additional global search in parameter space.

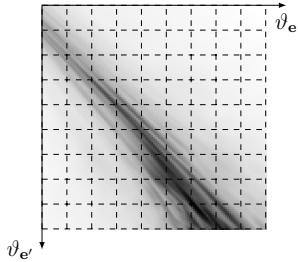


Figure 4.2: Close-up of a density image $f(\tau_{\boldsymbol{\theta}})$ for perfect correspondences. The orientation $\boldsymbol{\theta}$ has been varied by changing co-latitudes of the two epipoles $\vartheta_{\mathbf{e}}$ and $\vartheta_{\mathbf{e}'}$. High density values correspond to dark pixels.

Figure 4.2 shows a 2D density surface $f(\hat{\tau}_{\boldsymbol{\theta}})$. The x- and y-axis of the image correspond to co-latitudes $\vartheta_{\mathbf{e}}$, $\vartheta_{\mathbf{e}'}$ of the two epipoles. The used correspondences are exact. Still, it

can be seen that the density surface is not convex.

In order to avoid local minima, an additional global search may be performed on a parameter subset based on an initial estimate: The space $\vartheta_{\mathbf{e}} \times \vartheta_{\mathbf{e}'} = [0, 90°] \times [0, 90°]$ of co-latitudes is sampled while leaving longitudes $\varphi_{\mathbf{e}}, \varphi_{\mathbf{e}'}$ constant. Sampling has been reduced to co-latitudes $\vartheta_{\mathbf{e}}, \vartheta_{\mathbf{e}'}$ as longitudes are less affected by noise if epipoles are not located near the optical axis (see section 3.1.2). Such a global search has not been used by other camera calibration techniques before due to its computational complexity which is exponential in the number of parameters. Here, it could be reduced to a subset of parameters as a suitable interpretation of the degrees of freedom in epipolar geometry has been utilized.

The modified pbM-estimation algorithm including global search is summarized in algorithm 4.

Algorithm 4 Geometry-driven PBMSAC estimation of relative orientation

1. Repeat n times:
   - Find an initial estimate $\{\boldsymbol{\theta}, \tau\}$ using a subset of measured data
   - Keep result if leading to a higher density $f(\hat{\tau}_{\boldsymbol{\theta}})$
2. Perform a global search on $\vartheta_{\mathbf{e}}, \vartheta_{\mathbf{e}'}$
3. Perform non-linear optimization of $\boldsymbol{\theta}$ by an intermediate mode search for $\tau_{\boldsymbol{\theta}}$

4.4.2 Implementation

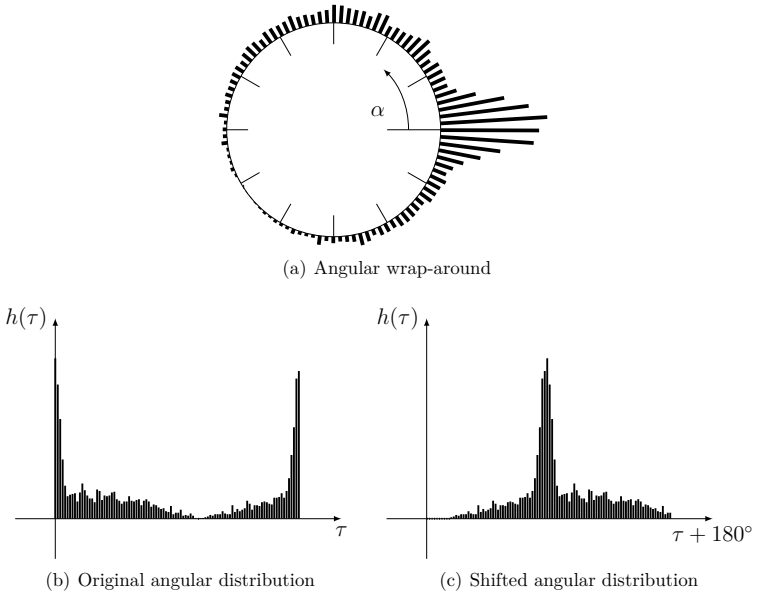
Angular Wrap-Around

Section 3.3.4 has shown how to decompose an essential matrix using Householder transformations.

As $\tau = \varphi_E$ is used as shift for the inner optimization problem, bandwidth estimation becomes crucial: If the mode is located near $\tau = 0°$, wrap-arounds can occur as angles may be shifted to their next periodic locations, according to equation (3.27). In this case, their median might not be close to the true mode so that the MAD yields erroneous bandwidth estimates. In this work, the minimum MAD estimate of an original and a shifted distribution is used. Figure 4.3 shows an illustration.

4.5 Conclusion

This chapter has introduced a geometry-driven method for estimating two cameras' relative pose by means of the combination of random sampling and a projection pursuit. It requires normalized point correspondences and, in contrast to previously existing techniques for pbM-based weak calibration, is able to retain the constraints of an essential matrix during



(a) Angular wrap-around

(b) Original angular distribution

(c) Shifted angular distribution

Figure 4.3: Qualitative illustration of wrap-around problem using angular histograms

processing. The epipoles and the epipolar line homography can be treated as two separate components and have been approached with different optimization techniques such as the Nelder-Mead simplex algorithm and an adaptive mode search.

The proposed technique estimates the scale of noise as a bandwidth for kernel density estimation. Also, a global search step has been introduced that accounts for non-convex error functions. In chapter 6, the proposed approach is evaluated quantitatively and compared to other methods.

Chapter 5

Parallax-based Relative Pose Estimation

This chapter discusses the second approach mentioned in section 3.1.3. It describes the estimation of epipoles given dense motion fields for solving the outer estimation problem described in section 3.1.3. The residual epipolar transformation, i. e. the parameter φ_E in an intrinsically calibrated setting, may then be estimated by the adaptive mode search described in the last chapter.

First, an overview of existing methods for estimating epipoles needs to be given. Some of the subsequently introduced methods only deal with a *single* epipole $\mathbf{e}'$. However, this is not a restriction: When exchanging the order of the two cameras, the second epipole $\mathbf{e}$ may be determined likewise.

5.1 State of the Art

5.1.1 Methods using sparse correspondences

Early approaches

There are many ways to estimate epipoles. The problem was first mentioned by Hesse [Hes63] who considered the epipolar line homography and concluded that the cross-ratio of four corresponding lines (2.4) remains constant for a given setup. It has later been revisited by Luong [LF98] and Nistér [NS04] who noted that four point correspondences give rise to a decic curve of possible epipoles. The space of solutions can be limited by finding six point correspondences. In this case, they constrain an epipole to lie on a plane cubic. Using a cross-ratio-based approach, a unique solution for an epipole may be determined in case of seven correspondences.

Epipoles from Epipolar Geometry

In the context of epipolar geometry, epipoles can be determined after weak calibration as the left and right null-space of a fundamental or an essential matrix (see eq. (2.22) and (2.27)).

Epipoles from Plane and Parallax

$\mathtt{F}$ itself can be decomposed into a product

$$\mathtt{F} = [\mathbf{e}']_\times \mathtt{H}_{\boldsymbol{\pi}} \tag{5.1}$$

of a skew-symmetric matrix $[\mathbf{e}']_\times$ containing the epipole and a non-unique planar homography $\mathtt{H}_{\boldsymbol{\pi}}$ [BM95]. This "plane + parallax" decomposition is motivated by *motion parallax* which describes the fact that two points $\mathbf{X}$ and $\mathbf{X}_{\boldsymbol{\pi}}$ that project to the same point $\mathbf{x}$ but differ in depth are both located at different points on an epipolar line in the second image. Figure 5.1 shows an illustration of motion parallax.

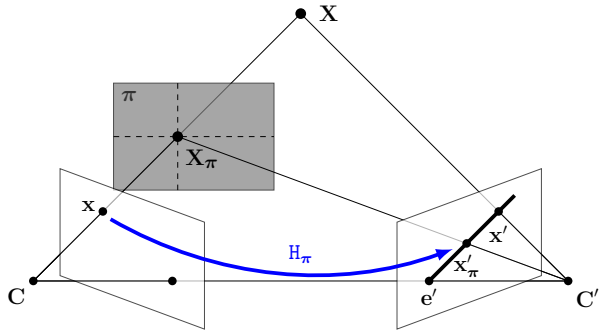


Figure 5.1: Plane and parallax decomposition of epipolar geometry

As the depth of a world point $\mathbf{X}$ is unknown, it is first assumed to be part of an arbitrary, predefined plane $\boldsymbol{\pi}$ in space, even though $\boldsymbol{\pi}$ does not necessarily have to exist. Then, there is a homography $\mathtt{H}_{\boldsymbol{\pi}}$ mapping $\mathbf{x}$ onto the second image. An epipolar line can be constructed given two points: The true correspondence $\mathbf{x}'$ of $\mathbf{x}$ and the transformed point $\mathbf{x}'_{\boldsymbol{\pi}} = \mathtt{H}_{\boldsymbol{\pi}}\mathbf{x}$. Finally, an epipole is computed as the intersection of at least two epipolar lines.

This step requires seven non-planar point correspondences. In this sense, it is equivalent to the standard weak calibration technique of section 2.5.

Epipoles from a Linear Subspace Constraint: Optical Flow

Other techniques do not involve epipolar geometry and are based on the optical flow, the instantaneous velocity field as introduced in section 2.6.1.

Longuet-Higgins and Prazdny [LHP80] showed that a 3D velocity vector $\mathbf{V}_E$ (2.36) can be resolved into a *translational* component $\mathbf{T}$ and *rotational* part $\mathbf{\Omega}$ as

$$\mathbf{V}_E = -\mathbf{T} - \mathbf{\Omega} \times \mathbf{X}_E \tag{5.2}$$

They further showed that the corresponding translational velocity in an image

$$\mathbf{v}_T = \begin{pmatrix} v_{xT} \\ v_{yT} \end{pmatrix} \tag{5.3}$$

points towards the epipole.

Based on this observation, Jepson and Heeger estimated the location of an epipole by deriving $N - 6$ epipolar lines $\mathbf{l}_i, i = 1 \dots N - 6$ given a set of optical flow vectors at N discrete points [JH92, HJ92]. An estimate of an epipole $\mathbf{e}'$ minimizing a least-squares error can be dermined by finding the eigenvector corresponding to the smallest eigenvalue of

$$\mathtt{T} = \sum_i \mathbf{l}_i \mathbf{l}_i^\top \tag{5.4}$$

As at least two epipolar lines are needed for computing their intersection, velocities at $N \geq 8$ image points need to be available.

In a similar approach, Verri and Trucco [VT98] derived a non-linear constraint for at least $N = 6$ optical flow vectors.

Epipoles from a Linear Subspace Constraint: Point Matches

Lawn and Cipolla [LC96] modified Jepson and Heeger's approach by estimating an epipolar line $\mathbf{l}_i$ given four point correspondences in small image patches. This is possible by assuming that the motion of an image patch is affine from one image to the next. Section 5.2.3 gives some more insight into local motion fields.

The affine motion assumption has been avoided by Ponce and Genc [PG98]. Their approach is based on finding a new basis of projective space given four non-planar point correspondences. In transformed space, an epipole is then found by four additional corresponding points.

5.1.2 Methods using dense correspondences

Longuet-Higgins and Prazdny noted that the translational component $\mathbf{v}_T$ of optical flow (5.3) can be directly obtained by analyzing the motion of two points that project to the same point $\mathbf{x}$ but differ in depth [LHP80].

This property is important at object borders: Let $\mathbf{F}$ be a contour point of some foreground object $\mathcal{F}$, i. e. the point on a ray $\mathcal{L}$ that is tangent to $\mathcal{F}$, and let $\mathbf{B}$ be a point of a scene background $\mathcal{B}$ that lies on $\mathcal{L}$ as well. When looking at the displacement vector field $\mathbf{u}(\mathbf{x})$ from the left to the right image, two displacements can be found for $\mathbf{x}$: $\mathbf{u}_f$ describes the motion of the foreground from $\mathbf{x}$ to $\mathbf{x}'_f$, $\mathbf{u}_b$ is related to background motion and points

to $\mathbf{x}'_b$. Hence, the epipolar line $\mathbf{l}' = \mathtt{E}\mathbf{x}$, the projection of $\mathcal{L}$, is the one that joins $\mathbf{x}'_f$ and $\mathbf{x}'_b$. Its 2D direction is identical to the vector $\mathbf{u}_d = \mathbf{u}_f - \mathbf{u}_b$. Thus, $\mathbf{l}'$ can be determined if $\mathbf{u}_d \neq \mathbf{0}$, i. e. if the projections of foreground and background at a contour point move differently. Figure 5.2 shows an illustration of both camera views and the epipolar line defined by the displacement vectors of foreground and background. At foreground/background transitions, a dense motion vector field is discontinuous.

Given different rays $\mathcal{L}_i, i \in 1, ..., N$ through $\mathbf{C}$ that are tangent to a foreground object, more than one epipolar line $\mathbf{l}_i$ can be determined. The epipole $\mathbf{e}'$ finally is their intersection.

The fact that two points at motion discontinuities are sufficient for determining an epipole is an indicator for their high density of 3D information. This is consistent with the human visual system: 3D information about a scene is perceived best at motion discontinuities where foreground and background move differently.

Epipoles from Dense Parallax

Longuet-Higgins and Prazdny's observation has been applied first by Rieger and Lawton [RL85] in a time-differential setting.

In their approach, for a point $\mathbf{x}_i$ and an estimated velocity $\mathbf{v}_i(\mathbf{x}_i)$ in an image, adjacent optical flow vectors $\mathbf{v}_j(\mathbf{x}_j)$, $\|\mathbf{x}_{iE} - \mathbf{x}_{jE}\| < \epsilon$ are analyzed. Then, a line $\mathbf{l}_i$ passing through $\mathbf{x}_i$ is fitted into the set of differences

$$\{\Delta_{ij}\mathbf{v} = \mathbf{v}_j - \mathbf{v}_i\}. \tag{5.5}$$

This is done by an eigenvector analysis similar to equation (5.4). In order to focus on true motion discontinuities caused by foreground/background transitions, lines $\mathbf{l}_i$ are discarded if the following relation holds:

$$\|\lambda_{i,\text{small}}/\lambda_{i,\text{large}}\| > t \tag{5.6}$$

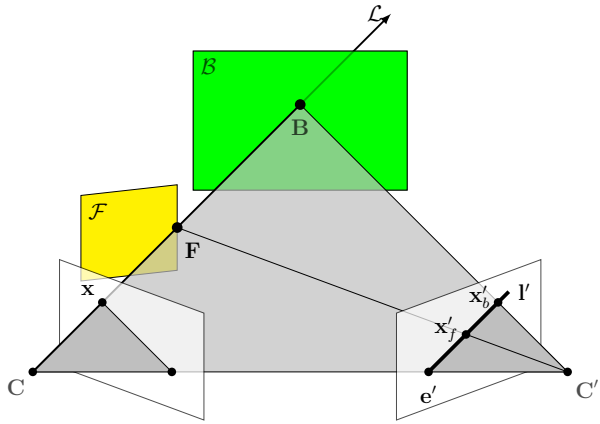
Here, $\lambda_{i,\text{small}}$ and $\lambda_{i,\text{large}}$ are the small and large eigenvalue of $\sum_i (\Delta_{ij}\mathbf{v}) (\Delta_{ij}\mathbf{v})^\top$ whereas t is a predefined threshold value. Relation (5.6) ensures that epipolar lines $\mathbf{l}_i$ are kept only if a dominating difference vector $\Delta_{ij}\mathbf{v}$ can be found.

Finally, a least-squares estimate of the epipole is computed as the intersection of remaining lines $\{\mathbf{l}_i\}$ with equation (5.4).

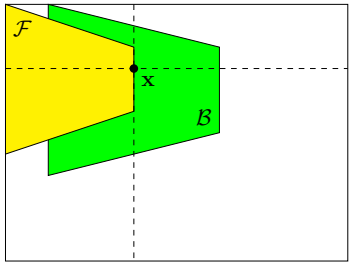
Rieger and Lawton's method has later been refined by Hildreth [Hil91] who used a Hough transform in order to be able to deal with outliers. Figure 5.3 shows a snapshot of the used Hough map. It can be seen that it consists of overlapping accumulator cells of different size covering the camera's field of view.

Several objections can be made regarding this approach:

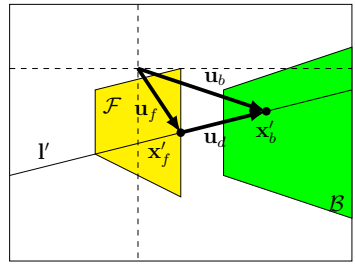
1. Their approach is motivated by flow. However, optical flow describes the instantaneous velocity field, so it is an approximation for point correspondences between two images (see section 2.6.1). Only time-discrete motion vector fields $\{\mathbf{u}\}$ may describe point correspondences correctly [BFBB92].



