INAUGURAL — DISSERTATION

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht – Karls – Universität
Heidelberg

vorgelegt von
Diplom-Informatiker Andreas Grützmann
aus Weimar

Tag der mündlichen Prüfung: 7.10.2009

Reconstruction of Moving Surfaces of Revolution from Sparse 3-D Measurements using a Stereo Camera and Structured Light

Gutachter: Prof. Dr. Bernd Jähne
Prof. Dr. Dr. h.c. Hans Georg Bock

Abstract

The aim of this thesis is the development and analysis of an algorithmic framework for the reconstruction of a parametric model for a moving surface of revolution from a sequence of sparse 3-D point clouds. A new measurement device with a large field of view that allows for acquisition of three-dimensional data in challenging environments is utilized. During the measurement process, the observed object may be subject to motion which can be described in terms of an analytical model.

The proposed method is developed and analyzed, along with an application for the surface reconstruction of a wheel. It is shown that the precision of the coarse surface model independently fitted to each measurement can be significantly improved by fitting a global model to all measurements of the sequence simultaneously. The global model also takes into account object's motion.

The three-dimensional point clouds are acquired by an optical device which consists of a stereo camera and an illumination unit projecting a dot pattern. A rather high density of surface points within the camera's field of view is established by means of multiple laser projectors. Through an elaborate calibration procedure of the stereo camera and the projector, and by utilizing the trifocal epipolar constraints of the measurement device, a high accuracy in the three-dimensional point cloud is achieved.

Zusammenfassung

Das Ziel dieser Arbeit ist die Entwicklung und Analyse der algorithmischen Methodik zur Rekonstruktion eines parametrischen Oberflächenmodells für ein rotationssymmetrisches Objekt aus einer Sequenz von dünnen 3D-Punktwolken. Dabei kommt ein neuartiges Messsystem mit großem Sichtfeld zum Einsatz, dass auch in schwierigen Bedingungen eingesetzt werden kann. Das zu vermessende Objekt kann während der Aufnahme der Sequenz einer als analytisches Modell formulierbaren Bewegung unterliegen.

Das Verfahren wird anhand einer praktischen Anwendung zur Oberflächenrückgewinnung eines Rades analysiert und entwickelt. Es wird gezeigt, dass die durch Fit eines einfachen Models für jede Einzelmessung erzielbare Genauigkeit durch Anpassung eines globalen Modells unter gleichzeitiger Einbeziehung aller Einzelmessungen und unter Berücksichtigung eines geeigneten Bewegungsmodells erheblich verbessert werden kann.

Die Gewinnung der dreidimensionalen Punktdaten erfolgt mit einem Stereokamerasystem in Verbindung mit aktiver Beleuchtung in Form eines Punktmusters. Eine relativ hohe Punktdichte im gesamten Sichtfeld des Stereokamerasystems wird durch Verbindung mehrerer Laserprojektoren zu einer Projektionseinheit erzielt. Durch exakte Kalibrierung des Kamerasystems und der Projektionseinheit wird trotz großer Streuung der Laserpunkte im Kamerabild unter Ausnutzung der trifokalen geometrischen Bedingungen eine hohe Genauigkeit in den dreidimensionalen Punktdaten erzielt.

Contents

Acknowledgements

I would like to thank all those who contributed to this thesis. First of all, my thanks go to my doctoral advisor Prof. Dr. Bernd Jähne from the Heidelberg Collaboratory for Image Processing (HCI), University of Heidelberg. His support in both, scientific and in administrative issues was the key to the successful conclusion of this thesis. I also want to thank Walter Happold from the Corporate Sector Research at Robert Bosch GmbH, Stuttgart, who proposed this research and gave me the opportunity to work in a project with immediate importance for future products.

Thanks go to my colleagues in Schwieberdingen and Hildesheim for their cooperation, particularly to Dr. Matthias Gebhard and Dr. Steffen Abraham for their valuable inputs and support. Special thanks go to Jochen, Marco, Mark, Thomas, Ming, Matthias, Andreas, Stefan, Florian, Sabine, and Patrick (in chronological order) for creating an inspiring and pleasant working environment.

I would also like to thank present and past members of the HCI in Heidelberg. It has been a pleasure being a member of the group, partaking in the group's activities, and being able to discuss recent problems and results with so many excellent researchers working on interdisciplinary cutting edge topics. Special thanks go to Björn, Claudia, and Linus.

I also want to express my gratitude to all the people who supported me up to this point. First of all, I would like to thank my parents for their encouragement and assistance, my wife for her patience when working long weekends and nights. I want to thank Gerardo, Manuel, Matthias, Joachim, and Thomas for their help with proof reading. I am also indebted to my friends and my computer science teachers at University of Jena, in particular Prof. Dr. Klaus Küspert.

Notation and Abbreviations

The format of the symbols used throughout this thesis determines their meaning. Unless specified otherwise, a bold lower case letter, e.g. $\mathbf{x}$, denotes a vector in $\mathbb{R}^n$ and a bold upper case letter, e.g. $\mathbf{A}$, denote a matrix in $\mathbb{R}^{m \times n}$ for any $m, n \in \mathbb{N}$. The superscript ^{T} denotes the vector or matrix transpose. Upper case calligraphic letters, e.g. $\mathcal{A}$, denote a finite or an infinite set depending on the context. The hat above a symbol, e.g. $\hat{\mathbf{x}}$, indicates that it is an approximation to the unknown true value. The superscripts ^{L} and ^{R} discriminate between variables that coexist for simultaneously processed data originating from the left and the right camera. The unit pel refers to *pixel elements* and is used for the coordinates in the domain of a digital image.

Abbreviations

CV	Cross Validation
GCV	Generalized Cross Validation
DMD	Digital Micromirror Device
DoG	Difference of Gaussian
LCD	Liquid Crystal Display
LCOS	Liquid Crystal on Silicon
LED	Light Emitting Diode
LoG	Laplacian of Gaussian
LMedS	Least Median of Squares
MCS	Multiple Camera System
PCA	Principal Component Analysis
PPM	Point Pattern Matching
SoR	Surface of Revolution
TLS	Total Least Squares

ETWAS KURZ-GESAGTES KANN DIE FRUCHT UND ERNTE VON VIELEM LANG-GEDACHTEN SEIN.

(Friedrich Nietzsche)

1 Introduction

The representation of the three-dimensional shape of an existing object by a computer model is a common task for a wide range of applications in industrial inspection, reverse engineering, and virtual reality. The construction of such a model requires the availability of three-dimensional data acquired from the surface of the given object using an eligible tool. Optical methods [Schwarte et al., 1999] have been motivated by the principle of human depth perception and play an important role in object digitization as they provide means for non-destructive and contact-free three-dimensional shape measurement for objects of almost arbitrary form and scale. The availability of high-resolution digital cameras in combination with powerful computers with high speed interfaces at low-cost have made computationally complex systems feasible for industrial use. Even though the resolution of conventional optical measurement (confocal microscopy) is physically limited by the wave length of light, recent research activities have found a way to operate below this limit [Klar et al., 2001].

The amount of data that is digitized basically depends on the method employed and the time spent. Whilst several devices sequentially measure the distance of single points in space, other state of the art optical scanning devices are able to acquire up to 10^6 points per measurement. The measurement task for a given object in a given situation requires the choice of the appropriate method. From all available options, a method has to be chosen that best fits the objects surface characteristics, the given situation and the requirements on the measurement task.

The digitized surface data is usually processed in a further step in order to identify a desired computer model representation. The intention of this model extraction step is to provide a compact representation of a potentially huge amount of measurement data and eventually get access to features like the object's shape and orientation for use in a measurement task or for object classification. Popular object representation techniques are meshes [Hoppe et al., 1992], appearance based models using landmark points [Cootes et al., 2001], statistical surface characterization [Schmähling, 2006] and parametric surface models.

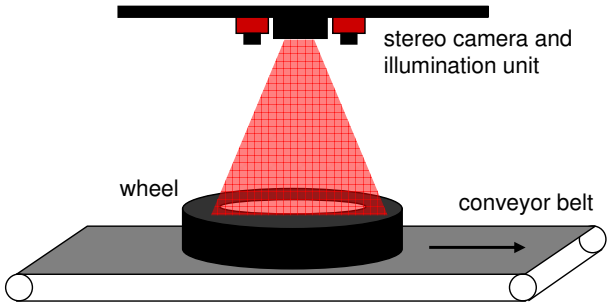


Figure 1.1: Experimental setup

1.1 Motivation

Optical three-dimensional measurement devices are widely used for quality control and metrology in industrial applications [Chen et al., 2000; Kowarschik et al., 2000]. The need for fast and reliable results in challenging environments has led to the development of new concepts and further enhancement of both, the measurement hardware and the model extraction algorithms. In turn, with the introduction of new concepts, new fields of application have been identified.

The reconstruction of a moving surface of revolution is motivated by industrial wheel inspection, where the observed wheel is subject to motion with respect to the measurement device along an approximately linear trajectory. The inspection task involves the reconstruction of a detailed surface model and the motion trajectory within the field of view. It is thus necessary to acquire a depth map of the whole object at a point in time by means of a surface sampling as dense as possible. In order to provide a well quantization of the trajectory, depth maps should be acquired at as many points in time as possible, while the object is passing through the field of view. A further requirement for the inspection task is that it shall produce consistent data in an operating environment with inhomogeneous ambient light for objects of undefined and varying surface texture. Due to these limitations the system needs to work independently of the surface texture. Active triangulation with structured illumination is applicable under the above constraints if a light source of sufficient intensity is utilized.

An exemplary situation of a wheel inspection application is depicted in Figure 1.1, where a wheel is loaded onto a conveyor belt and carried, while being observed by a 3-D measurement device installed at a fixed position above the conveyor. The device is oriented such that the longest extension of the field of view is parallel to the direction of the observed object's motion. A sequence of measurements is acquired while the wheel is passing through the field of view.

1.2 Related work and Contributions

Three-dimensional shape reconstruction of a wheel has been considered in various applications. An approach based on the intensity images of a stereo camera for an autonomous tire disassembly station has been proposed by [Büker et al., 2001]. A dense depth map is reconstructed from stereo correspondences assuming that sufficient texture is provided by the wheel. Structured light has been used by [Scholz et al., 2007] for high-precision tire surface reconstruction and inscription extraction. The tire is loaded on a turn-table and its profile is measured continuously by a laser line and a camera synchronized with the rotation of the turn-table.

In this thesis a different approach is presented which enables for shape reconstruction of a tire or a wheel without the implicit assumption of surface texture availability. Furthermore, the wheel may be subject to motion along an unknown smooth trajectory which can be described analytically. The setting of the application indicates the use of a laser illumination unit projecting a point pattern onto the object's surface. The illumination device is integrated into a stereo camera as an additional component, and the algorithms for identification of point correspondences and the 3-D triangulation are presented. As a result of the object's motion within the illuminated area and registration of the measurements, a dense surface sampling is acquired.

The surface of the wheel is represented by a surface of revolution, and thus the reconstructed shape is the profile outlined by the major part of the wheel. Surfaces of revolution are very important in industrial applications and the mathematical fundamentals for reconstruction of a surface of revolution from a scattered point sampling have been studied by [Elsässer & Hoschek, 1996; Pottmann & Randrup, 1998; Qian & Huang, 2004]. However, a single sparse point cloud does not allow for the extraction of a precise surface model, and additional surface samples are desired wherever possible.

This work addresses this limitation and presents an extension of the existing algorithms for mutual reconstruction of shape and motion from multiple sparse measurements. A sequence of sparse point clouds is acquired while the wheel is passing through the field of view of the measurement device, and a parametric surface model and the motion trajectory are recovered simultaneously in a mutual algorithmic framework. Initially, a coarse model is estimated for each sparse measurement with a high level of uncertainty. The mutual approach overcomes these drawbacks by providing access to the information of the whole sequence to all measurements.