(a) Stereo setup with foreground object $\mathcal{F}$ and background object $\mathcal{B}$



(b) Left camera view



(c) Right camera view

Figure 5.2: Motion parallax and epipolar line

2. For a point of interest $\mathbf{x}_i$, high accuracy of the corresponding flow vector $\mathbf{v}_i$ at a motion discontinuity is required as it is used for computing each difference $\Delta_{ij}\mathbf{v}$ according to equation (5.5).

3. As analysis is done completely in Euclidean coordinates, the intersection of the set of lines $\{\mathbf{l}_i\}$ is an ill-posed problem. In particular, it is not possible to deal with epipoles at infinity, similarly to Luong's parametrization (3.10) of the fundamental matrix.

4. Hildreth's improvement added a Hough transform in order to deal with outliers. However, similar to the previous objection, the Hough map only covered the image area. Hence, the approach cannot be used for non-visible epipoles.

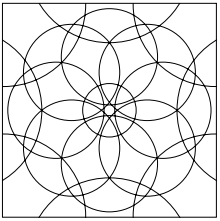


Figure 5.3: Hildreth's Hough map for estimating epipoles

Despite those shortcomings, the described way of directly estimating epipoles is appealing because of the error distribution in optical flow estimates: When integrating smoothness constraints in an motion estimator, outliers are not isolated measurements in a neighborhood of inliers but they can most likely be identified as erroneous clusters. A demonstration can be seen in figure 5.4. It shows the distribution of outliers in the case of dense motion estimation [vdHWG06] using the Yosemite sequence. The image considers motion vectors with an angular error $\Delta e > 5°$ as outliers and shows them as black pixels.

As a quantitative measure for the distribution of outliers, the number of outliers for each motion vector in its 8-neighborhood has been measured for different values of additive Gaussian noise $\mathcal{N}(0, \sigma)$. Figure 5.5 shows that the average number of neighbors for outlier vectors decreases with σ but remains above a value of 7.

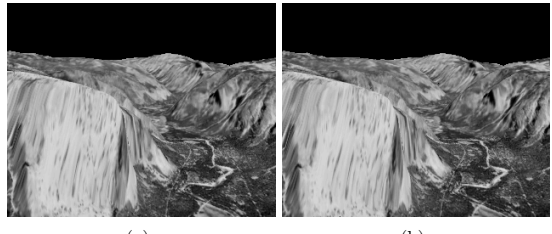
A random sampling algorithm like RANSAC neglects spatial coherence. Also, dense motion field estimation has improved during the last years (see e. g. [BBPW04, AJ05] for recent implementations) so that the objection concerning inaccurate estimates at motion boundaries does not generally hold.

In this chapter, a parallax-based approach to the estimation of epipoles is introduced that overcomes the mentioned shortcomings of Rieger and Lawton's approach. The following aspects give an outline:

1. *Motion vectors*: The proposed method is motivated by a time-discrete motion vector field as described in section 5.1.1

2. *Accuracy requirements*: For a point of interest $\mathbf{x}_i$, all motion vector $\mathbf{u}_j$ in its neighborhood are covered by a two-dimensional complex filter kernel. Section 5.2 shows that if an odd filter kernel size is used, the vector $\mathbf{u}_i$ at $\mathbf{x}_i$ is not considered for determining an epipolar line $\mathbf{l}'$. Hence, the presented approach has relaxed requirements with respect to the accuracy of single motion vectors at motion discontinuities.

3. *Outliers*: Similarly to Hildreth, a Hough transform is used for the robust detection of an epipole. Instead of defining the image plane as a Hough map as shown in figure 5.3, the LAMBERT parametrization of section 3.2 is used for experiments in section 5.6 and chapter 6. Thus, the proposed method is able to deal with arbitrary locations of epipoles.



(a) (b)

(c)

Figure 5.4: (a), (b): Two frames of the "yosemite" sequence. (c): Outlier distribution for block-based motion etimates [vdHWG06]

5.2 Filtering the Dense Motion Field

5.2.1 Spatial Filter

The following section describes an approach for filtering a dense motion field that has been inspired by a linear model for motion edges. It has been introduced by Fleet et al. [FBYJ00] and uses two two-dimensional filter kernels for detecting foreground/background discontinuities in the horizontal and vertical motion field. It is first introduced for spatially filtering intensity edges. In section 5.2.2, the technique will be extended to motion vectors. A filter consists of a complex-valued angular sinusoidal filter kernel b of size D:

$$b = \begin{cases} \cos(\phi) - i\sin(\phi) & \text{if } \| \begin{pmatrix} x, y \end{pmatrix} \| < D/2, \begin{pmatrix} x, y \end{pmatrix}^\top \neq \mathbf{0} \\ 0 & \text{for } \| \begin{pmatrix} x, y \end{pmatrix} \| \geq D/2 \text{ or } \begin{pmatrix} x, y \end{pmatrix}^\top = \mathbf{0} \end{cases} \text{, where} \tag{5.7}$$

$$\phi = \arctan(y/x) \tag{5.8}$$

A mean-zero, unit amplitude intensity edge $E(\mathbf{x})$ and the filter kernel b can be seen in figure 5.6.

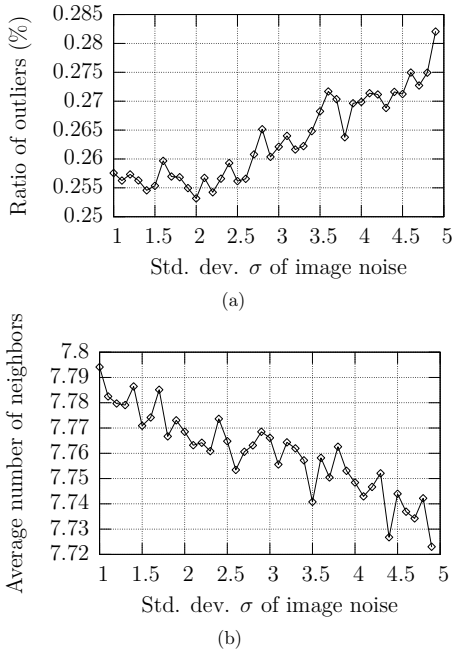
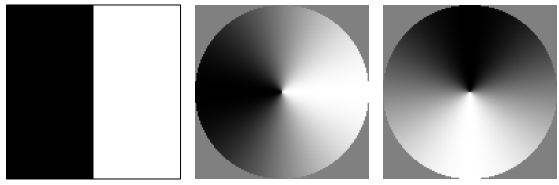


Figure 5.5: Number of adjacent outliers for each motion vector for different noise levels

Real and imaginary part of b form a quadrature pair. This way, the filter kernel can also be applied to rotated versions $E_\phi(\mathbf{x})$ of an intensity edge. When computing the filter response by convolution of E_ϕ and b

$$\gamma = \alpha + i\beta = \sum_{\mathbf{x}} E_\phi(\mathbf{x}) b(-\mathbf{x}), \tag{5.9}$$

the spatial rotation angle ϕ of the edge is identical to the argument of γ so that $\phi = \arctan(\beta/\alpha)$. The amplitude of a scaled intensity edge, on the other hand, is identical to $\|\gamma\|$, supposed that the kernel is normalized such that $\|b\| = 1$. Thus, the chosen filter resembles the computation of a discrete spatial gradient. Filter responses can be interpreted correctly at edges of homogeneous intensities. In the proposed method, the rotation angle is limited to $\phi \in [-90°, 90°)$, and negative edge amplitudes $\mathrm{sgn}(\alpha)\|\gamma\|$ are allowed in order to avoid ambiguities.

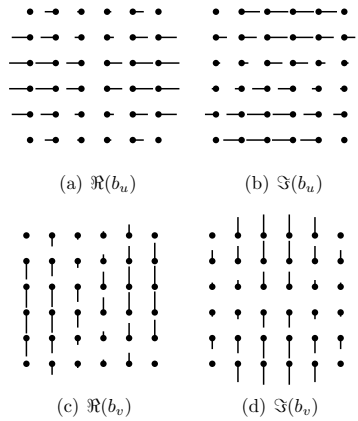


(a) $E(\mathbf{x})$ (b) $\Re(b)$ (c) $\Im(b)$

Figure 5.6: Intensity edge (a). Real (b) and imaginary (c) part of the filter kernel b. Images (b) and (c) have been normalized for visualization. Here, a zero component is represented by a gray value.

5.2.2 Motion Filter

Similar to Fleet's method, a set $\left(b_u, b_v\right)$ is used for the horizontal and vertical component of the motion field $\mathbf{u}(\mathbf{x}) = \left(u, v\right)^\top$ [FBYJ00]. Subsampled versions of the filter kernels are shown in figure 5.7.



(a) $\Re(b_u)$ (b) $\Im(b_u)$

(c) $\Re(b_v)$ (d) $\Im(b_v)$

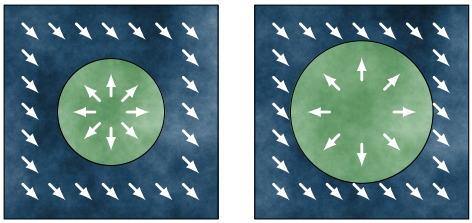
Figure 5.7: Real (a) and imaginary (b) part of the horizontal filter. (c), (d): Real and imaginary part of the vertical filter.

It can be seen in figure 5.7 (a), (d) that the two filter components $\Re(b_u)$ and $\Im(b_v)$ describe occlusion and disocclusion, depending on the sign of the filter response. The two remaining components refer to changes in motion parallel to the spatial orientation of an edge.

Given the motion field, its horizontal and vertical components are filtered by convolution such that filter responses $\gamma_u = \alpha_u + i\beta_u$ and $\gamma_v = \alpha_v + i\beta_v$ exist for each pixel.

Figure 5.8 shows an illustration of a synthetic motion vector field. Its origin is located at the image center. The field can be segmented into a "zooming" circular foreground that is defined by $\mathbf{u}_f(\mathbf{x}) = c_f \left(x, y\right)^\top$ and a translating background with motion vectors $\mathbf{u}_b(\mathbf{x}) = c_b \left(1, 1\right)^\top$. The constants c_f and c_b are chosen such that foreground and background at the lower right border of the foreground move identically. Hence, there is a foreground/background transition at the upper left border of the foreground.

Real and imaginary parts of the filter responses are shown in figure 5.9.



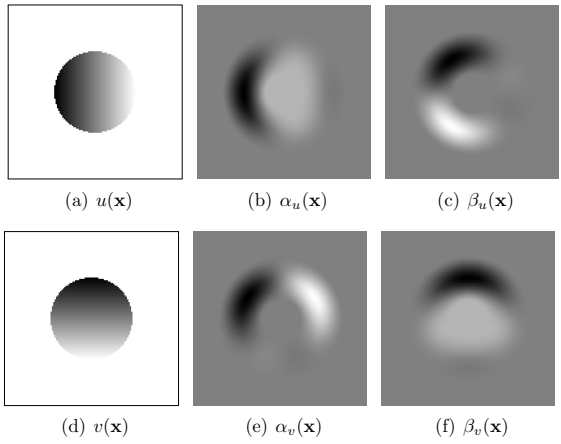
(a) Snapshot of first frame (b) Snapshot of second frame

Figure 5.8: Illustration of synthetic motion field

5.2.3 Differential Homographies

So far, the horizontal and vertical component of the motion field have been processed separately. Hence, phase coefficients ϕ_u, ϕ_v denote the *spatial orientation* of the horizontal and vertical part of a motion edge whereas $\|\gamma_u\|$, $\|\gamma_v\|$ represent their amplitudes. In order to detect motion edges that are caused by foreground/background transitions, incorrect filter responses have to be suppressed. Table 5.1 gives an overview of possible combinations of horizontal and vertical motion components. It can be seen that motion types like rotation (7) or zooming (5) yield responses even though they are caused by continuous motion.

In contrast to the method of Rieger and Lawton [RL85], an eigenvalue analysis like equation (5.6) cannot be applied in order to discard candidates that are not caused by true motion edges as a filter response already covers adjacent motion vectors. However, the observations of table 5.1 suggest an approach based on the phases of the filter responses: Table 5.1 indicates that the spatial orientation of the horizontal and vertical gradient must be identical at motion boundaries caused by different translations of foreground and background (see table 5.1, (3), (4)), and that perpendicular orientations might be caused by zooming or rotation. The following property helps justify that identical orientations of the gradients indeed can be used as an indicator for true motion edges:



(a) $u(\mathbf{x})$ (b) $\alpha_u(\mathbf{x})$ (c) $\beta_u(\mathbf{x})$

(d) $v(\mathbf{x})$ (e) $\alpha_v(\mathbf{x})$ (f) $\beta_v(\mathbf{x})$

Figure 5.9: Motion intensities: (a) Horizontal component, (d) Vertical component. (b),(e): Real parts of filter responses. (c),(f): Imaginary parts of filter responses. Zero motion is represented by gray responses, white and black pixels corresponds to positive and negative values accordingly.

Theorem 5.2.1 *Gradients of a continuous motion field yield identical vertical and horizontal orientations only if local motion propagates to a 2D subspace.*

Proof Let a smooth surface at a scene point $\mathbf{X}_0$ be locally approximated by a plane in the sense that there exists a homography $\mathtt{H}_d$ describing local motion from the first image at $\mathbf{x}_0 = \left(x_0, y_0, w_0\right)^\top = \mathtt{P}\mathbf{X}_0$ to the second at $\mathbf{x}'_0 = \left(x'_0, y'_0, w'_0\right)^\top = \mathtt{P}'\mathbf{X}_0$. Here, $\mathtt{H}_d$ resembles the linear term of a motion field's Taylor approximation at $\mathbf{x}_0$, so that corresponding points $(\mathbf{x}, \mathbf{x}')$ in the neighborhood of $(\mathbf{x}_0, \mathbf{x}'_0)$ can be expressed as $\mathbf{x}' = \mathtt{H}_d\mathbf{x} + \mathbf{r}(\mathbf{x})$ with a residual $\mathbf{r}(\mathbf{x})$ containing second and higher order terms [Vos98]. An illustration can be seen in figure 5.10.

In this work, $\mathtt{H}_d$ is called *differential homography*. It should be emphasized that *differential* is meant *spatially* in this context and not *temporally*, as in optical flow.

Without loss of generality, it will be assumed that $\mathbf{x}_0 = \left(0, 0, 1\right)^\top$. Hence, the horizontal and vertical gradient of a motion vector $\mathbf{u}$ at $\mathbf{x}_0$ can be written as

$$\left.\frac{\partial \mathbf{u}}{\partial x}\right|_{x,y=0} = \frac{1}{{h_{d,33}}^2}\begin{pmatrix} h_{d,33}h_{d,11} - h_{d,31}h_{d,13} \\ h_{d,31}h_{d,21} - h_{d,31}h_{d,23} \end{pmatrix} \tag{5.10}$$

$$\left.\frac{\partial \mathbf{u}}{\partial y}\right|_{x,y=0} = \frac{1}{{h_{d,33}}^2}\begin{pmatrix} h_{d,33}h_{d,12} - h_{d,32}h_{d,13} \\ h_{d,31}h_{d,22} - h_{d,32}h_{d,23} \end{pmatrix} \tag{5.11}$$

	Horizontal Motion	Vertical Motion	Combined Motion		Horizontal Motion	Vertical Motion	Combined Motion
(1)				(5)			
(2)				(6)			
(3)				(7)			
(4)				(8)			

Table 5.1: Basic local motion horizontal and vertical motion edges and combined motion. All shown types already have zero mean motion as it is suppressed by the used filter. It can be seen that the components of (5)–(8) are caused by continuous motion. In this case, spatial orientations of horizontal and vertical motion are perpendicular.

By using (5.10) and (5.11), the condition of identical orientations $\phi_u = \phi_v$ yields

$$\begin{aligned} &h_{d,11}h_{d,22}h_{d,33} + h_{d,12}h_{d,23}h_{d,31} + h_{d,13}h_{d,21}h_{d,32} \\ &-h_{d,13}h_{d,22}h_{d,31} - h_{d,12}h_{d,31}h_{d,33} - h_{d,11}h_{d,23}h_{d,32} = 0 \end{aligned}$$

which is equal to $\det(\mathtt{H}_d) = 0$. Hence, $\mathtt{H}_d$ has to be singular, i. e., according to section 2.1.4, it projects to a subspace of $\mathbb{P}^2$. In the present case, the infinitesimal surface patch at $\mathbf{x}_0$ degenerates to a line at $\mathbf{x}'_0$ in the second image. Figure 5.11 shows an illustration.