Figure 1.2: Developments in this thesis

1.3 Thesis outline

This thesis is organized in six chapters including the introduction. Chapter 2 provides a summary of computer vision fundamentals which are the basis to the developments in the following chapters. The following chapters each individually cover one of three basic concepts illustrated in Figure 1.2. A novel calibration procedure of an optical measurement device using multiple laser projectors is presented in Chapter 3. The methodology for three-dimensional data acquisition using the previously introduced measurement device is covered in Chapter 4. In Chapter 5, the algorithms for reconstruction of a surface of revolution are developed and applied to the reconstruction of a wheel. Finally, Chapter 6 summarizes the results and provides a conclusion.

2 Foundations

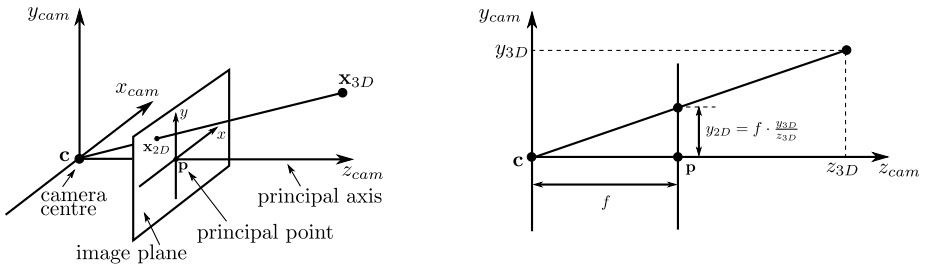
The goal of this chapter is to provide a reference of several well established computer vision concepts that are the basis to the developments throughout this thesis. The state of the art of each such concept is presented by a review of the literature. In some cases the theory is illustrated using simple examples.

The basics of three-dimensional data acquisition using multiple digital cameras are covered in Section 2.1 – Section 2.3 and an overview of the most important topics in object reconstruction given scattered three-dimensional surface data are introduced in Section 2.4 – Section 2.8.

2.1 Optical 3-D Measurement

Optical 3-D measurement methods allow for non-destructive, contact-free, and accurate reconstruction of the three-dimensional shape of an object even over long distances. The known methods can be classified into one of the three categories, triangulation, time-of-flight, and interferometry. An overview over all existing methods and their particular properties are given in [Schwarte et al., 1999] and [Chen et al., 2000]. The most widely used techniques are based on *triangulation*, where an unknown point $\mathbf{x}$ is determined by the triangle formed by a known optical basis and the two lines starting from each end of this basis and pointing to $\mathbf{x}$.

A *multiple camera system* (MCS) is a very commonly used measurement device based on triangulation. It consists of at least two optical units (e.g. digital camera, laser, video projector) with overlapping field of view. A single camera may be placed at multiple positions and thus virtually establish a MCS. A MCS utilizing a projection unit that generates a pattern on the surface of the object of interest is called an active system. If the MCS consists of imaging devices only or does not consider the projection unit in the triangulation it is called a passive system. In either case, the imaging parameters and the geometry of the optical units have to be estimated in an initial calibration procedure. An introduction to the calibration of a camera is given in Section 2.1.1. Section 2.1.3 gives an overview over devices and patterns utilized in active systems. MCS are well established in diversified fields, such as reverse engineering [Wiora, 2001], industrial quality inspection [Kowarschik et al., 2000], medicine, cultural heritage, archeology, security surveillance, and many others.



(a) Pinhole camera model

(b) Geometry of ordinate

Figure 2.1: Pinhole camera model and geometry in pinhole camera

2.1.1 Camera Models and Calibration

Triangulation in a MCS requires knowledge of the mapping from the 3-D world to the camera image and the location of the camera in the world. The parameters describing the mapping are referred to as the cameras *intrinsic* parameters and the location parameters in a fixed world coordinate frame are called the *extrinsic* parameters. The goal of the *camera calibration* procedure is the estimation of the intrinsic and the extrinsic parameters of a particular camera by an eligible method. This section intends to give a rough introduction to the basic mathematical concepts of multiple camera systems. For a deeper insight the reader is referred to the text books [Faugeras, 2001; Luhmann, 2003; Hartley & Zisserman, 2004].

The six extrinsic parameters of a camera are expressed in terms of a 3×3 rotation matrix $\mathbf{R}$ and the 3-vector $\mathbf{c}$ representing the position of the camera center in world coordinates. The number of intrinsic parameters depends on the complexity of the chosen camera model. The simple *pinhole camera model* assumes central projection and has three intrinsic parameters, the coordinates of the principal point $\mathbf{p} = (x_0, y_0)$ and the focal length f. The *finite projective camera* is the most widely used model and extends the pinhole camera by considering anisotropic scale using two parameters α_x and α_y and an additional *skew* parameter s which usually is zero. A further extension of projective camera also considers *lens distortion*, which causes the image points to be displaced radially or tangentially from their actual position in the image plane. The distortion model can be arbitrarily complex though usually a power series expansion with two or three parameters is used to model radial distortion. If applicable, two further parameters are used for tangential distortion [Heikkilä & Silven, 1997; Zhang, 2000]. In [Abraham & Hau, 1997] orthogonal polynomials (see 2.4.3) have been used to model general physically motivated lens distortion with the optimal number of parameters determined automatically.

Homogeneous coordinates[1] are used to represent the projective transform of a point as product of a matrix with the point. The mapping from the 3-D world to the image plane of a distortion free camera can thus be written as

$$\tilde{\mathbf{x}}_{2D} = \mathbf{KR}\left[\mathbf{I}\right| - \tilde{\mathbf{c}}]\,\tilde{\mathbf{x}}_{3D} \tag{2.1}$$

where

$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

is called the *camera calibration matrix*, $\tilde{\mathbf{c}} = (x_c, y_c, z_c, 1)$ is the camera center, $\tilde{\mathbf{x}}_{3D} = (x_{3D}, y_{3D}, z_{3D}, 1)$ is a point in 3-D space, and $\tilde{\mathbf{x}}_{2D} = (x_{2D}, y_{2D}, 1)$ its image in the camera. The matrix $\mathbf{P} = \mathbf{KR}\left[\mathbf{I}\right| - \tilde{\mathbf{c}}]$ is called the camera matrix. The distorted image mapping is usually described by the collinearity equations

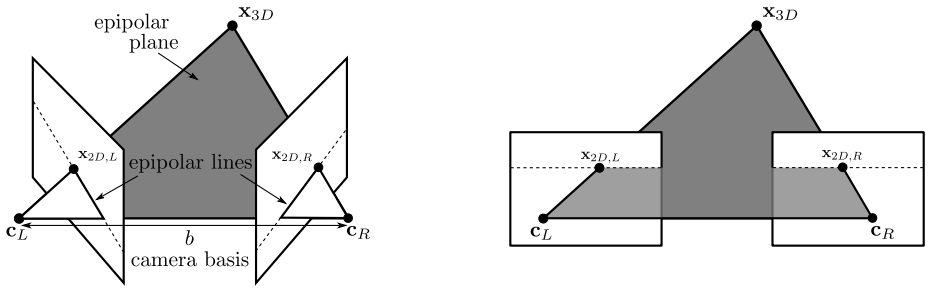
$$x_{2D} = x_0 - \alpha_x \frac{r_{11}\left(x_{3D} - x_c\right) + r_{12}\left(y_{3D} - y_c\right) + r_{13}\left(z_{3D} - z_c\right)}{r_{31}\left(x_{3D} - x_c\right) + r_{32}\left(y_{3D} - y_c\right) + r_{33}\left(z_{3D} - z_c\right)} + \delta x \tag{2.3}$$

$$y_{2D} = y_0 - \alpha_y \frac{r_{21}\left(x_{3D} - x_c\right) + r_{22}\left(y_{3D} - y_c\right) + r_{23}\left(z_{3D} - z_c\right)}{r_{31}\left(x_{3D} - x_c\right) + r_{32}\left(y_{3D} - y_c\right) + r_{33}\left(z_{3D} - z_c\right)} + \delta y \tag{2.4}$$

where the r_{ij} are the elements of the rotation matrix $\mathbf{R}$ and δx and δy are the distortion correction terms.

The actual calibration process estimates the intrinsic and extrinsic parameters of the chosen camera model automatically from correspondences between points in 3-D space and their image in the image plane. Basically, three different types of camera calibration can be distinguished. In *photogrammetric calibration* a calibration object whose geometry in 3-D space is known with good precision is observed. Such an object usually consists of two or three planes orthogonal to each other. This approach requires an expensive apparatus, and an elaborate setup. In *self calibration* the rigidity of the observed scene is utilized and the camera parameters are computed from correspondences between at least three images. If no absolute extension of any observed objects is given, the extrinsic parameters can be computed only up to similarity. Finally, a recent third approach takes advantage of both previous types. Images of a planar pattern placed in at least two different positions are acquired and the constraint on the planarity of the pattern is plugged into Equation (2.1) leading to a minimization problem that can be solved by maximum likelihood estimation [Zhang, 2000]. In the first stage of the camera calibration procedure a distortion free model is assumed and distortion parameters are estimated in a second stage from the errors of the initial model.

[1] A vector $\mathbf{x}$ in p-space is represented in homogeneous coordinates as $(p+1)$-vector and will be denoted by $\tilde{\mathbf{x}}$. where the additional dimension defines the scale of the vector.



(a) Stereo setup

(b) Rectified stereo setup

Figure 2.2: Geometry of a general stereo setup and a recitified stereo setup

2.1.2 Stereo Camera Geometry

The geometry of a stereo camera setup is described by its *epipolar geometry*. A 3-D point $\mathbf{x}_{3D}$ imaged by the two cameras of a stereo setup produce a pair of corresponding points $\mathbf{x}_{2D,L}$ and $\mathbf{x}_{2D,R}$. The projection of the line of sight of the point in the image plane of one camera to the image plane of the other camera is called the *epipolar line*. The *epipolar plane* is the plane spanned by corresponding epipolar lines. The algebraic relation of the corresponding points is represented by

$$\tilde{\mathbf{x}}_{2D,L}^T \, \mathbf{F} \, \tilde{\mathbf{x}}_{2D,R} = 0. \tag{2.5}$$

where $\mathbf{F}$ is a 3×3 rank 2 matrix, called the *fundamental matrix*, and the points are given in homogeneous coordinates. The fundamental matrix can be computed from at least eight point correspondences in the stereo image pair and the camera matrices can be derived from the fundamental matrix when the image of at least five points (at most three points are coplanar) with known Euclidean position are given. A general stereo setup with its epipolar geometry labeled is shown in Figure 2.2(a).

The process of *image rectification* refers to the transformation of both image planes of a stereo setup such that pairs of conjugate epipolar lines become collinear and parallel the x-axes of the image plane [Fusiello et al., 2000]. The advantage of a rectified stereo image pair is the complexity reduction of the point correspondence problem to a search along the same x-axes. A schematic illustration of a rectified stereo setup is shown in Figure 2.2(b). The rectified image I' is computed from the original image I by the inverse mapping $\left(f_x^{-1}, f_y^{-1}\right)$ from the rectified image coordinates to the original coordinates

$$I'(x,y) = g\left(I\left(f_x^{-1}(x,y), f_y^{-1}(x,y)\right)\right) \tag{2.6}$$

using a desired interpolation function g, e.g. linear interpolation.

2.1.3 Structured Light

Structured light is utilized in multiple camera systems in order to provide a well-defined pattern on the surface of the observed object which enables for fast and reliable detection of correspondences needed for triangulation. This depth measurement method is referred to as *active triangulation*. The projector is usually regarded as an inverse camera and triangulation is performed for correspondences between camera and projector. The inverse camera approach requires a calibration procedure to be carried out for the projector as well.

Different technologies for active illumination have been utilized. The most commonly used light emission technologies are light emitting diodes (LED), lamps as integrated in standard video projectors, and laser diode modules. In combination with coherent light sources (lasers) a spatial pattern is generated by a diffraction grating mounted in front of the laser module. In combination with other light sources transmissive or reflective devices such as liquid crystal displays (LCD), liquid crystal on silicon displays (LCoS) or digital micromirror devices (DMD) are utilized for pattern generation. Recently [Notni et al., 2004] have proposed the use of OLED displays. The latter displays allow for generation of dense patterns with very high resolution whereas a laser diode is very bright and thus allows for application in environments of inhomogeneous ambient illumination.