Consequently, non-singular homographies do *not* yield identical orientations of the horizontal and vertical motion gradient. ∎

Based on the previous observations, the normalized horizontal and vertical inner product of γ_u and γ_v in vector notation

$$a^*_{uv} = \frac{\left\langle \begin{pmatrix}\alpha_u\\ \beta_u\end{pmatrix}, \begin{pmatrix}\alpha_v\\ \beta_v\end{pmatrix} \right\rangle}{\|\gamma_u\| \, \|\gamma_v\|} \tag{5.12}$$

may be used as a weight for suppressing incorrect filter responses. However, the vertical or horizontal gradient vanishes in situations (1) and (2) of table 5.1. In this case, eq. (5.12)

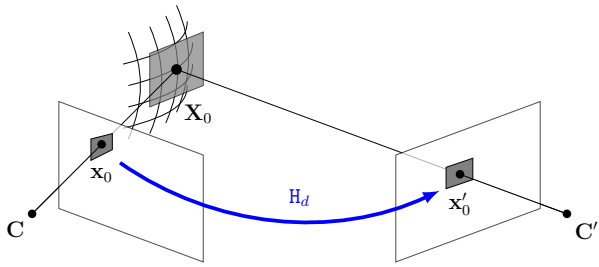
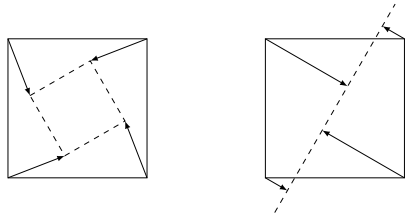


Figure 5.10: Differential homography describing local motion



(a) Regular differential homography

(b) Singular differential homography

Figure 5.11: Only singular differential homographies yield identical spatial orientations of horizontal and vertical motion filter responses in the case of smooth motion.

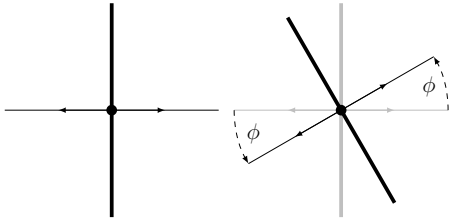
is not defined. Moreover, if using real estimates of motion vectors, small magnitudes of gradients are dominated by noise so that even in the case of non-vanishing gradients, the computation of the inner product as a weight is increasingly ill-conditioned with a decreasing ratio

$$\lambda := \frac{\min(\|\gamma_u\|, \|\gamma_v\|)}{\max(\|\gamma_u\|, \|\gamma_v\|)} \tag{5.13}$$

of the horizontal and vertical filter responses' magnitudes.

In order to derive a modified weight that accounts for the ratio of different magnitudes $\|\gamma_u\|, \|\gamma_v\|$, a disocclusion perpendicular to its spatial edge is considered. Figure 5.12 shows an illustration.

Let the amplitude of the filter responses in this example be chosen such that, for the unrotated version, they may be computed as $\gamma_u = 1$, $\gamma_v = 0$. Consequently, a rotated disocclusion may be written as



(a) Horizontal disocclusion (b) Rotated disocclusion

Figure 5.12: Disocclusion: Horizontal and rotated version

$$\gamma_{u,\phi} = \cos(\phi) \cdot \exp(i\phi) \tag{5.14}$$
$$\gamma_{v,\phi} = \sin(\phi) \cdot \exp(i\phi) \tag{5.15}$$

in which the term $\exp(i\phi)$ describing the spatial orientation is scaled by the magnitude $\cos(\phi)$ of the horizontal and $\sin(\phi)$ of the vertical edge. In this case, the inner product (5.12) yields $a^*_{uv} = \frac{\cos\phi \sin\phi}{\|\cos\phi\| \|\sin\phi\|}$. Hence, the magnitude of the normalized inner product exists with $\|a^*_{uv}\| = 1$ if $\phi \neq n\pi/2, n \in \mathbb{Z}$.

For the following experiments, a modified version a_{uv} additionally accounts for the ratio of the horizontal and vertical magnitude λ by a linear combination of the normalized inner product and a constant:

$$a_{uv} = \lambda \frac{\left\langle \begin{pmatrix} \alpha_u \\ \beta_u \end{pmatrix}, \begin{pmatrix} \alpha_v \\ \beta_v \end{pmatrix} \right\rangle}{\|\gamma_u\| \, \|\gamma_v\|} + (1 - \lambda) \tag{5.16}$$

It is easy to verify that $\|a_{uv}\| = \|a^*_{uv}\| = 1$ and that a_{uv} may also continuously extended at $\|\gamma_v\| = 0$ or $\|\gamma_u\| = 0$ as the smaller magnitude $\min(\|\gamma_u\|, \|\gamma_v\|)$ in the denominator of (5.16) is canceled by the ratio λ of eq. (5.13). The remaining case in which both magnitudes vanish does not need to be considered.

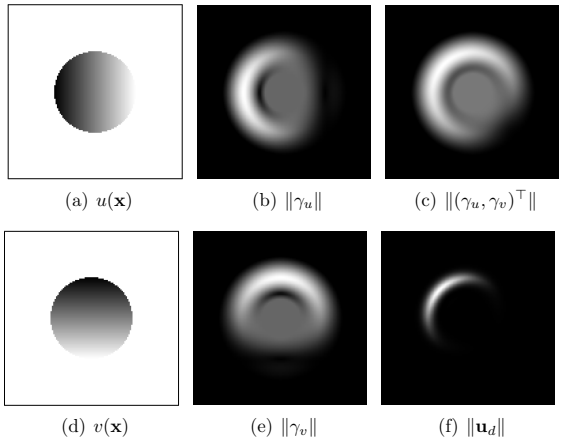
It has been mentioned in section 5.2.3 that even in the case of smooth motion fields, spatial orientations are identical if the corresponding differential homography is singular. Suppressing them would be necessary if the motion field can indeed be described by singular homographies, e. g. if regions in the scene degenerate to a line in the second view. In this work, degenerate smooth motion fields have not been considered.

The *direction of the motion change*, i. e. the orientation of an epipolar line $\mathbf{l}'$ (see equation 5.1), can be expressed by the filter responses' horizontal and vertical amplitude. With a_{uv} and consistent signs, it can be computed with

$$\mathbf{u}_d = a_{uv} \begin{pmatrix} \operatorname{sgn}(\alpha_u)\|\gamma_u\| \\ \operatorname{sgn}(\alpha_v)\|\gamma_v\| \end{pmatrix} \tag{5.17}$$

$\|\mathbf{u}_d\|$ represents the *amplitude* of a motion edge. It serves as a weight in the Hough transform as described in section 5.3.

Figure 5.13 shows coefficient images for the example motion field described in section 5.2.2. It can be seen that suppressing false motion edges with a_{uv} is necessary at the zooming foreground around the center of the image. The corresponding motion type can be found in table 5.1(5). Fig. 5.13(f) shows remaining filter reponses at the true foreground/background transition.



(a) $u(\mathbf{x})$ (b) $\|\gamma_u\|$ (c) $\|(\gamma_u, \gamma_v)^\top\|$

(d) $v(\mathbf{x})$ (e) $\|\gamma_v\|$ (f) $\|\mathbf{u}_d\|$

Figure 5.13: Workflow of filtering optical flow. Motion vector field: (a) horizontal component, (d) vertical component. Motion edge intensities: (b) horizontal component, (e) vertical component. (c),(f): Amplitude of coefficient vectors: (c) without supressing false motion edges, (f) with suppressing false motion edges. The intensity of each image has been normalized for visualization.

The result of the described filtering process is a parallax vector $\mathbf{u}_d = \left(u_{dx}, u_{dy}\right)^\top$ for each pixel $\mathbf{x}_E = \left(x, y\right)^\top$ that provides information about the corresponding epipolar line. This way, parameters for individual epipolar lines

$$\mathbf{l}'(\mathbf{x}) = \left(u_{dy}, -u_{dx}, y'u_{dx} - x'u_{dy}\right)^\top \tag{5.18}$$

can be associated with each pixel in an image.

5.3 Hough Transform

The previous section has shown how to extract parallax vectors $\{\mathbf{u}_d\}$ for each pixel by filtering the dense motion field. In this approach, an epipole is determined as the intersection

of epipolar lines using a Hough transform, similar to the experiments of section 3.2.5. In this approach, each accumulator cell corresponding to an epipolar line $\mathbf{l}_i$ is incremented by $\|\mathbf{u}_d\|_i$ (see equation (5.17)) in order to amplify large motion discontinuities.

In following experiments, the position and orientation of intersecting lines has been isotropically scaled and shifted prior to Hough transform such that the width and height of an input image does not exceed a horizontal and vertical field of view of 90°. As a result, the origin of the linearly transformed coordinate system coincides with the center of an image whereas points on the coordinate axes at the image's border can be found at $\vartheta = 45°$. This step is similar to prenormalization of point correspondences as described in section 2.6.2. A snapshot of a LAMBERT map can be seen in figure 5.14.

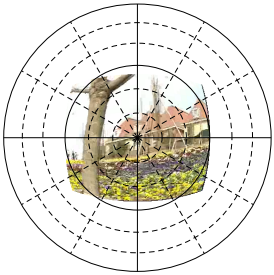


Figure 5.14: Prenormalized image coordinates for Hough transform

5.4 Limitations: Sensitivity to Noise

This section analyzes accuracy with respect to noisy correspondences. Zero motion is assumed with additive 2D Gaussian noise $\mathbf{u}(\mathbf{x}) = \mathcal{N}_2(\mathbf{0}, \sigma\mathtt{I})$. In this case, for the method to yield unbiased results, line orientations $\phi_l = \arctan(l_y/l_x)$ need to be identically distributed.

As filter kernels are antisymmetric and used for a convolution (5.9), which is a linear operation, filter responses γ_u, γ_v have zero mean as well. However, the direction of an epipolar line (5.17) is determined by their magnitudes $\|\gamma_u\|$ and $\|\gamma_v\|$. In this case, noise obeys a Rayleigh distribution [Pap65]

$$\mathrm{prob}(\|\gamma\|) = \frac{\|\gamma\| \exp\left(-\left(\frac{\|\gamma\|}{2\sigma}\right)^2\right)}{\sigma^2} \tag{5.19}$$

with the following properties:

$$E\{\|\gamma\|\} = \sigma\sqrt{\frac{\pi}{2}} \tag{5.20}$$

$$\sigma^2_{\|\gamma\|} = \left(2 - \frac{\pi}{2}\right)\sigma^2 \tag{5.21}$$

As horizontal and vertical noise is uncorrelated, the two-dimensional probability function can be given as

$$\text{prob}(\|\gamma_u\|, \|\gamma_v\|) = \frac{\|\gamma_u\| \exp\left(-\left(\frac{\|\gamma_u\|}{2\sigma}\right)^2\right)}{\sigma^2} \cdot \frac{\|\gamma_v\| \exp\left(-\left(\frac{\|\gamma_v\|}{2\sigma}\right)^2\right)}{\sigma^2} \tag{5.22}$$

Hence, a polar representation (r, ϕ) of the probability yields

$$\text{prob}(\|\gamma_u\|, \|\gamma_v\|) = \frac{\cos\phi \sin\phi}{\sigma^4} \exp\left(-\left(\frac{r}{2\sigma}\right)^2\right) r^3 \tag{5.23}$$

$$= \frac{\sin(2\phi)}{2\sigma^4} \exp\left(-\left(\frac{r}{2\sigma}\right)^2\right) r^3 \tag{5.24}$$

$$\text{prob}(r, \phi) = \frac{\sin(2\phi)}{2\sigma^4} \exp\left(-\left(\frac{r}{2\sigma}\right)^2\right) r^3 = \text{prob}(\phi)\,\text{prob}(r). \tag{5.25}$$

It can be seen that orientations are not identically distributed but that their probability follows $\sin(2\phi)$. Figure 5.15 shows a Hough map for the described situation. It can be seen that indeed filter responses yield epipolar lines which tend to be diagonally oriented.

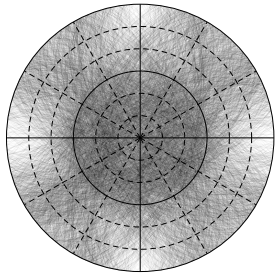


Figure 5.15: Hough map for zero motion in the presence of Gaussian noise

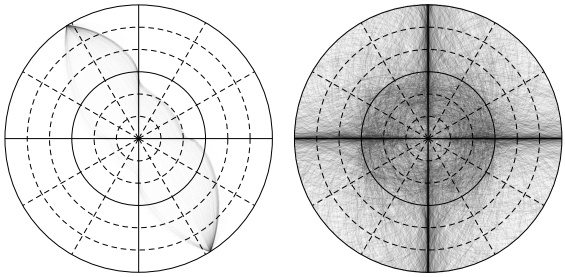
It is possible to find a function $\phi' = g(\phi)$ such that the probability density function of ϕ' is constant [Roh06]. For simplicity, the following equations are reduced to $0 < \phi < \frac{\pi}{2}$ and use a generic constant c:

$$\text{prob}(\phi)d\phi = \text{prob}(\phi')d\phi' \tag{5.26}$$

$$\Leftrightarrow \quad \text{prob}(\phi) = c\frac{d\phi'}{d\phi} = c\frac{dg(\phi)}{d\phi} \tag{5.27}$$

$$\Leftrightarrow \quad g(\phi) = c\int_0^{\phi} \sin(2\xi)d\xi = c\sin^2(\phi) \tag{5.28}$$

The effects of transforming polar coordinates on the Hough map are shown in figure 5.16. In this case, orientations are identically distributed, as desired. However, due to discretization, accumulator near the coordinate axes attract a high number of votes, similarly to the attracting equator in case of the ORTHO parametrization shown in section 3.2.5.



(a) (b)

Figure 5.16: Transforming coordinates of the Hough map in order to remove bias

It is obvious that the impact of discretization errors shown in figure 5.16 to the estimation of intersecting epipolar lines is more severe than the non-uniform distribution of figure 5.15. Therefore, the proposed methods have been evaluated in chapter 6 without accounting for the systematic bias introduced by Gaussian noise.

5.5 Implementation

In order to increase processing speed, an implementation of the proposed approach can benefit from two optimizations:

- The filter kernel is not separable. Hence, convolution (5.9) needs $\mathcal{O}(W \cdot H \cdot B^2)$ multiplications. However, the filter is odd-antisymmetric with respect to the origin and even-antisymmetric with respect to one of the two coordinate axes. If convolution takes advantage of the symmetries, the number of multiplications is still $\mathcal{O}(W \cdot H \cdot B^2)$ but can be reduced by a factor of 8.

- For the special case of $B = 3$, convolution is identical to computing a discrete gradient and can be done exclusively by the following equation:

$$\gamma_{u,v}(x, y) = \mathbf{u}_{x,y}(x + 1, y) - \mathbf{u}_{x,y}(x - 1, y) + i(\mathbf{u}_{x,y}(x, y + 1) - \mathbf{u}_{x,y}(x, y - 1))$$

5.6 Experiments

In an experiment, the accuracy and robustness of the proposed method ("MEH") with the normalized eight-point algorithm ("N8P") has been compared by using a synthetic, dense motion vector field idential to the evaluation in chapter 6. Gaussian noise has been added with a standard deviation σ from 0 to 6 pixels and the residual angular error Δe of an epipole has been analyzed. For the outlier-free case, results are shown in figure 5.17. It can be seen that the normalized eight-point algorithm clearly outperforms the proposed method even in the case of perfect correspondences. This is true for both used filter kernel sizes. However, a bigger kernel size D results in averaging over several pixels such that the effect of noise can be reduced.

In a second scenario, four rectangular areas of the optical flow have been replaced by outliers (see figure 6.2). MSAC has been used as a robust estimator for the normalized eight-point algorithm by setting a fixed number of $n = 500$ iterations. Figure 5.18 shows results for two different outlier ratios and a filter width of $D = 11$ pixels. In the case of outliers, the proposed method performs better at low noise levels.