The commonly used structured light patterns can be divided into three categories by their complexity, (1) uncoded patterns, (2) spatially coded patterns, and (3) temporally coded patterns. Patterns which are generated by a laser diode (e.g. points, lines, and grids) belong to the first category. Such patterns usually result in a rather sparse sampling of the observed object because depth is acquired only at a small number of points or along one or more lines. Coded patterns are dense in the sense that the depth for any point on the illuminated area of an objects surface can be acquired. Within a spatially coded pattern the local neighborhood of any point in the pattern is unique. The 3-D scene can be reconstructed from a single image but the correspondence search algorithm for patterns of this category is rather complex and may be inapplicable for objects with surface discontinuities. Various kinds of pseudo random patterns and color coded patterns have been used. In patterns of the third category a discrete or continuous function is encoded over the projector domain by a sequence of patterns successively projected to the observed scene. The function's inverse for a code word extracted at a fixed point of the illuminated object gives the corresponding point in the projector domain. The plain binary code due to [Posdamer & Altschuler, 1982] and its improved version of Gray code sequences [Inokuchi et al., 1984] provide a discrete function. The gap to a continuous code has been bridged using phase shifted sinusoidal patterns [Malz, 1999] with increased spatial resolution. A systematic review of pattern codification strategies has been given by [Salvi et al., 2004].

2.2 Feature Detection

Many computer vision tasks rely on the identification of features such as corners, edges, blobs or other structures that emerge from their neighborhood in the domain of an image. Many detectors for such *points of interest* exist in the literature. A recent review on feature detectors and an evaluation with respect to repeatability and information content is given in [Schmid et al., 2000]. The variance of the size of features in an image is usually not known in advance. Features of any size or scale are considered by scale invariant feature detection algorithms introduced in Section 2.2.1. The focus in Section 2.2.2 is particularly on detectors for blob like structures of varying size as is the case with spots in the image of an object illuminated by a laser projector. In Section 2.2.3 a method for deriving a neighborhood relation for a set of unorganized image features is given.

2.2.1 Scale Invariance

An object appears in an image with varying scale depending on the observation distance and the angle of view. In order to isolate, analyze and interpret structures of different and unknown scale in an image it is necessary to construct a multi-scale representation where fine-scale structures are suppressed with increasing scale [Lindeberg, 1994]. The main idea is to generate a one-parameter family of derived images

$$L\left(x, y, \sigma\right) = I\left(x, y\right) \otimes G\left(x, y, \sigma\right) \tag{2.7}$$

obtained by convolving the original image $I\left(x, y\right)$ with Gaussian kernels

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

of increasing variance σ^2. The effect of the Gaussian kernel $G\left(x, y, \sigma\right)$ is to suppress most of the structures with characteristic size less than σ.

2.2.2 Blob Detection

The term *blob detection* refers to algorithms which are designed for identification of regions in a digital image that are either brighter or darker than its neighborhood. The existing methods can be classified into (1) methods based on local extrema in the intensity signal and (2) differential methods with automatic scale selection [Lindeberg, 1998]. The first class of methods is introduced shortly in Section 2.2.2. All other approaches belong to the class of differential methods.

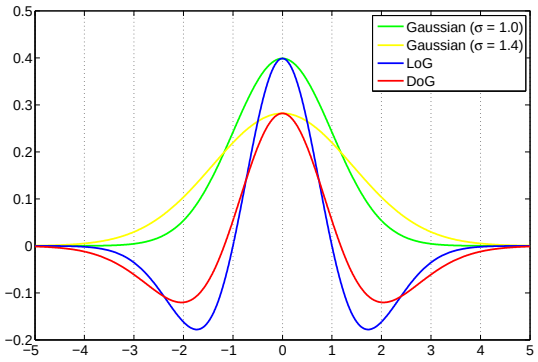


Figure 2.3: Comparison of LoG and DoG for $k = \sqrt{2}$

Regions

The first class of blob detection algorithms is based on local extrema in the intensity signal. It is assumed that the blob like structures in the original or preprocessed image are circumscribed by a well defined boundary. The regions defined by such boundaries are determined using a thresholding algorithm, e.g. [Otsu, 1979]. Depending on its size and shape a region may be rejected or stored for further use. The location of a region is specified by its center of gravity or using geometric features like the center of an ellipse which is fitted to the boundary.

Laplacian of Gaussian (LoG)

The *Laplacian of Gaussian*, also known as *Mexican Hat* wavelet due to the shape of its response, is the second derivative of a Gaussian. Given the multi-scale representation L of an image I, the scale-normalized LoG of I at scale σ is the normalized sum of all unmixed second partial derivatives in the Cartesian coordinates

$$\nabla^2 L\left(x, y, \sigma\right) = \sigma^2 \left(\frac{\partial^2 L\left(x, y, \sigma\right)}{\partial x^2} + \frac{\partial^2 L\left(x, y, \sigma\right)}{\partial y^2} \right). \tag{2.9}$$

The LoG operator results in strong positive responses for dark blobs of extent σ and strong negative responses for bright blobs of similar size. The points of interest for blob detection are the local minima/maxima with respect to both, scale and space [Lindeberg, 1998]. The zero-crossings of $\nabla^2 L_\sigma$ can also be used for edge detection [Marr & Hildreth, 1980].