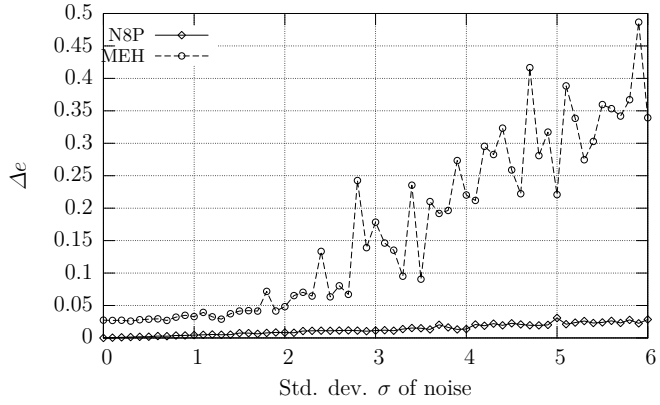
5.7 Conclusion

This chapter has introduced an algorithm for the parallax-based estimation of epipoles. This technique can be used as an initial independent step for relative pose estimation as shown in section 3.1.3. It is able to solve the primary, outer estimation problem, requires two images with mutual dense motion fields and uses a motion filter for detecting discontinuities. It could be shown analytically that identical orientations of the horizontal and vertical gradient cannot occur in continuous motion fields besides degenerated motion types and may therefore generally be used to locate foreground/background discontinuities.

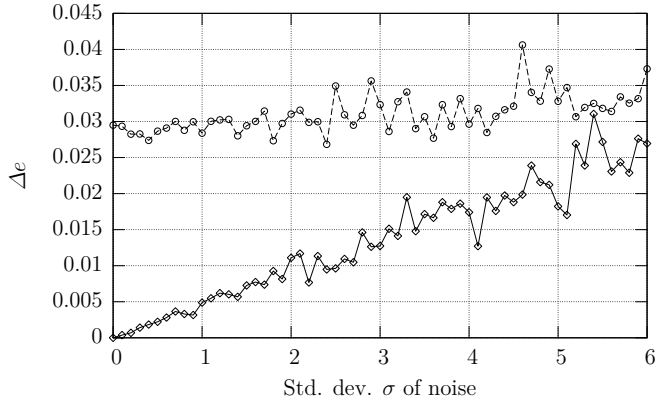
After postprocessing filter responses, epipolar line information, including different amplitudes, is available for each pixel. The epipole as the intersection of epipolar lines has been determined via Hough transform utilizing a suitable parametrization introduced in section 3.2. This way, a Gaussian lowpass filter may additionally be used for reducing the effects of noise.

It has been shown that in case of uncorrelated Gaussian noise, filter responses do not yield identically oriented epipolar lines. It is possible to postprocess polar coordinates on a Hough map in order to obtain identically distributed orientations, but this solution introduces severe artifacts caused by discretization.

Figure 5.19 shows a summary of the proposed technique for one epipole.

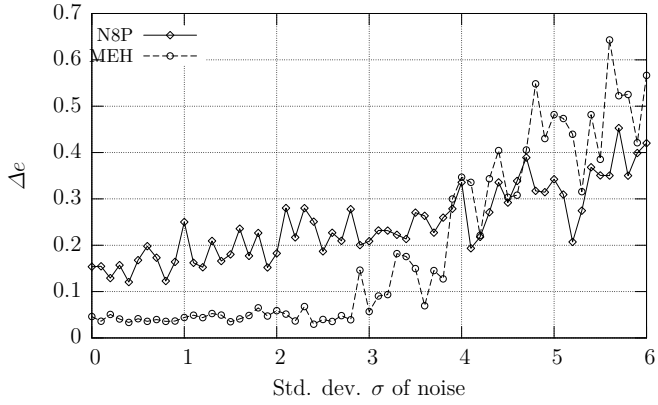


(a) $D = 3$ pixels

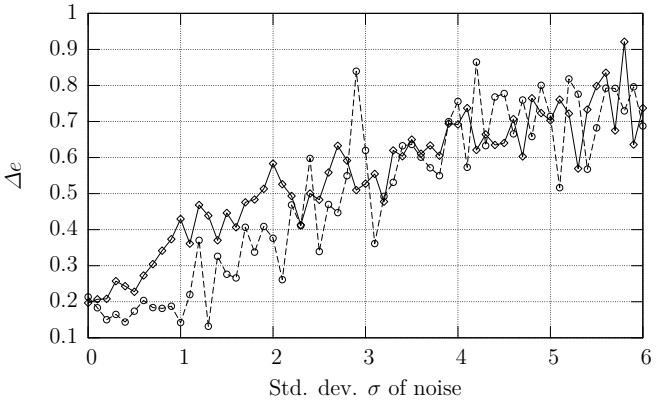


(b) $D = 11$ pixels

Figure 5.17: Residual angular error for different noise levels, no outliers



(a) 20 % outliers



(b) 50 % outliers

Figure 5.18: Robustness of epipole estimation against outliers

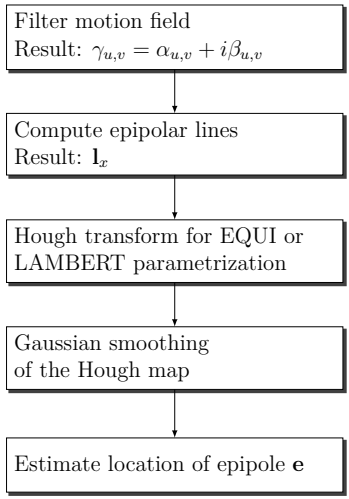


Figure 5.19: Workflow of parallax-based estimation of epipoles

Chapter 6

Evaluation

In this section, the two approaches for geometry-driven relative pose estimation are evaluated. In order to obtain quantitative results, real motion sequences with ground truth as well as synthetic ones have been used. For all experiments, the mean angular error (3.2) between each epipole and its true reference as well as the rotation error related to φ_E has been analyzed:

$$\Delta e = \frac{1}{3}\left(\Delta e_{\mathbf{e}} + \Delta e_{\mathbf{e}'} + \Delta e_{\varphi_E}\right) \tag{6.1}$$

6.1 Overview

6.1.1 Input data

Three motion types have been chosen:

- **FORWARD**: The dominant motion component between two successive frames is translation along the optical axis. Hence, epipoles are located near the image center. This type of motion can be found frequently in application scenarios like robot or automotive navigation in which the camera is oriented towards the direction of heading.

- **ROTATION**: The camera rotates around an object of interest. In this case, epipoles are finite but are not close to the optical axis. Not only can this configuration be found in case of a single moving camera but also for a stereo setup with converging cameras.

- **SIDEWAYS**: In case of synthetic data, a third motion type has been used: The camera translates sideways so that epipoles are located at infinity. This type of motion can also be found for the two cameras of a rectified stereo setup.

6.1.2 Evaluated methods

The two geometry-driven approaches have been compared with the robust normalized eight-point algorithm described in section 2.6.3. The following notations are used for illustrations:

- N8P: Normalized eight-point algorithm with MSAC, using the geometric error as cost function.
- N8P+PBM: Normalized eight-point algorithm with MSAC, using the geometric error as cost function. Subsequent refinement with pbM-based outer and inner optimization.
- PBM: Random sampling approach with pbM-based outer and inner optimization for each iteration.
- PBM+GLOBAL: Random sampling approach with pbM-based outer and inner optimization for each iteration. Additional global search for co-latitudes $\vartheta_{\hat{\mathbf{e}}}$, $\vartheta_{\hat{\mathbf{e}}'}$.
- MEH+PBM: Motion edges and Hough transform (dense approach, chapter 5) for initial values of epipoles $\hat{\mathbf{e}}$, $\hat{\mathbf{e}}'$. Subsequent refinement and estimation of the epipolar line homography with pbM-based outer and inner optimization.

6.1.3 Test system

Measurements have been done on a standard PC (Athlon XP 3200, 2 GB RAM). The used methods have been implemented in C++ (see also appendix A) using the gcc compiler version 4.0.2.

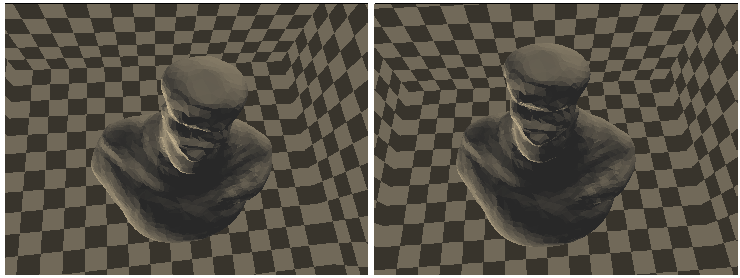
6.2 Synthetic Data

For the following experiments, synthetic data provided by a interactive OpenGL camera and an image of 320×240 pixels has been used. This way, a ground truth motion field has been available by using OpenGL's homographies $\mathtt{H}_{GL}$ and $\mathtt{H}'_{GL} \in \mathbb{P}^{3,3}$ and the z-buffer that stores depth information for a frame [WNDS99].

The point $\mathbf{x}'$ corresponding to $\mathbf{x} = \left(x, y, 1\right)^\top$ can be computed given the width W and height H of an image and the corresponding depth value $z \in [0, 1]$ of OpenGL's z-buffer as

$$\mathbf{x}' = \begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{bmatrix} W & 0 & 0 & W \\ 0 & -H & 0 & H \\ 0 & 0 & 0 & 2 \end{bmatrix} \mathtt{H}'_{GL}\mathtt{H}_{GL}^{-1} \begin{bmatrix} \frac{2}{W-1} & 0 & 0 & 0 \\ 0 & -\frac{2}{H-1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{6.2}$$

Snapshots of the synthetic scene are shown in figure 6.1.



(a) (b)

Figure 6.1: Snapshot of a synthetic OpenGL sequence: rotational motion

Two parameters $\{\sigma, r\}$ have been used for modifying the accuracy of motion vectors. Whereas σ denotes the standard deviation of additive Gaussian noise $\mathcal{N}_2(\mathbf{0}, \sigma\mathtt{I})$, r has been used to set the ratio of outliers in the motion field. In order to account for spatial coherence, outliers have been placed as four rectangular clusters as shown in figure 6.2.

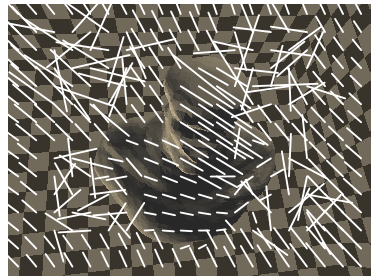


Figure 6.2: Example synthetic motion field containing 30% outliers.

In order for the methods to be comparable, the number of iterations has first been globally fixed to $n = 500$. For the normalized eight-point algorithm, point correspondences have been classified as outliers using an error threshold of $t = 1 \cdot 10^{-4}$.

It can be seen that the MEH approach yields the lowest error for both error levels regardless of the motion type. In the case of an outlier-free data set, the normalized eight-point algorithm performs best in the case of FORWARD. This is contrary to the case of ROTATION and SIDEWAYS, in which the residual error of the is lower. It can be explained by the fact that motion vector fields at discontinuities are more homogeneous for sideways than for forward motion.

Moreover, for a noise level of $\sigma = 1$ pixel, the effect of increasing the number of iterations for three different thresholds is shown in figure 6.4. It can be seen that parameters for the robust linear eight-point algorithm may be tuned to a given noise level. For the sparse method developed within this work, tuning is not necessary.

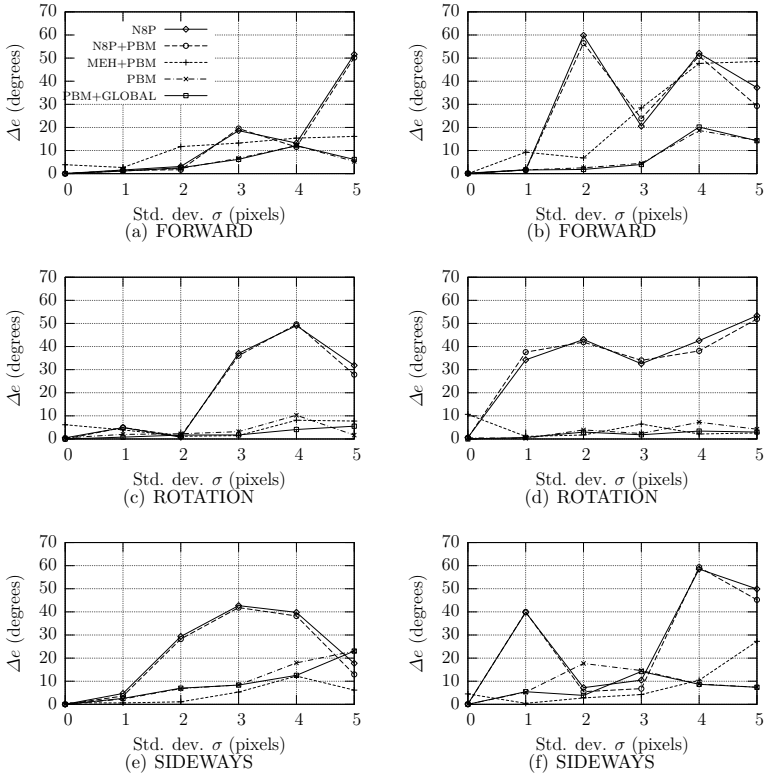
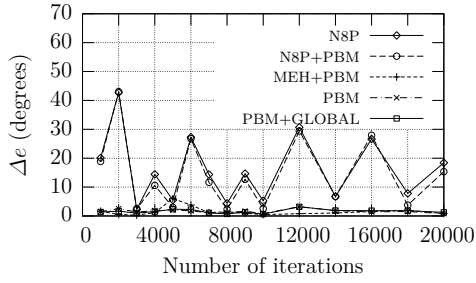
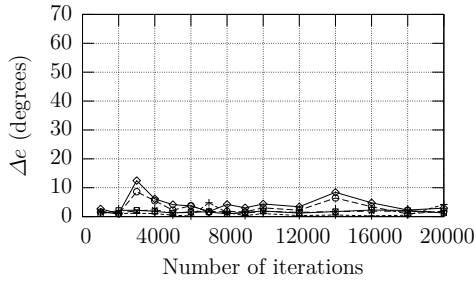


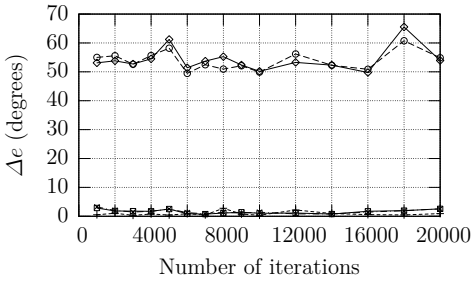
Figure 6.3: Mean Δe (6.1), $n = 500$, left(a,c,e): $r = 0$, right(b,d,f): $r = 0.4$. Abbreviations for evaluated methods are explained in subsection 6.1.2



(a) $t = 1 \cdot 10^{-6}$



(b) $t = 1 \cdot 10^{-4}$



(c) $t = 1 \cdot 10^{0}$

Figure 6.4: Mean Δe (6.1) for different numbers of iterations and thresholds. Abbreviations for evaluated methods are explained in subsection 6.1.2

6.3 Real data

Two real sequences including ground truth have been available [Cor]. They have been equipped with ground-truth information in terms of projection matrices for every frame. Hence, quantitative measurements could be made.

Stereo images of the following image sequences have been used:

- **CORRIDOR**: In this example, a camera moves along a corridor. This sequence corresponds to FORWARD motion.
- **HOUSE**: A house model on a textured surface rotates in front of the camera. In this example, the position of the camera is fixed while the foreground object is moving. Obviously, the motion type of this sequence is ROTATION.

A third sequence without ground truth information has been used for testing [Mid]. In this case, as the camera has been translating sideways between the analyzed frames, an essential matrix could be heuristically given as ground truth. Internal camera parameters have been approximated by assuming square pixels and an optical axis that intersects the image plane at its center.

- **TSUKUBA**: A camera translates sideways showing a shelf in the background as well as some foreground objects like a desk lamp and a head model. The motion type corresponds to SIDEWAYS.

Figures 6.5 and 6.6 show the two considered frames and additional sparse and dense motion vectors. They have been estimated using techniques mentioned in section 2.6.1, i. e. the HVDH block-matcher with Expanding Box search strategy and scanline motion predictors, as well as the feature-based algorithm for finding a sparse set of $N = 300$ correspondences by Shi and Tomasi [vdHWG06, ST94].

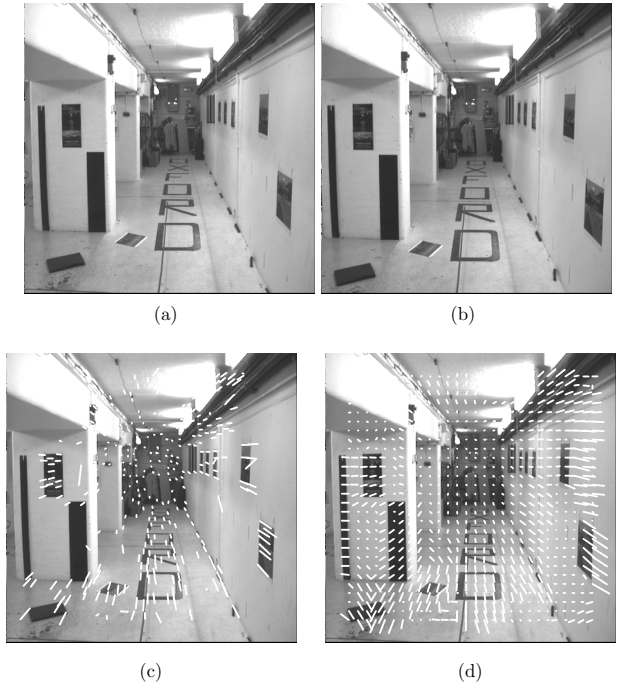
Figure 6.3 shows results for sparse relative pose estimation using a varying number of iterations. It is imporant to note the different ranges of the average angular error Δe.

It can be seen that in the case of CORRIDOR at a low ratio of outliers, all evaluated methods yield similar results: An average angular error $\Delta e < 1.8°$ could be achieved.

In the case of HOUSE, due to the high ratio of outliers, a large number of iterations is needed in order for the normalized eight-point algorithm to reduce Δe. However, it can be seen that a global search for their co-latitudes may significantly improve accuracy, due to the fact that epipoles are not located near the optical axis (see also section 3.1.2).

The two sequences are not optimal for parallax-based estimation of epipoles for the following reasons:

- In the case of CORRIDOR, there is no prominent foreground/background transition. Motion discontinuities can be found in figure 6.5(d) near the image border.



(a) (b)

(c) (d)

Figure 6.5: Two frames and found correspondences of the CORRIDOR sequence

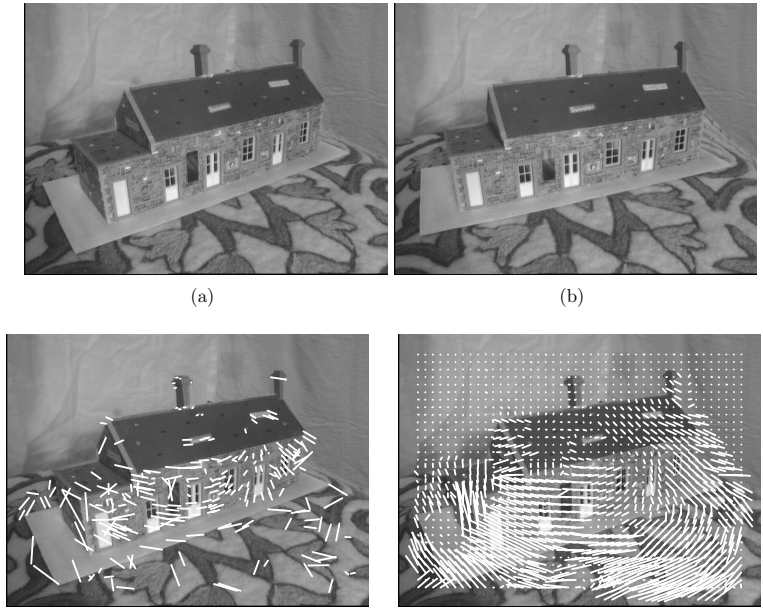
- In the case of HOUSE, the static background does not match the motion of the model. However, motion discontinuities can be found at the border between house and background. Also, due to rotation, the motion model for block-matching is violated so clusters of outliers providing motion discontinuities can also be found in the foreground area (see figure 6.6(d)).

The residual angular error Δe (6.1) has been computed as shown in table 6.1.

Sequence	Δe (degrees)
CORRIDOR	30.3
HOUSE	30.2

Table 6.1: Residual angular error Δe (6.1) for MEH

It can indeed be seen that the quality of MEH is inferior to sparse methods.



(a) (b)

(c) (d)

Figure 6.6: Two frames and found correspondences of the MODEL sequence

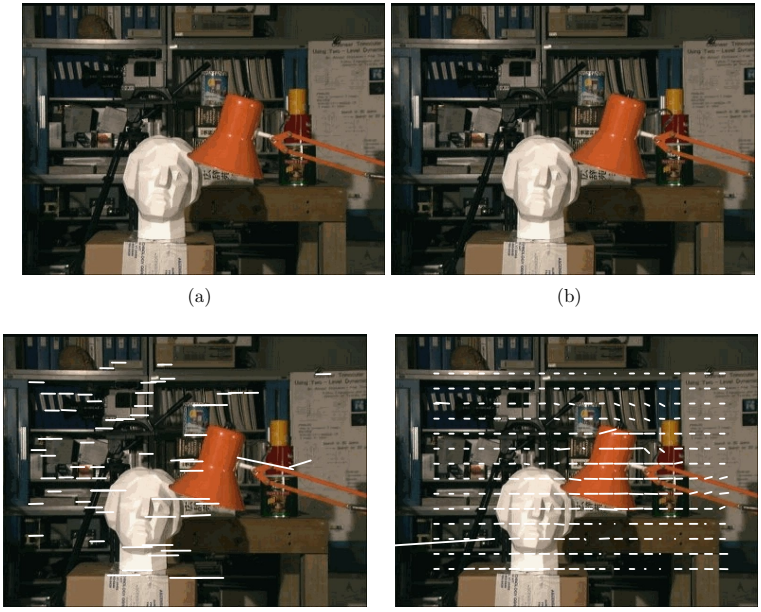
In case of TSUKUBA, requirements for parallax-based relative pose estimation are met: Parallax can be seen at the contours of the moving foreground objects. Furthermore, there is a single motion model for which a matching global epipolar geometry exists. Figure 6.8 shows the residual average angular error Δe.

As dense motion vectors do not contain many outliers, all analyzed methods perform well. However, it can be observed that global optimization (PBM+GLOBAL) may yield the lowest residual error of all analyzed methods if a high number of iterations is applied. However, in some cases, the quality of weak calibration could not be increased with respect to the normalized eight-point algorithm.

6.4 Conclusion

Synthetic and real experiments have demonstrated the performance of the methods developed for relative pose estimation within this work.

Even though the parallax-based approach does not need single, precise motion vectors



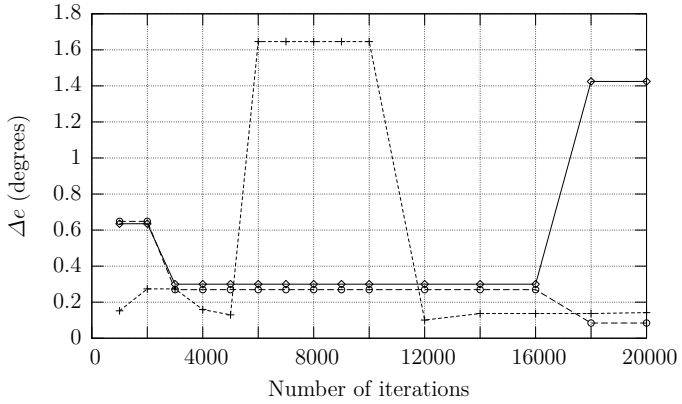
(a) (b)

(c) (d)

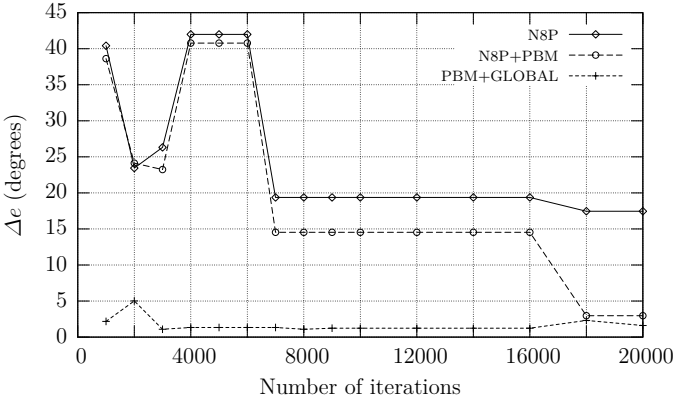
Figure 6.7: Two frames and found correspondences of the TSUKUBA sequence

at motion discontinuities and, while filtering the dense motion field, errors of several motion vectors are averaged due to convolution (see chapter 5), estimates of block-matching algorithms do not always yield the required accuracy for real motion sequences. Experiments using synthetic data have shown that the dense method for weak calibration is able to outperform other techniques if motion discontinuities can be estimated sufficiently accurate. These observations could be confirmed by the case of translational camera motion.

For both, synthetic as well as real data, high accuracy could be robustly achieved by using the PBM+GLOBAL technique. Using this estimation algorithm, a residual angular error below 10° could be obtained even in the case of high outlier ratios like HOUSE or synthetic data with $r = 40\%$. This method does not need additional parameters that have to be used for tuning and yields angular errors in a constant range regardless of the number of iterations in random sampling algorithms.



(a) Angular error (6.1) for CORRIDOR



(b) Angular error (6.1) for HOUSE

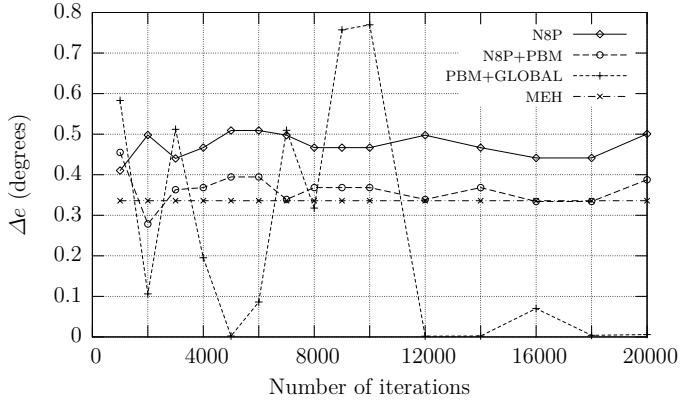


Figure 6.8: Angular error for TSUKUBA. Here, MEH has not been combined with PBM estimation, thus is not an iterative technique. Corresponding results are shown as a constant line for comparison. Abbreviations for evaluated methods are explained in subsection 6.1.2

Chapter 7

Conclusion

7.1 Summary

In this work, methods for robust relative pose estimation of two cameras by decomposing epipolar geometry have been developed. They are based on an orthonormal factorization of the fundamental and the essential matrix that relate two views of a static scene geometrically. Besides its descriptive and conceptual value, the factorization has been used as the basis for a minimal, symmetric parametrization. In case of the essential matrix, parameters directly describe geometric alignment of the two images. The parametrization has been related to the singular value decomposition (SVD) and, for the first time, to an explicit representation of the epipolar line homography.

The proposed factorization enables the task of relative pose estimation to be decomposed into an outer optimization problem, in which epipoles are determined, and an inner optimization problem for estimating the epipolar line homography. As the epipolar line homography is a rotation around the baseline in the case of known internal camera parameters, the inner optimization problem is univariate. Within the scope of this work, two possible approaches have been developed in which the proposed factorization could be applied. Whereas the first is based on recent research on projection-based M-estimation and operates with a sparse set of point correspondences, the latter requires a dense motion vector field and utilizes spatial coherence of motion vectors.

In chapter 4, the estimation of the epipolar line homography has been formulated in the context of kernel density estimation. In contrast to other, similar approaches, the cubic constraint of an essential matrix ensuring two identical and one vanishing singular value can be retained [CM02a]. Additionally, singular configurations arising from a polar representation have been avoided. A bandwidth estimation technique has been proposed to account for angular wrap-arounds, and an optional global search strategy for co-latitudes of the epipoles, as they tend to be particularly sensitive to noise in general camera setups.

Chapter 5 reviewed parallax-based methods for the direct estimation of epipoles as an alternative way for solving the outer optimization problem. The method of Rieger and Lawton as well as the improvements by Hildreth have been considered. Subsequently, a

technique for processing dense motion fields has been established that is similarly based on motion parallax but does not have conceptual drawbacks regarding the treatment of outliers, the accuracy of single motion vectors or restrictions of camera motion. The proposed approach is based on filtering a dense motion vector field and determines the location of epipoles by using a Hough transform. In order to detect motion parallax given a dense field of motion vectors, filter responses have been analyzed with respect to their spatial orientation. In this context, a property of local motion has been described: It could be shown that horizontal and vertical gradients of continuous motion vector fields may only yield identical orientations if they propagate to a degenerated space. This property has been used to discard filter responses at regions with continuous motion so that parallax information describing the orientation of epipolar lines at motion discontinuities can be gathered.

The Hough transform served as a robust estimation technique for intersecting epiolar lines. For this task, suitable minimal parametrizations of homogeneous points have been identified based on non-linear projections of a unit hemisphere that have been previously used in cartography. Experiments in section 3.2.5 have shown that azimuthal equidistant as well as Lambertian projection may be used as singular-free parametrizations that have advantageous properties for Hough maps.

Chapter 6 showed evaluation results for real data with ground truth as well as synthetic motion sequences. It could be demonstrated that geometry-driven estimation techniques for the relative pose of two cameras can be used to enhance the results of a random sampling algorithm. Experiments using real data showed that the quality of the dense motion field estimates used in this work is not sufficient for the parallax-based approach developed in this work. However, the proposed sparse technique is suitable as an intermediate step prior to miminizing an optimal objective function like the reprojection error.

In the presence of outliers, it has been demonstrated that the number of iterations is not crucial to final accuracy for the sparse geometry-driven estimation technique. Therefore, to some extent, answers to the questions posed in the introduction could be found.

7.2 Outlook

This section outlines possibilities for further research that are based on the methods which have been introduced in this work.

- This work assumes internal camera properties to be known. Consequently, properties like the focal length has to be fixed while analyzing subsequent images of a video sequence. It would be interesting to integrate ways for relaxed requirements, e. g. for varying focal length. Also, concepts for uncalibrated geometry-driven weak calibration have been described in this work. Hence, a possibility for further research is to set up a calibration system for the case of unknown camera parameters.

- In this work, analysis has been restricted to two images of a scene. However, the developed concepts can be extended to cover complete image sequences. In this

context, bundle adjustment may be approached equivalently as the estimation of all epipoles may serve as an outer estimation step whereas residual rotations around the z-axis in transformed space remains an inner optimization step.

- Finally, the proposed techniques may be limited to the purpose of identifying outliers in a set of given point correspondences, so that remaining inliers may be used for standard techniques, similar to the approach of Chen [CM03].

Appendix A

VistaLab

A.1 Motivation

In this work, new techniques for the estimation of two camera's relative pose have been introduced. Section 3.1.3 introduced two approaches which have been discussed in chapter 4 and 5 based on a factorization of the fundamental and essential matrix. The proposed factorization is a "modular" decomposition of epipolar geometry in the sense that relative pose estimation can be considered as consisting of two steps, both of which can be treated with different techniques.

From a technical point of view, a modular software framework helps develop different components for estimation algorithms and lets them connect during runtime in an easy way. For the scope of this work, a software framework in C++ has been developed. Modular implementation and evaluation of the introduced algorithms would not have been possible without it in an easy way. This section discusses design aspects and argues that they are advantageous for developing algorithms in other fields of signal processing as well.

Researchers and developers in the field of signal processing often consider software aspects as a side-effect. On the other hand, as implementing methods plays a central role in their domain, architectures and designs are an important issue.

Many solutions for image processing problems share a common pool of methods. Hence, *reuse* plays an important role and thus software libraries have evolved in recent years. A developer can benefit from a huge amount of existing code in order to solve a specific problem — on the other hand, existing software forces developers to adapt to a given environment that may, or may not, meet their needs. This is a problem especially for languages like C or C++ which are not flexible regarding the interoperability of data types [Str97]. On the other hand, both are still the languages of choice for the majority of real-time algorithms due to low level and portability.

Image acquisition and visualization is an important part in almost every application. Thus, other technical issues like the underlying operating system, GUI toolkits, or threading play a role. However, dealing with them does not contribute to solving an image processing problem.

Modular software frameworks are able to deal with some of these problems. Blocks for acquisition and visualization may be assembled with others in order to build complex algorithms. On the other hand, they often restrict the scope of applicability even more if they do not provide ways to integrate custom data structures.

This section describes *VistaLab*, a cross-platform plug-in-based software framework for image processing algorithms. Its design is different from existing frameworks for image processing as it provides a high degree of flexibility by minimizing the overhead for developing new modules. This way, existing software and custom data types can easily be integrated. In constrast to other software frameworks, *VistaLab* does not primarily focus on runtime aspects like parallel processing as it can be found in other publications on image processing software [Fra04]; it is targeted at the *development* and *integration* of new algorithms.

This appendix is organized as follows. The first part describes generic requirements of a software framework for image processing. In the following sections, solutions for each requirement are provided: After reintroducing the well-known "pipes & filters" architecture that can be found in many existing frameworks, a generic filter interface is introduced. Section A.4.2 demonstrates how such an interface can be implemented with almost no overhead. It will be shown how separating different aspects of code can be achieved with the same technique. Subsequently, *VistaLab* is described, a framework in which these techniques have been successfully applied. Finally, some sample applications will be presented and an outlook for possible extensions to other fields of signal processing will be given.