Difference of Gaussian (DoG)

The *Difference of Gaussian* is defined as difference

$$D(x, y, \sigma) = L(x, y, k \cdot \sigma) - L(x, y, \sigma) \tag{2.10}$$

of two nearby scales $k \cdot \sigma$ and σ. The DoG function provides a close approximation to the scale-normalized Laplacian of Gaussian $\sigma^2 \nabla^2 L$ [Marr & Hildreth, 1980]. The relationship between D and $\sigma^2 \nabla^2 L$ can be understood from the heat diffusion equation

$$\frac{\partial L}{\partial \sigma} = \sigma \nabla^2 L. \tag{2.11}$$

With the finite difference approximation

$$\frac{\partial L}{\partial \sigma} \approx \frac{L(x, y, k \cdot \sigma) - L(x, y, \sigma)}{k \cdot \sigma - \sigma} \tag{2.12}$$

it follows

$$(k - 1)\, \sigma^2 \nabla^2 L \approx L(x, y, k \cdot \sigma) - L(x, y, \sigma)\,. \tag{2.13}$$

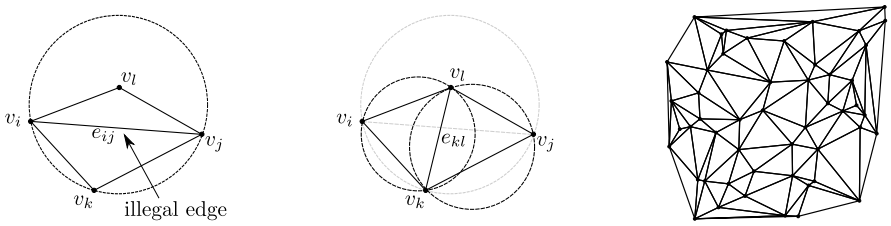
The value often chosen in practice is $k = 1.6$ [Marr & Hildreth, 1980; Lowe, 2004]. Two one-dimensional Gaussians, their difference and the Laplacian of the first Gaussian is shown in Figure 2.3.

Determinant of Hessian (DoH)

The determinant of Hessian is based on the second derivative of the image intensity signal $f(x, y)$

$$\det HL(x, y, \sigma) = \sigma^2 \left(\frac{\partial^2 L(x, y, \sigma)}{\partial x^2} \cdot \frac{\partial^2 L(x, y, \sigma)}{\partial y^2} + \frac{\partial^2 L(x, y, \sigma)}{\partial x \partial y} \right) \tag{2.14}$$

where H denotes the Hessian matrix of L. The determinant of Hessian also responds to saddles [Lindeberg, 1998].



(a) (b) (c)

Figure 2.4: Adjacent triangles with (a) illegal edge and (b) legal edge obtained by edge flip; (c) Delaunay triangulation of a random set of 50 points

2.2.3 Delaunay Triangulation

Let $\mathcal{V} = \{v_1, ..., v_n\}$ be the abstract set of features detected in an image. A neighborhood relation $\mathcal{N}$ among the features with respect to the shortest Euclidean distance is defined by the Delaunay triangulation. A *triangulation* $\mathcal{T}$ of a set of points in the plane is in general a planar embedding of a graph with vertices $\mathcal{V}$ and whose set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is maximal, such that no edge connecting two vertices can be added without destroying the planarity of $\mathcal{T}$. It follows from the definition that all bounded faces of $\mathcal{T}$ are triangles. The *unbounded face* is the area that is not bounded by a single triangle. The edges of the unbounded face in $\mathcal{T}$ make the convex hull of $\mathcal{V}$.

Let $e_{ij} = (v_i, v_j)$ be an edge in a triangulation $\mathcal{T}$ of $\mathcal{V}$. If e_{ij} is not an edge of the unbounded face it has two adjacent triangles $v_i v_j v_k$ and $v_i v_j v_l$. The edge e_{ij} is called *illegal* if v_l lies in the interior of the circle through v_i, v_j, and v_k. The example in Figure 2.4(a) shows the convex quadriliteral formed by the adjacent triangles $v_i v_j v_k$ and $v_i v_j v_l$ with the illegal edge (v_i, v_j) and the circle through v_i, v_j, and v_k containing v_l. The illegal edge is eliminated by the *edge flip* operation removing e_{ij} and inserting e_{kl} instead. This process is shown in Figure 2.4(b). The *Delaunay triangulation* $\mathcal{T}_D$ of $\mathcal{V}$ is a triangulation that does not contain any illegal edge. The Delaunay triangulation is unique if and only if $\mathcal{V}$ contains no four points on the same circle without any point inside the circle. Let b the number of vertices on the convex hull, it is not hard to prove by Euler's formula that the number of triangles in $\mathcal{T}_D$ is $2n - 2 - b$ the number of edges is $3n - 3 - b$.

For any given triangulation $\mathcal{T}$ of $\mathcal{V}$, $\mathcal{T}_D$ can be obtained by simply flipping all illegal edges until all edges are legal. However, the asymptotic runtime of this algorithm is not optimal. An optimal incremental algorithm for the computation of the Delaunay triangulation runs in $O(n \log n)$ expected time. The reader is refered to [de Berg et al., 2000, ch. 9] for the details.

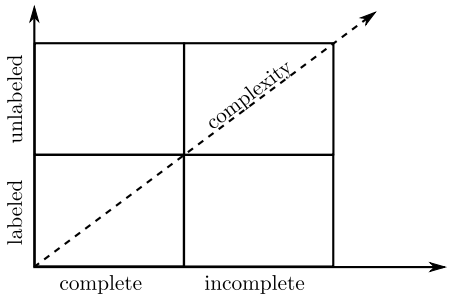


Figure 2.5: Complexity of point pattern matching

2.3 Stereo Matching

The detection of correspondences in multiple images is an important task for many computer vision applications, such as image registration, motion tracking, and object recognition. *Stereo matching* is the problem of finding the correspondences in two images taken from different points of view. Dense stereo correspondence algorithms estimate the disparity in all image regions, even those that are occluded or without texture. A framework for evaluation of existing algorithms using a set of test images has been developed by [Scharstein & Szeliski, 2002]. The performance of these algorithms depends on the available texture in the source images and are thus unfeasible for images of an unstructured scene. Feature based algorithms find correspondences only for points, lines, and other structures extracted from the input images. In *point pattern matching* (PPM) the features are points and each point is represented by its coordinates $\mathbf{x} = (x, y)$ and assigned an optional label vector $\mathbf{v}$ with information about intensity, gradient, size, and shape. The PPM problem has been studied by many authors and a recent review of the literature is given in [Rangarajan et al., 1997; Li et al., 2003; Yin, 2006]. The authors of the more recent publication distinguish four complexity classes as shown in the matrix in Figure 2.5. In horizontal direction it is distinguished between complete and incomplete matching where a complete matching means that there exists an exact one-to-one mapping involving all points. The vertical direction discriminates the availability of labels for the features. The complexity increases with the positive direction of both axes and the unlabeled and incomplete matching is the most complex case.