A.2 Requirements for Developing Image Processing Software

This section characterizes the process of developing algorithms in image processing. Each aspect leads to a requirement that frameworks for software development need to provide:

1. **Code reuse:** Techniques in the signal processing domain share a common pool of methods that may already be available.

⇒ **Modularity:** Modular software provides a high degree of reusability as modules, i. e. individual components that perform specific atomic tasks, are ready to be combined to build complex algorithms (section A.3).

2. **Functional focus:** The objective of researchers is to solve a specific problem in their domain. Secondary aspects, like interaction with users via graphical user interfaces (GUIs), do not gain high priority.

⇒ **Automation:** In order to focus on the functional core of an algorithm, other technical aspects have to be automated as much as possible. As an example, it is desirable to generate corresponding code for configuration dialogs automatically so that it does not have to be done by a developer (section A.4).

3. **Logging, tracing and visualization:** Newly implemented methods tend to contain errors. Whereas syntactic errors are recognized by the compiler or interpreter of the used programming language, many semantic errors can only be detected during runtime by analyzing results of a computation. Debug information is therefore needed until it has been made sure that an algorithm is implemented correctly. As a consequence, image processing algorithms contain different aspects of code — not only for debugging, but also for presenting results in terms of visualization. Code for different aspects is often interleaved, i. e. , a single aspect cannot be removed in a straightforward way.

⇒ **Separation of concerns:** Functionality for logging or debugging is mandatory when implementing methods. Moreover, visualizing results is one of the main reasons for using a graphical software framework. A suitable software design provides entry points that help developers bundle specific aspects of their code. This way, logging functionality can be removed easily as soon as it is no longer needed. Unit tests are another technique that can be used for verifying the correctness of an algorithm and should be supported by a software framework [JAHJ00]. Also, by separating the corresponding code within a module, it is possible to evaluate only relevant parts of an algorithm. Another advantage is that certain aspects can be switched on or off at runtime (section A.5).

4. **Uncertain requirements:** In the beginning of a development phase, aspects like the target operating system or final toolkits to be used are not fixed and are likely to change in the future, either because of limitations of a current setup or because of other issues, like modifications in a customer's specification.

⇒ **Platform independence:** In order not to be tied to a specific software platform, tools have to be chosen that are independent of

   - the underlying operating system
   - language extensions
   - external dependencies

   In the case of image processing, external dependencies often cannot be avoided, e. g. for visualization, but in this case they have to be chosen in a platform independent fashion (section A.6).

The next sections will show how these requirements can be met in practice.

A.3 Pipes & Filters

An established architecture for processing signals like speech or images is the "pipes & filters" design [BMRS96]. Besides its simplicity, pipes & filters directly imply a high degree

of modularity and are used in signal processing applications as well as software layers like Microsoft's *DirectShow* [Pes03]. Figure A.1 shows an illustration.

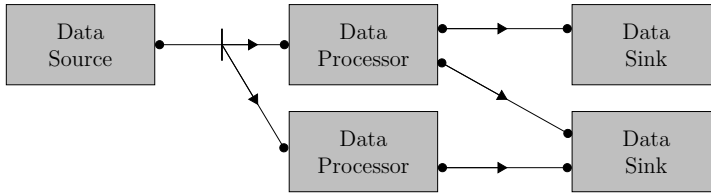


Figure A.1: The "pipes & filters" - design for signal processing

In this design, filters are atomic entities that receive data as input and provide processed data at their outputs. Pipes serve as data channels and can be interpreted as links between modules. A software framework based on this design enforces a standardized interface in order to be able to connect different modules. Moreover, modules can be implemented as plug-ins so that they can be easily distributed and changed even during runtime. Hence, in the following sections, the terms *filter*, *module* and *plug-in* will be used equivalently.

A.4 Generic Filter Interface

A close-up of an individual filter reveals two components that are usually visible externally from a software point of view (see figure A.2): Input- and output-pins. These pins identify *dynamic information* that changes during runtime. An input pin must provide information about the data type which the module is able to process. Similarly, an output pin has to describe the data type it produces. Both classes of pins must contain hooks for data transfer as well as identifiers. Hence, *meta-information* about incoming and outgoing data is needed.

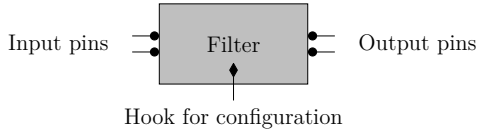


Figure A.2: Externally visible parts of a filter

Filter parameters define its state and usually remain constant while processing. In contrast to input/output data, parameters are *static*. Therefore, they cannot be modeled with the "pipes & filters" design as they have to be changed interactively. A common technique for changing parameters is to provide an entry point for custom configuration. In *VistaLab*, parameters are exposed in a similar way as pins, i. e. , each configurable parameter

has to provide an identifier as well as a hook for data transfer and the underlying data type. Hence, the software design of *VistaLab* does not distinguish between dynamic and static data of a module, it requires both, input/output data as well as parameters to be externally accessible information. An illustration can be seen in figure A.3.

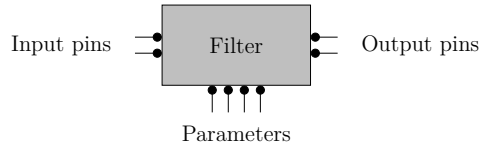


Figure A.3: Exposing configurable parameters and input/output-data of a filter in the same way.

By providing meta-information about externally visible data, it is possible for the framework to transfer data into and out of a filter — while processing or when changing parameters. This way, automatic construction of GUI elements for configuration is possible. Hence, a developer can *focus on the functional part* of a module, leaving technical tasks like building configuration dialogs to the framework.

A.4.1 Meta-information in C++: State of the Art

Technically, exposing internal data to an external interface means that a developer must register data in some way by conforming to the appliction interface (API) of the framework. This is a problem as some requirements introduced in section A.2 are not met: The API may change in future versions; also, a developer cannot focus on the functional core of a module. But as setting up a module's external interface cannot be avoided, corresponding code can be found in available signal processing frameworks, e. g. NeatVision [WM00] (see also section A.8).

Unlike Java, C# or smalltalk, the C++ language does not provide methods for exposing meta-information. Hence, the developer has to implement this functionality either by extending the C++ language or by designing classes according to a meta object framework. Several techniques have been proposed to integrate a meta object protocol into the C++ language [Chi95, GC96, IHS+96, CKW02]. However, this approach adds dependencies to the final application.

Meta object frameworks like Corba [HV00] or Microsoft's COM [Rog97] can be found more frequently in existing software, but they can be considered as additional dependencies as well. Finally, there are hybrid techniques like the Qt meta object compiler MOC which use a non C++-compliant code generator in order to adapt generic C++ classes to Qt's meta object model [Tro].

The approach of *VistaLab* is to automate the process of exposing data as much as possible by generative programming [CE00]. Pins and parameters are identified in the

source code at a high-level. The implementation for making them externally visible is achieved by means of a generator.

The C++ language provides two native code generators [CE00]:

1. The C++ template mechanism
2. The macro language of the C preprocessor

A major drawback of the C-preprocessor compared to C++-templates which is often mentioned in the literature is the fact that macros are not typesafe [Lib04]. On the other hand, macros may be used to a accomplish tasks that are cumbersome or impossible by using the C++-template mechanism alone [AG04]. For example, a single C++ template cannot simultaneously generate functions as well as variables without encapsulating them in specific object datastructures.

A.4.2 Automation

This section shows how to design a meta-information mechanism using a combination of the C preprocessor language and C++ templates. This way, source code does not need to contain C/C++-parts for exposing data to the framework at all. Moreover, the overhead for exposing data can be kept minimal. To demonstrate this technique, two examples are given: A configurable parameter of an integer type and an input pin for real-valued data:

```
// Declaring data to be exposed
INPUTPIN (double, m_inputData);
PARAMETER(int   , m_parameter);

// Adding Identifiers
CONFIGINPUTPIN  (m_inputData, "Input Value" );
CONFIGPARAMETER (m_parameter, "My Parameter");
```

It can be seen that declaring externally visible data items can be done with two macros. The first one is located inside the filter's declaration, the second one in its implementation. The description that is attached to each data with the second macro is also used as a run-time identifier or a tag in an XML project file, apart from e. g. displaying a label in a configuration dialog. No data structures of the framework have to be used so that changes in its interface do not require a module's code to be changed. Moreover, it is easy to use modules for different applications by simply changing macros. This is important if software frameworks serve as an evaluation platform only. Data type information and references to memory are automatically generated by using C++ runtime information (RTTI, [Str97]) so that custom data types can be passed through the framework. This approach heavily uses techniques of the C/C++ programming language while still being standard-conforming, as opposed to similar techniques, like Qt's MOC [Tro]. The introduced ideas can be transfered to other languages that support generative techniques as well.

A.5 Separating different Aspects of Code

Exposing a module's internals is not limited to data members of a module. In this section, an extension of the proposed scheme is presented: It is possible to separate different aspects of code by applying the same technique to member *functions*, also denoted as *custom hooks* of a module. Figure A.4 shows an illustration.

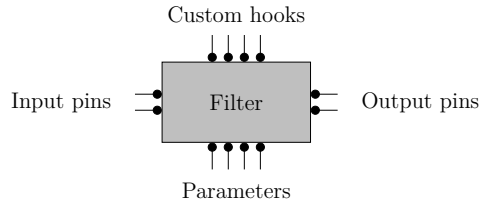


Figure A.4: Externally visible parts of a filter including function- and data pins

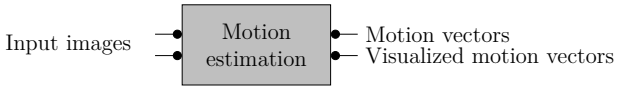
Possible hooks range from visualization to logging. Triggering them can either be done by the framework or by other modules. Hence, this extension allows the construction of a network of communicating modules.

An example is presented in figure A.5. It demonstrates a filter estimating *motion vectors* between subsequent images of a video sequence. The filter needs two images as input and provides a motion vector field at its output. In order to validate the results by inspection, the estimated motion vectors have to be rendered onto the first input image for visualization.

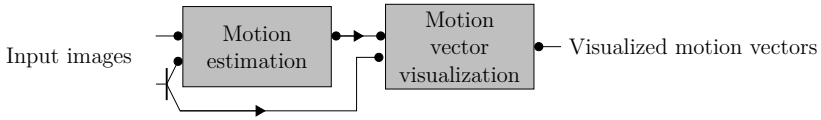
Three possible designs are considered. The first possibility, figure A.5(a), shows a filter which performs both estimating motion vectors and rendering. This choice is not optimal: Different aspects like motion estimation and visualization are not separated. Thus, rendering takes place even though if it may not be needed in a final application. The second design, fig. A.5(b), requires a custom module with the only purpose of visualizing motion vectors. Even though the two aspects are clearly separated now, there is still room for improvements: The additional module is useless without a motion estimator. In this sense, separation of the two aspects has gone too far. A better solution can be found with function pins as shown in figure A.5(c): A general purpose module for visualization, denoted as "image renderer", is connected to the rendering hook of a motion estimator. Hence, the auxiliary aspect of visualizing motion vectors can still be found in the filter performing motion estimation, but it is separated from the module's functional core. As a side-effect, the image renderer of figure A.5(c) can connect to different rendering pins at the same time so that results of different filters may be visualized in one image.

By means of the technique introduced in the last section, exposing a member function of a module can be achieved with a single macro:

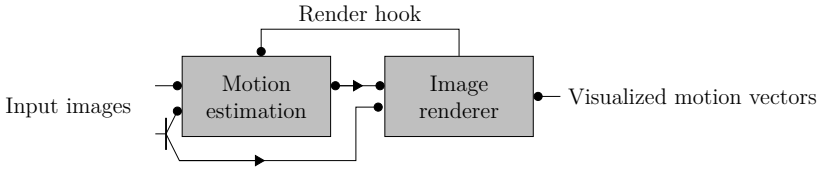
```
// Exposing member function "renderVectors()"
```



(a) Combined estimation and visualization in one filter



(b) Custom visualization module



(c) Render module that triggers visualizaton in the motion estimator

Figure A.5: Example of function pins: Three possible designs for visualizing motion vectors

```
CONFIGRENDERPIN(renderVectors, "Render Motion Vectors")
```

A.6 VistaLab: A Framework for Image Processing

This section describes more details of *VistaLab*, in which the aspects introduced above have been successfully implemented [WG05a, WG05b]. A platform independent toolkit (wxWidgets, [SH05]) has been chosen so that it is possible to develop and evaluate algorithms on commonly used operating systems. As no language extensions are required, VistaLab can be used in conjunction with any standard compliant C++-compiler. It is even suitable for cross-compilation so that the development system can differ from target systems. This way, the requirements of section A.2 could be met.

Configurable parameters are exposed to the framework by the macros introduced in section A.4.2. They are used by the framework for assembling configuration dialogs with

matching GUI controls. Also, parameters are displayed in a graphical configuration tree and used when saving current setups to a project file. Additionally, the mechanism described in section A.5 is used for separating aspects like visualization, logging, and unittests.

A.6.1 Decoupling data types from GUI via traits

There is another feature of *VistaLab* that may be used optionally, as it violates the independence of a module's code from data types of the framework. However, in some cases, configurable parameters of the same data type may have different purposes. As an example, a `string` parameter may either be used in a generic way, but it may also denote a file name, for which special GUI elements in a configuration dialog are desirable.

VistaLab offers an additional layer which is able to decouple data types from their purpose via traits [Mye95]. Traits provides a technique for mapping data types to data types via specialized templates. As an example, a pseudo data type `vistaLab::fileName` may be used for generating a string member and a special identifier denoting its purpose via a macro

```
// Denoting a parameter's purpose using traits
PARAMETER(vistaLab::fileName, m_fileName)
```

In this example, the member variable `m_fileName` is of type `string` if the `PARAMETER` macro uses traits like

```
#define PARAMETER(a,b) \
    vistaLab::configTraitsData<a>::dataType b;

template<> struct configTraitsData< fileName > {
    enum { configType = configItem::FILENAME };
    typedef std::string dataType;
};
```

A.6.2 Memory Management, Buffering, Threads

In *VistaLab*, dynamic data is passed from filter to filter via a common memory block, a "slice". In each slice, pointers to output data are stored, so duplicating dynamic data can be avoided. For the setup shown in figure A.1, a slice would consist of four entries pointing to each data item provided by the filters' output pins. Besides effects related to memory management, slices enable convenient evaluation of algorithms at high frame rates: A global ringbuffer stores a user-defined number of slices during processing. Users can review key frames later when processing in real time by stepping back in the ringbuffer. Also, slices are deleted automatically as soon as they are overwritten by a new one. Thus, dynamic data does not have to be deleted manually.

When reading data from cameras or other hardware input devices, global wait states can be avoided by using threads. However, they do not comply with the second and third

requirement of section A.2: Since threads depend on the operating system and do not fall into the functional domain of a module, they should be completely moved to the framework. When implementing a module in *VistaLab*, it can be declared as a an input thread. While processing, it will be operated by an own wrapper thread of the framework.

VistaLab uses image data types and algorithms of the *Vigra* image processing library [Kö00].

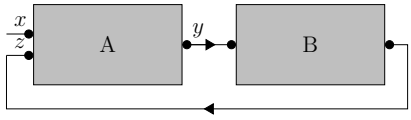
As threads do not run synchronously, a synchronization mechanism is used that triggers connected modules as soon as all threads have delivered output data. As a consequence, processing speed is determined by the thread running at the lowest frame rate.

A.6.3 Sample applications

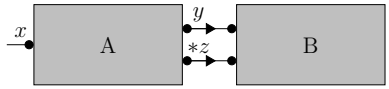
In this section, three setups are introduced to demonstrate possibilites of *VistaLab*. The first example shows a realization of feedback loops. Subsequently, a stereo computer vision application and a setup for real-time ego-motion estimation is described.

Feedback loops

The "pipes and filter" design implies a sequential connection of modules so that circular dependencies like feedback loops cannot be modeled in a straightforward way [Fra04]. However, they can be simulated by using references as data-type between modules. Figure A.6 illustrates how they can be realized with *VistaLab*. Module "A" gets feedback "z" from module "B" as "B" is able to access the reference to "z" provided by "A".