This section gives an overview of popular PPM techniques with their respective scope and limitation. In Section 2.3.1 a selection of algorithms for the PPM problem is introduced. A graph theoretic view on stereo matching and a combinatorial optimization algorithm for its solution is given in Section 2.3.2.

2.3.1 Point Pattern Matching

The PPM problem has found a large interest and most of the presented algorithms can be categorized into one a few major categories. A very intuitive approach is the individual point matching based on feature labels provided that the labels are distinct. If such labels are not given the algorithms need to find the optimal matching with respect to a global measure. Most of these algorithms assume a global transformation which preserves inter-point relations between the point patterns to match. Each of the following paragraphs gives an outline of a category of PPM algorithms and summarizes the most frequently referred literature.

Clustering

The clustering approach assumes the point patterns to underly an affine transformation and simultaneously estimates the transformation parameters and the point matching. The transformation parameters are computed for all combinations of point pairs from both point sets. The strongest clusters in the parameter space represent the most likely transformation parameters for the best match. These methods are computationally intensive due to the large number of combinations of point pairs and the dimensionality of the parameter space. Clustering has been applied to PPM by [Stockman et al., 1982; Goshtasby & Stockman, 1985; Umeyama, 1991; Yuen, 1993; Chang et al., 1997].

Eigenvector Approach

This algorithm proposed by [Scott & Longuet-Higgins, 1991] builds the *proximity matrix* $\mathbf{G}$ from the Gaussian weighted distances

$$G_{ij} = e^{-d_{ij}^2/2\sigma^2} \tag{2.15}$$

with the squared Euclidean distance between two features

$$d_{ij}^2 = \|\mathbf{x}_{1,i} - \mathbf{x}_{2,j}\|^2 . \tag{2.16}$$

The element G_{ij} in the proximity matrix records the attraction between the ith feature in image I_1 and the jth feature in image I_2. By singular value analysis of $\mathbf{G}$ a new matrix $\mathbf{P}$ is computed where the maximum items in both, rows and columns define the matching pairs. The mapping becomes ambiguous if an item has maximum value in its row or its column only. The parameter σ plays an important role and must be chosen large enough in order to consider the variance of the distance of corresponding points. The weakness of the algorithm against large rotations has been overcome by [Shapiro & Brady, 1992] using a shape description of the point pattern.

Relaxation

Relaxation labeling algorithms have been applied to PPM by [Ranade & Rosenfeld, 1980; Ogawa, 1984; Ton & Jain, 1989]. For each assumed point correspondence the displacement and a merit score are registered. The merit score is iteratively updated according to how closely other pairs would match given the displacement. The algorithm converges when the merit values become consistent or hardly change. The point mappings with maximum value are considered as the true point correspondences. However, these methods do in general do not impose the one-to-one mapping.

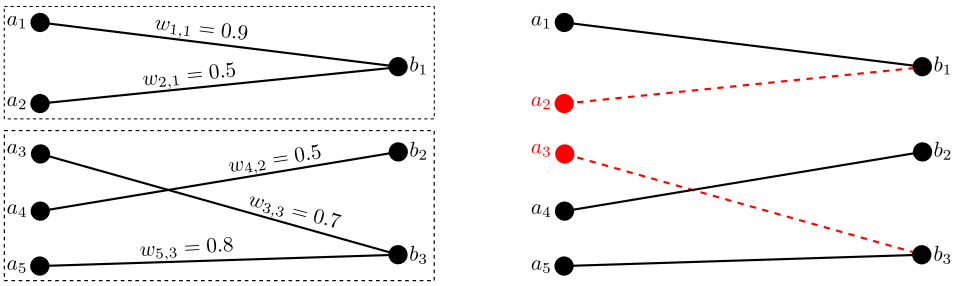
Graph matching

The PPM problem can be reduced to graph matching if a graph representation of the point pattern is available. The authors of a review on application of graph-matching in computer vision distinguish between exact and inexact graph-matching [Conte et al., 2007]. Most exact graph matching algorithms are based on some form of tree search with backtracking. Important algorithm of this family are due to [Ullmann, 1976; Cordella et al., 2004]. Exact graph matching algorithms are usually infeasible in practical matching due to deformed and incomplete data.

Inexact graph-matching algorithms often cast the inherently discrete optimization problem so as to use one of the many continuous non-linear optimization algorithms. Many of these algorithms are closely related to the previously listed approaches for point pattern matching. The relaxation labeling algorithm has been embedded into a theoretically motivated probabilistic framework by [Kittler & Hancock, 1989; Christmas et al., 1995]. Spectral methods are based on the invariance of the eigenvalues and eigenvectors of the adjacency matrix of isomorphic graphs [Umeyama, 1988; Carcassoni & Hancock, 2003]. Deformable models have been used to simultaneously find the correspondence and estimate the transformation underlying the point patterns. [Sclaroff & Pentland, 1995] use the eigenmodes of a finite element model and match feature points in a deformation invariant coordinate system. Thin-plate splines have been conducted by [Chui & Rangarajan, 2000; Rangarajan et al., 2001].

2.3.2 Bipartite Matching

The previous section has shown that the output of the PPM algorithms may be ambiguous and a unique one-to-one correspondence may not always be given. Bipartite matching can be applied to the output of the above algorithms if one seeks for a optimal one-to-one correspondence mapping. Given an initial set of potential correspondences, the PPM problem can be efficiently solved by reducing it to the assignment problem. Let $\mathcal{A} = \{a_i\}_{i=1}^{m}$ and $\mathcal{B} = \{b_j\}_{j=1}^{n}$ each be the abstract set of features detected in images $I_{\mathcal{A}}$ and $I_{\mathcal{B}}$, respectively. The graph representation $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of the



(a) Weighted bipartite graph

(b) Matching

Figure 2.6: (a) Weighted bipartite graph and (b) the maximum weighted bipartite matching.

stereo matching problem can be derived by regarding features as vertices $\mathcal{V} = \mathcal{A} \cup \mathcal{B}$ and feature correspondences as edges $\mathcal{E} \subseteq \mathcal{A} \times \mathcal{B}$. A *bipartite graph* is a graph whose vertices can be divided into two disjoint sets such that every edge connects a vertex of one set with one of the other set. By definition, $\mathcal{G}$ is a bipartite graph with disjoint sets $\mathcal{A}$ and $\mathcal{B}$. A bipartite graph is complete if $\mathcal{E} = \mathcal{A} \times \mathcal{B}$. A *weighted bipartite graph* is a bipartite graph with weights $w \in [0, 1]$ assigned to the edges. Examples for both, an unweighted and a weighted bipartite graph are shown in Figure 2.6.