(a) Generic feedback loop



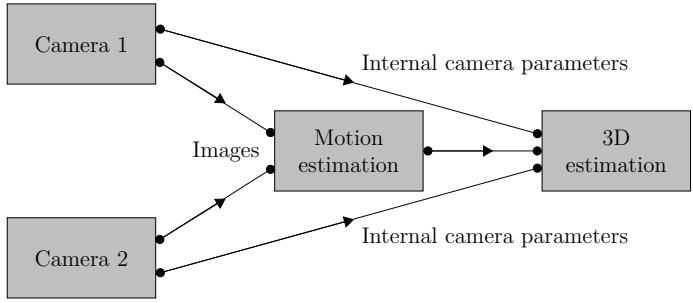
(b) Realization without circular connections.
∗ indicates a reference

Figure A.6: Original and simulated implementation of feedback loops.

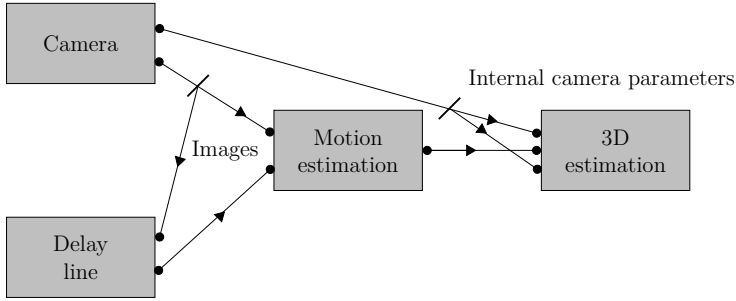
Stereo Computer Vision

A second example shows how the design can be utilized for stereo computer vision. Here, two or more images of a scene are required. They may originate from different cameras or be subsequently captured by a single moving one.

Figure A.7 demonstrates two possible module configurations. A.7(a) shows a stereo setup with two cameras that run in parallel. As soon as both images have been captured, motion vectors between the first and second image are computed so that 3D calculations can follow. A.7(b) illustrates that the same methods can be used in case of a single video stream: A delay line serves as an image buffer so that motion can be estimated between different snapshots of a scene.



(a) Two images of a scene taken simultaneously by two different cameras



(b) Two images taken by one camera at different times

Figure A.7: Two variants of a 3D computer vision scenario

Real-time camera ego-motion

A final example shows a demonstrator system for camera ego-motion estimation in real-time [SGG04a, SGG04b]. Its setup is based on the previous example, but uses a combination of a corner detector and a variable number of template matchers to estimate dominant 2D motion in the image. An illustration is shown in figure A.8.

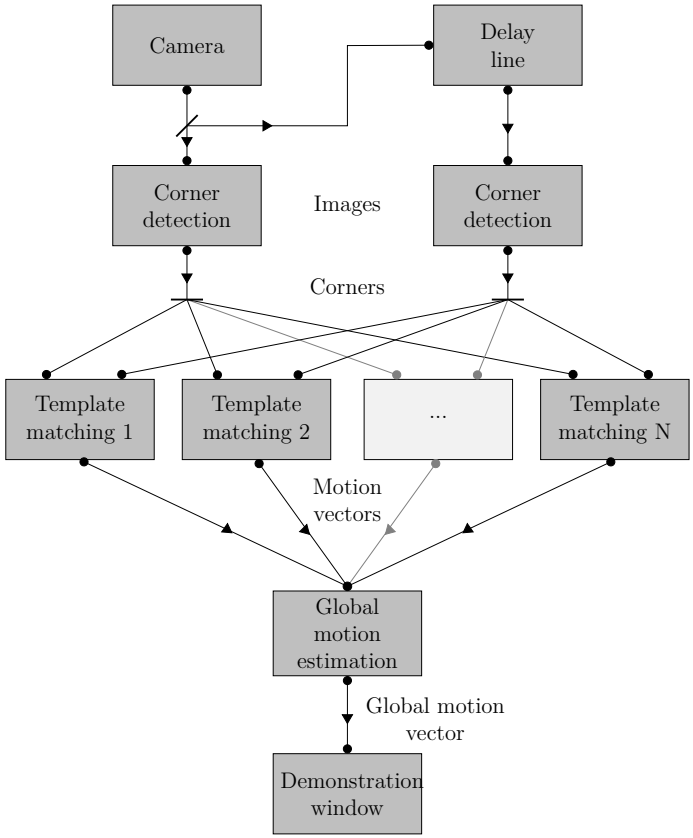


Figure A.8: Setup of an application for estimating real-time camera motion

The system is targeted at an optical remote control for interactive television: The dominant motion is used to navigate through a menu structure. Therefore, the demonstrator contains a module simulating a TV-screen.

It is obvious that the demo window depends on the used GUI toolkit. However, as it is implemented as a filter, exchanging it with a different module that communicates with a TV-set in a final application is easy.

A.7 Future Extensions

Even though developing image processing methods is the main objective of this framework, it may be adapted to other signal processing fields as well: The pipes & filters design is not limited to transferring images (see also section A.6.3). In fact, due to the generic interface, it is possible to integrate other data like audio. The only limitation of *VistaLab* is the fact that modules cannot operate at different frequencies. However, the paradigms of *VistaLab* can be combined with others like flow scheduling [Fra04].

A.8 Related work

Modular GUI-based frameworks for image processing have also been developed in the past. This section reviews three other plug-in-based cross-platform frameworks.

A.8.1 ImageJ

Even though being written in Java, ImageJ has similar design principles with respect to *VistaLab* [AMR04]. It also constructs image processing algorithms by connecting plug-ins that can be provided externally. However, plug-ins are either tightly coupled to the underlying image data structures, e. g. when acting as a filter, or they can only process string arguments. Like *VistaLab*, non-constant parameters may be modified during run-time by the user in a configuration dialog. In ImageJ, the developer is responsible for the creation of GUI elements, and for transferring data into and out of them. There is no automatic mechanism based on the detection of a parameter's data type like in *VistaLab*.

A.8.2 NeatVision

The second framework considered in this section, NeatVision, is also implemented in Java [WM00]. Plug-ins provide input and output pins by virtue of a data container. However, its interface is also predefined by the framework so that no custom data types may be used by NeatVision. Similarly to ImageJ, configuration dialogs for parameters have to be provided by the developer.

A.8.3 MeVisLab

In contrast to the two previous applications, MeVisLab is written in C++ [Gmb]. Similarly to *VistaLab*, plug-ins may be used as processing blocks and the C preprocessor language is used as code generator. Generic MeVisLab plug-ins may only process a predefined image

data type. Configurable parameters may be exposed to the framework manually. In this case, no code generators are used so that the developer has to use function calls and data structures provided by the main application. Qt, a cross-platform C++ toolkit, is used in order not to be limited to a single operating system.

A.8.4 Discussion

In all considered existing image processing frameworks, image data types are coupled to the main application tightly in the sense that their data types can directly be found in the application's programming interface (API). Other data types may not be used at all or are limited to a predefined set. Configurable parameters are not exposed to the framework except for MeVisLab. In this case, exposing them has to be done by the developer in C++ so that dependencies of the a plug-in's source code to MeVisLab are introduced.

A.9 Summary

This section has shown requirements and solutions for software frameworks in the field of image processing in order to develop algorithms with minimal overhead. The proposed approach is based on a generic interface for processing modules and generative programming. This design has two consequences: Firstly, developers are not required to gain knowledge of foreign domains like GUI programming. Secondly, the source code for new methods can be kept in a future-proof way. Moreover, the introduced scheme can be used to separate different aspects of code. It has been shown how such an approach can be realized in C/C++ without using predefined data types or extending the C++-language. Finally, design aspects related to memory management and threading have been described.

Appendix B

Singular Value Decomposition

B.1 Introduction and Properties

A singular value decomposition (SVD) of an arbitrary matrix $\mathtt{A} \in \mathbb{R}^{n,m}$ has the following form:

$$\mathtt{A} = \mathtt{U}\Sigma\mathtt{V}^\top \tag{B.1}$$

In this equation, orthogonal matrices $\mathtt{U}$ and $\mathtt{V}^\top$ contain the eigenvectors of the symmetric matrices $\mathtt{A}\mathtt{A}^\top$ and $\mathtt{A}^\top\mathtt{A}$ respectively. The non-vanishing elements $\sigma_1, \sigma_2, \ldots \sigma_q$ of the diagonal matrix Σ are called *singular values* and are sorted in descending order $\sigma_i \geq \sigma_{i+1}$. The rank of $\mathtt{A}$ equals q. Hence, a square matrix $\mathtt{A} \in \mathbb{R}^{n,n}$ is singular if $q < n$. In this case, as the lower right diagonal element of Σ is zero, the null-space $\mathbf{h}$ of a singular matrix $\mathtt{A}$ satisfying $\mathtt{A}\mathbf{h} = \mathbf{0}$ can be found as the last column $\mathbf{v}_3$ of $\mathtt{V} = \begin{bmatrix}\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\end{bmatrix}$ due to

$$\mathtt{A}\mathbf{h} = \mathtt{U}\Sigma\mathtt{V}^\top\mathbf{h} = \mathtt{U}\Sigma\begin{bmatrix}\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\end{bmatrix}^\top\mathbf{v}_3 = \mathtt{U}\Sigma\begin{pmatrix}0\\ \vdots \\ 1\end{pmatrix} = \mathtt{U}\mathbf{0} = \mathbf{0}$$

The condition κ_A of a matrix using the spectral norm $\|\mathtt{A}\|_2$ is defined as the ratio of the smallest and the largest non-vanishing singular value $\kappa_A = \sigma_1/\sigma_q$. It describes the propagation of errors $\Delta\mathbf{x}$ in a linear transformation $\mathbf{x}' = \mathtt{A}\mathbf{x}$ [MV93]. A large condition number amplifies $\Delta\mathbf{x}$. Large condition numbers may be caused by different scales of coordinates. In projective geometry, this situation occurs frequently, e. g. when describing images with pixel coordinates ranging from 0 to 1024 or more. In this case, the x and y component of a homogeneous coordinate $\mathbf{x} = \begin{pmatrix}x, y, w\end{pmatrix}^\top \in \mathbb{P}^2$ cover three magnitudes, whereas the projective component is fixed at $w = 1$.

This is the reason for the prenormalization step in weak calibration (see section 2.6.2), after which corresponding scales of homogeneous coordinates are equalized [Har97],[CBvdHG03].

B.2 Algorithms

Algorithms to derive a singular value decomposition can be found in many books on numerical methods [GvL96, PFTV86] and are based on QR-decomposition. Section 2.4 particularly introduced the SVD of an essential matrix $\mathtt{E}$ in order to derive external camera parameters. In this case, the singular value decomposition may be derived with a simpler approach mentioned in [Nis04] and is listed in algorithm 5.

Algorithm 5 Computing the SVD of an essential matrix

1. Determine essential matrix $\mathtt{E} = \left[\mathbf{e}_a, \mathbf{e}_b, \mathbf{e}_c\right]^\top$
2. Compute cross products $\mathbf{e}_{ab} = \mathbf{e}_a \times \mathbf{e}_b$, $\mathbf{e}_{ac} = \mathbf{e}_a \times \mathbf{e}_c$, $\mathbf{e}_{bc} = \mathbf{e}_b \times \mathbf{e}_c$
3. Pick the cross product with the largest magnitude. Without loss of generality, let this vector be $\mathbf{e}_{ab}$.
4. Define $\mathbf{v}_c = \mathbf{e}_{ab}/\|\mathbf{e}_{ab}\|$, $\mathbf{v}_a = \mathbf{e}_a/\|\mathbf{e}_a\|$, $\mathbf{v}_b = \mathbf{v}_c \times \mathbf{v}_a$
5. Define $\mathbf{u} = \mathtt{E}\mathbf{v}_a/\|\mathtt{E}\mathbf{v}_a\|$, $\mathbf{u}_b = \mathtt{E}\mathbf{v}_b/\|\mathtt{E}\mathbf{v}_b\|$, $\mathbf{u}_c = \mathbf{u}_a \times \mathbf{u}_b$
6. The SVD is given by $\mathtt{E} \sim \left[\mathbf{u}_a, \mathbf{u}_b, \mathbf{u}_c\right] \mathrm{diag}(1, 1, 0) \left[\mathbf{v}_a, \mathbf{v}_b, \mathbf{v}_c\right]^\top$

Appendix C

Orthogonal Matrices

This work uses properties of orthogonal matrices, especially of those in $\mathbb{R}^3$: The orientation between two cameras can be described by a rotation matrix $\mathtt{R} \in \mathbb{R}^{3,3}$ (see chapter 2.5). Moreover, when decomposing the fundamental or the essential matrix as described in section 3.3.3, orthogonal Householder transformations $\mathtt{H}_H \in \mathbb{R}^{3,3}$ are used.

The set of all matrices $\mathtt{A} \in \mathbb{R}^{n,n}, |\det(\mathtt{A})| = 1$ form the *orthogonal group* $O(n)$. In this case, the following properties hold [MV93]:

- $\langle \mathbf{a}_i, \mathbf{a}_j \rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$ for two rows or columns $\mathbf{a}_i$ and $\mathbf{a}_j$
- The singular value decomposition yields $\mathtt{\Sigma} = \mathtt{I}$
- $\mathtt{A}\mathtt{A}^\top = \mathtt{I}$

As a consequence, metric units like distances or errors are retained after a transformation $\mathbf{x}' = \mathtt{A}\mathbf{x}$.

All matrices with $\det(\mathtt{A}) = 1$ build a subgroup, the *special orthogonal group* $SO(n)$. In 3-dimensional space, it corresponds to the group of rotation matrices $\mathtt{R}$. Hence, $O(3) \backslash SO(3)$ contains all matrices $\mathtt{A}$ with $\det(\mathtt{A}) = -1$. They can be related to reflections of 3D space.

It is easy to verify that reflections or rotations do not change the singular values of a matrix $\mathtt{A}$ as the product of two matrices of the orthogonal group remains an orthogonal matrix:

$$\mathtt{A} = \mathtt{U}\mathtt{\Sigma}\mathtt{V}^\top \quad \Leftrightarrow \quad \mathtt{A}\mathtt{R} = \mathtt{U}\mathtt{\Sigma}\mathtt{V}^\top\mathtt{R} = \mathtt{U}\mathtt{\Sigma}\mathtt{V}'^\top \quad \text{, with} \quad \mathtt{V}'^\top = \mathtt{V}^\top\mathtt{R} \tag{C.1}$$

It is interesting to note that the negation of all coordinate axes is a rotation in $\mathbb{R}^2$ and a reflection in $\mathbb{R}^3$. Hence, in 2D, a negative rotation matrix $-\mathtt{R}_2$ remains a rotation matrix, whereas the same construct $-\mathtt{R}_3 \in \mathbb{R}^{3,3}$ changes a rotation into a reflection matrix. This aspect is important when analyzing the SVD of a fundamental matrix for Householder-based decomposition in section 3.3.3. In order to relate the presented approach to the SVD, it has to be ensured that $\mathtt{U}$ and $\mathtt{V}$ represent rotations. This can be done by switching

their overall signs. As shown in section 3.3.3, for matrices $\mathtt{R}_2 \in \mathbb{R}^{2,2}$, sign changes have to be restricted to a single column/row only.

Bibliography

[AG04] D. Abrahams and A. Gurtovoy. *C++ Template Metaprogramming*. Addison-Wesley, 2004.

[AJ05] A.Bruhn and J.Weickert. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *Proc. International Conference on Computer Vision (ICCV)*, volume 1, pages 749–755, Beijing, People's Republic of China, 2005.

[AMR04] M.D. Abramoff, P.J. Magelhaes, and S.J. Ram. Image processing with imagej. *Biophotonics International*, 11(7):36–42, 2004.

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 25–36, Prague, Czech Republic, 2004.

[BFBB92] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt. Performance of optical flow techniques. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 236–242, 1992.

[BM95] B.-S. Boufama and R. Mohr. Epipole and fundamental matrix estimation using virtual parallax. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1030–1036, 1995.

[BMRS96] F. Buschmann, R. Meunier, H. Rohnert, and P. Sommerlad. *Pattern-Oriented Sofware Archicture*, volume 1 — A System of Patterns. Wiley, 1996.

[BS04] A. Bartoli and P. Sturm. Nonlinear estimation of the fundamental matrix with minimal parameters. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 26(4):426–432, 2004.

[BSS93] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley, 2. edition, 1993.

[CBvdHG03] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley. Revisiting hartley's normalized eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 25(9):1172–1177, 2003.

[CE00] K. Czarnecki and U. W. Eisenecker. *Generative Programming.* Addison-Wesley, 2000.

[Che95] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 17(8):790–799, 1995.

[Chi95] S. Chiba. A metaobject protocol for c++. In *Conference Proceedings of Object-Oriented Programming Systems, Languages and Applications*, pages 285—-299. ACM Press, October 1995.

[CKW02] T.-R. Chuang, Y.S. Kuo, and C.-M. Wang. Non-intrusive object introspection in C++. *Software - Practice and Experience*, 32(2):191–207, 2002.

[CM02a] H. Chen and P. Meer. Robust computer vision through kernel density estimation. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 236–250, 2002.

[CM02b] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 24:603–619, 2002.

[CM03] H. Chen and P. Meer. Robust regression with projection based m-estimators. In *Proc. International Conference on Computer Vision (ICCV), Nice, France*, pages 878–885, 2003.

[Cor] University of oxford, visual geometry group, http://www.robots.ox.ac.uk/ vgg/data1.html.

[DBF98] R. Deriche, C. Bouvin, and O. Faugeras. Front propagation and level-set approach for geodesic active stereovision. In *Proc. of the Asian Conference on Computer Vision (ACCV)*, volume 1, pages 640–647, 1998.

[dH93] G. de Haan. True-motion estimation with 3-d recursive search block matching. *IEEE Transactions On Circuits And System for Video Technology*, 3(5):368–379, 1993.

[Fau93] O. Faugeras. *Three-Dimensional Computer Vision - A Geometric ViewPoint.* MIT Press, 1993.

[FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[FBYJ00] D. Fleet, M. Black, Y. Yacoob, and A. Jepson. Design and use of linear models for image motion analysis. *International Journal of Computer Vision*, 36(3):171–193, 2000.

[FCSK02] C. Fehn, E. Cooke, O. Schreer, and P. Kauff. 3d analysis and image-based rendering for immersive tv applications. *Signal Processing: Image Communication*, 17(9):705–715, October 2002.

[FH75] K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function. *IEEE Transactions on Information Theory*, 21:32–40, 1975.

[FL01] O. Faugeras and Q-T Luong. *The Geometry of Multiple Images*. MIT Press, Cambridge, 2001.

[FLS92] O. Faugeras, Q.-T. Luong, and J. Maybank S.˙Camera self-calibration: Theory and experiments. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 321–334, 1992.

[Fra04] A.R.J François. *Emerging Topics in Computer Vision*, chapter Software Architecture for Computer Vision, pages 586–654. Prentice Hall, 2004.

[GC96] B. Gowing and V. Cahill. Meta-object protocols for c++: The iguana approach. In *Proceedings of the Reflection '96 Conference*, pages 137—-152, April 1996.

[GD01] C. Geyer and K. Daniilidis. Catadioptric projective geometry. *International Journal of Computer Vision*, 45(3):223–243, 2001.

[Gmb] MeVis GmbH. Mevislab.

[Gra98] F. S. Grassia. Practical parameterization of rotations using the exponential map. *The Journal of Graphics Tools*, 3(3):29–48, 1998.

[GvL96] G. H. Golub and C. F. van Loan. *MATRIX Computations*. John Hopkins University Press, 1996.

[Har92] R. I. Hartley. Estimation of relative camera position for uncalibrated cameras. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 579–587. Springer Verlag, 1992.

[Har94] R. I. Hartley. Projective reconstruction and invariants from multiple images. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 16(10):1036–1041, 1994.

[Har97] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 19(6):580–593, 1997.

[Hes63] O. Hesse. Die cubische Gleichung, von welcher die Lösung des Problems der Homographie von M. Chasles abhängt. *J. reine angew. Math*, 62:188–192, 1863.

[HG93] R. I. Hartley and R. Gupta. Computing matched-epipolar projections. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 549–555, 1993.

[HHLM04] U. Helmke, K. Hüper, P. Y. Lee, and J. B. Moore. Essential matrix estimation via newton-type methods. In *Proc. Intern. Symposium of Mat. Theory of Networks and Systems*, 2004.

[Hil91] E. C. Hildreth. Recovering heading for visually-guided navigation, 1991. A.I. Memo No. 1297, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

[HJ92] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion i: algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117, 1992.

[Hor86] B. K. Horn. *Robot Vision*. McGraw-Hill Higher Education, 1986.

[Hou62] P. V. C. Hough. Methods and means for recognizing complex patterns, 1962. U. S. Patent 3069654.

[HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.

[HT99] J. Hornegger and C. Tomasi. Representation issues in the ml estimation of camera motion. In *Proc. International Conference on Computer Vision (ICCV)*, pages 640–647, 1999.

[Hub64] P. Huber. Robust estimation of a location parameter. *Annals of Math. Stat.*, 53:73–101, 1964.

[HV00] M. Henning and S. Vinoski. *Advances CORBA Programming with C++*. Addison-Wesley, Massachusetts, 4. edition, 2000.

[HZ03] R. I. Hartley and A. Zissermann. *Multiple View Geometry*. Cambridge University Press, 2. edition, 2003.

[IHS+96] Y. Ishikawa, A. Hori, M. Sato, M. Matsuda, J. Nolte, H. Tezuka, H. Konaka, M. Maeda, and K. Kubota. Design and implementation of metalevel architecture in c++ — mpc++ approach. In *Proceedings of the Reflection '96 Conference*, pages 153—166, April 1996.

[IT00] F. Isgro and E. Trucco. A general rank-2 parameterization of the fundamental matrix. In *Proc. Intern. Conf. on Pattern Recognition (ICPR)*, volume 1, pages 868–871, 2000.

[JAHJ00] R. Jeffries, A. Anderson, C. Hendrickson, and R.E. Jeffries. *Extreme Programming Installed*. Addison-Wesley Professional, 2000.

[Jan01] J. R. Janesick. *Scientific Charge-Coupled Devices*. SPIE Press, Bellingham, Washington, USA, 2001.

[JH92] A. D. Jepson and D. J. Heeger. Linear subspace methods for recovering translational direction, 1992. University of Toronto Technical Report RBCV-TR-92-40.

[Kam97] G. Kamberova. Understanding the systematic and random errors in video sensor data, 1997. GRASP Lab., Department of Computer and Information Science, University of Pennsylvania, Tech. Rep., 1997. 4.

[KKA99] H. Kalviainen, N. Kiryati, and S. Alaoutinen. Randomized or probabilistic hough transform: Unified performance evaluation. In *Scand. Conf. Image Anal.*, pages 259–266, Kangerlussuaq, Greenland, 1999. IAPR.

[Kö00] U. Köthe. *Generische Programmierung für die Bildverarbeitung*. Books on Demand, 2000.

[LC96] J. M. Lawn and R. Cipolla. Reliable extraction of the camera motion using constraints on the epipole. In *Proc. of the European Conference on Computer Vision (ECCV)*, volume 2, pages 161–173, 1996.

[LDFP93] Q.-T. Luong, R. Deriche, O. Faugeras, and T. Papadopoulo. On determining the fundamental matrix: Analysis of different methods and experimental results, 1993. INRIA Technical Report 1984.

[LF94] Q.-T. Luong and O. Faugeras. A stability analysis of the fundamental matrix. In *Proc. of the European Conference on Computer Vision (ECCV)*, volume 1, pages 577–588, 1994.

[LF98] Q. Luong and O. Faugeras. On the determination of epipoles using cross-ratios. *Computer Vision and Image Understanding*, 71(1):1–18, 1998.

[LH81] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[LHP80] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proc. Roy. Soc. London B*, 208:385–397, 1980.

[Lib04] Jesse Liberty. *C++ in 21 Days*. SAMS, 5. edition, 2004.

[LMLK94] E. Lutton, H. Maitre, and J. Lopez-Krahe. Contribution to the determination of vanishing points using hough transforms. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 16(4):430–438, 1994.

[May93] S. Maybank. *Theory of Reconstruction from Image Motion*, volume 28 of *Springer Series in Information Sciences*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1993.

[Mee04] P. Meer. *Emerging Topics in Computer Vision*, chapter Robust Techniques for Computer Vision. Prentice Hall, 2004.

[Mid] Middlebury stereo vision page, http://cat.middlebury.edu/stereo.

[MV93] W. Mackens and H. Voss. *Mathematik I für Studierende der Ingenieurwissenschaften*. Heco, 1. edition, 1993.

[Mye95] N. Myers. A new and useful template technique: Traits. *C++ Report Magazine*, 7(5):32–35, 1995.

[Nie94] W. Niem. Robust and fast modelling of 3-d natural objects from multiple views. In *SPIE Proceedings - Image and Video Processing II*, number 2182, pages 388–397, 1994.

[Nis04] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 26(6):756–777, 2004.

[NS04] D. Nistér and F. Schaffalitzky. What do four points in two calibrated images tell us about the epipoles? In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 41–57, 2004.

[Pap65] A. Papoulis. *Probability Random Variables and Stochastic Processes*. McGraw Hill, 1965.

[Pes03] M. D. Pesce. *Programming Microsoft DirectShow for Digital Video and Television*. Microsoft Press, 2003.

[PFTV86] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge (UK) and New York, 1. edition, 1986.

[PG98] J. Ponce and Y. Genc. Epipolar geometry and linear subspace methods: A new approach to weak calibration. *International Journal of Computer Vision*, 28(3):223–243, 1998.

[Phi98] J. Philip. Critical point configurations of the 5-, 6-, 7-, and 8-point algorithms for relative orientation, 1998. TRITA-MAT-1998-MA-13.

[Pol99] M. Pollefeys. *Self-Calibration and Metric 3D Reconstruction from Uncalibrated Image Sequences*. K.U. Leuven, 1999.

[QM89] L. Quan and R. Mohr. Determining perspective structures using hierarchical hough transform. *Pattern Recognition Letters*, 9:279–286, 1989.

[RD05] V. C. Raykar and R. Duraiswami. Very fast optimal bandwidth selection for univariate kernel density estimation, 2005. Computer Science Technical Report CS-TR-4774/UMIACS-TR-2005-73.

[RL85] J. H. Rieger and D. T. Lawton. Processing differential image motion. *Opt. Soc. Amer. A*, 2(2):354–360, 1985.

[RL03] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. Wiley, 2003.

[Rog97] D. Rogerson. *Inside COM*. Microsoft Press, 1997.

[Roh06] H. Rohling. *Skript: Stochastische Prozesse*. Technische Universität Hamburg-Harburg, 2006.

[SB95] S. M. Smith and J. M. Brady. Susan - a new approach to low level image processing, 1995. Technical Report TR95SMS1c.

[Sco92] D. W. Scott. *Multivariate density estimation : theory, practice, and visualization*. Wiley, 1992.

[SGG04a] F. Shafait, M. Grimm, and R.-R. Grigat. Evaluation of a vision based 2-button remote control for interactive television. In *Proceedings of 11th International Workshop on Signals, Systems, and Image Processing (IWSSIP)*, Poznan, Poland, 2004.

[SGG04b] F. Shafait, M. Grimm, and R.-R. Grigat. Low-complexity camera egomotion estimation algorithm for real time applications. In *Proceedings of the 8th IEEE International Multi-Topic Conference (INMIC)*, Lahore, Pakistan, 2004.

[SH05] Julian Smart and Kevin Hock. *Cross-Platform GUI Programming with wxWidgets*. Prentice Hall, 2005.

[Sha48] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.

[SM05] R. Subbarao and P. Meer. Heteroscedastic projection based m-estimators. In *Workshop on Empirical Evaluation Methods in Computer Vision*, San Diego, USA, 2005.

[Smi97] S. M. Smith. Reviews of optic flow, motion segmentation, edge finding and corner finding, 1997. Technical Report TR97SMS1.

[Sny87] J. P. Snyder. *Map Projections; A Working Manual*. U.S. Geological Survey, supersedes bulletin 1532 edition, 1987.

[ST94] J. Shi and C. Tomasi. Good features to track. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, Seattle, USA, Jun 1994.

[Ste99] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Rev.*, 41(3):513–537, 1999.

[Str97] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley, 3. edition, 1997.

[Stu64] J. Stuelpnagel. On the parametrization of the three-dimensional rotation group. *SIAM Review*, 6(4), 1964.

[SWV+00] F. Sauer, F. Wenzel, S. Vogt, Y. Tao, Y. Genc, and A. Bani-Hashemi. Augmented workspace: designing an ar testbed. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality*, pages 47–53, 2000.

[TBM02] T. Thormählen, H. Broszio, and P. N. Meier. Automatische 3d-rekonstruktion aus endoskopischen bildfolgen. In *Bildverarbeitung für die Medizin*, pages 207–210, 2002.

[Tek95] A. M. Tekalp. *Digital video processing*. Prentice-Hall, Inc., 1995.

[TM02] B. Tordoff and D. W. Murray. Guided sampling and consensus for motion estimation. In *Proc. of the European Conference on Computer Vision (ECCV)*, volume 1, pages 82–98, 2002.

[TM04] J. Torres and J. M. Menéndez. A practical algorithm to correct geometrical distortion of image acquisition cameras. In *Proc. Intern. Conf. on Image Processing (ICIP)*, pages 2451–2454, 2004.

[Tri98] B. Triggs. Optimal estimation of matching constraints. In *Proc. SMILE Workshop*, pages 63–77, 1998.

[Tro] Trolltech. Qt: A c++ application development framework.

[Tsa87] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.

[TZ00] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[vdHWG06] H. van der Heijden, F. Wenzel, and R.-R. Grigat. Strategies for fast true motion block matching. In *Proc. Int. Conf. on Comp. Vis. and Applications (VISAPP)*, 2006.

[Vos98] H. Voss. *Mathematik für Studierende der Ingenieurwissenschaften II.* Technische Universität Hamburg-Harburg, 1998.

[VT98] A. Verri and E. Trucco. Finding the epipole from uncalibrated optical flow. In *Proc. International Conference on Computer Vision (ICCV)*, pages 987–991, 1998.

[WG05a] F. Wenzel and R.-R. Grigat. Design aspects for the rapid development of image processing algorithms. *WSEAS Transactions on Information Science and Applications*, 2(9):1312–1320, 2005.

[WG05b] F. Wenzel and R.-R. Grigat. A framework for developing image processing algorithms with minimal overhead. In *Proc. WSEAS Int. Conf. on Signal, Speech and Image Proc. (SSIP)*, 2005.

[WG06a] F. Wenzel and R.-R. Grigat. Representing directions for hough transforms. In *Proc. Int. Conf. on Comp. Vis. and Applications (VISAPP)*, 2006.

[WG06b] F. Wenzel and R.-R. Grigat. *Skript: 3D Computer Vision.* Technische Universität Hamburg-Harburg, 2006.

[WJ95] M.P. Wand and M. Jones. *Kernel Smoothing.* Chapman and Hall, 1995.

[WM00] P.F. Whelan and D. Molloy. *Machine Vision Algorithms in Java: Techniques and Implementation.* Springer (London), 2000.

[WNDS99] M. Woo, J. Neider, T. Davis, and D. Schreiner. *OpenGL Programming Guide - The Official Guide to Learning OpenGL, Version 1.2.* Addison-Wesley, 3. edition, 1999.

[Zha97] Z. Zhang. Motion and structure from two perspective views: From essential parameters to euclidean motion through the fundamental matrix. *J. Opt. Soc. Am. A*, 14(11):2938–2950, 1997.

[Zha98] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, 1998.

[ZL01] Z. Zhang and C. Loop. Estimating the fundamental matrix by transforming image points in projective space. *Computer Vision and Image Understanding*, 82(5):174–180, 2001.

Curriculum Vitae

Personal Information

Name	Fabian Wenzel
Date of Birth	19.03.1975
Place of Birth	Hamburg, Germany

Education

1981-1985	Primary School, Grundschule Oststeinbek
1985-1994	Secondary School, Gymnasium im Schulzentrum Glinde
1995-2001	Study of Electrical Engineering, Hamburg University of Technology
1999	Exchange Semester, University of California at Berkeley
2000	Project thesis (Studienarbeit) "Design and Integration of a Real-time Tracking Subsystem for Medical Augmented Reality Applications", Siemens Corporate Research, Princeton
2001	Diploma thesis "Three-dimensional Reconstruction of Motion Trajectories by Digital Image Processing using two or more Cameras", Vision Systems Group, Hamburg University of Technology
2002-2003	Undergraduate Studies of Economics for Natural Scientists (Wirtschaftswissenschaften für Naturwissenschaftler), FernUniversität Hagen
Since 2001	Ph.D. Candidate (Doktorand), Vision Systems Group, Hamburg University of Technology

Profession

2001-2006	Research Assistant, Vision Systems Group, Hamburg University of Technology
Since 2006	Research Scientist, Philips Research Europe, Hamburg

Miscellaneous Activities

1994-1995	Community Service (Zivildienst), Sozialstation Barsbüttel

Awards and Honors

1998	Philips Vordiplompreis
1999	Foreign Exchange Scholarship, Stiftung zur Förderung der Technischen Universität Hamburg-Harburg

Hamburg, May 26, 2007