The graph theoretic term of a *matching* refers to a subset $\mathcal{E}_M \subseteq \mathcal{E}$ such that no two edges in $\mathcal{E}_M$ share a common vertex. A *perfect matching* is a matching that matches all vertices of the graph and a *maximum weighted bipartite matching* is a perfect matching where the sum of weights of edges in the matching is maximal. If a bipartite graph is not complete, missing edges are inserted with weight zero. If $m \neq n$ additional vertices and all necessary edges are inserted too.

The task of finding a maximum weighted bipartite matching is known as the assignment problem and arises in stereo matching when candidates for possible point correspondences are given with a probability or quality measure. The Hungarian algorithm [Kuhn, 1955] was the first solution to the assignment problem and runs in $O\left(|\mathcal{V}|^2 |\mathcal{E}|\right)$. The assignment problem can be reduced to the shortest path problem which can be solved by Dijkstra's algorithm or the Bellman-Ford algorithm. For details on these algorithms the reader is referred to the text books [Papadimitriou & Steiglitz, 1998; Cormen et al., 1990]. Weighted bipartite matching has been successfully applied to feature matching using the Hungarian algorithm by [Griffin & Alexopoulos, 1989; Wu & Leou, 1995; Fielding & Kam, 1997].

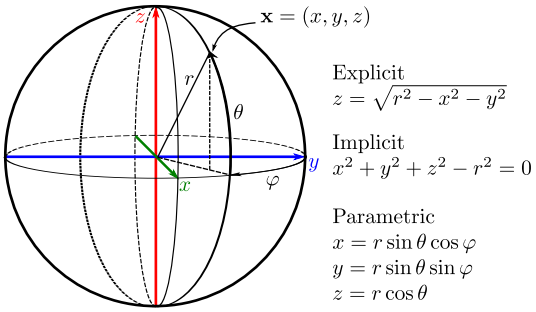


Figure 2.7: Sphere in explicit, implicit and parametric form

2.4 Free-Form Object Representation

The definition of free-form objects is more intuitive rather than formal. [Besl, 1990] defines a *free-form surface* as an object that "has a well defined surface normal that is continuous almost everywhere except at vertices, edges and cusps". This definition can be extended to apply for objects of any dimension, particularly for curves. The representation of *free-form* objects by computer models is a key element of many applications involving graphics and visualization. These models are either reverse engineered from point clouds using modern scanning technologies or designed through an elaborate CAD process [Campbell & Flynn, 2001]. Following [Brown, 1981], some important properties of object representation are ambiguity, conciseness and uniqueness. *Ambiguity* measures the ability of the representational scheme to completely define the object in the model space and is sometimes referred to as *completeness*. *Conciseness* represents how efficient the object description is and *uniqueness* is used to measure if there are more than one way to represent the same object given the construction methods of representation. The importance of the properties depends on the particular application considering the performance of the algorithms. Efficiency aspects often motivate the use of *discriminatory* instead of complete models since they capture only those details needed to distinguish objects from each other efficiently.

A selection of free-form object representation techniques is presented in the following paragraphs. The focus is on parametric models eligible for deriving a reasonable global shape of a smooth object from sparse unorganized point clouds which have been sampled from the object's surface. The introduction of the fundamental mathematical concepts in Section 2.4.1 is followed by an outline of a few commonly used curve and surface representation techniques in Section 2.4.2, Section 2.4.3, and Section 2.4.4. Two special classes of surfaces which can be represented by a curve underlying a motion are introduced in Section 2.4.5.

Description	Curve in $\mathbb{R}^2$	Surface in $\mathbb{R}^3$
Explicit	$y = f(x)$	$z = f(x, y)$
Implicit	$f(x, y) = 0$	$f(x, y, z) = 0$
Parametric	$\mathcal{C}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$	$\mathcal{S}(s, t) = \begin{bmatrix} x(s, t) \\ y(s, t) \\ z(s, t) \end{bmatrix}$

Table 2.1: Representation of free-form curves and surfaces

2.4.1 Mathematical Description

Free-form objects can be analytically described by curves and surfaces in explicit, implicit and parametric form [Bronstein et al., 1999]. The representation of curves and surfaces in 2-D and 3-D space is shown in Table 2.1 [Ahn, 2004]. For some objects an explicit, an implicit and a parametric form coexist whereas other objects can be represented only in parametric form. A sphere and its equations for the explicit, implicit and parametric representation is shown in Figure 2.7.

Implicit Form

An *implicit form* is the zero set $\mathcal{S} = \{\mathbf{x} : f(\mathbf{x}) = \mathbf{0}\}$ of an arbitrary smooth function $f : \mathbb{R}^n \to \mathbb{R}^k$ with $\mathbf{x}$ is a point in Euclidean space $\mathbb{R}^n$. An *implicit surface* in 3-D space is the zero set of $f : \mathbb{R}^3 \to \mathbb{R}$ and an *implicit curve* in the plane is the zero set of a function $f : \mathbb{R}^2 \to \mathbb{R}$. The curves or surfaces are called *algebraic* if the functions are polynomials [Taubin et al., 1992]. The special properties of algebraic curves and surfaces make them attractive to many applications.

Explicit Form

The *explicit form* describes one coordinate as a function of the other coordinates as given in the first row of Table 2.1. The explicit form can be easily converted to both, the implicit and the parametric form. Due to the fact that the explicit form is axis-dependent and single valued, many objects (e.g. a circle in $\mathbb{R}^2$ or a sphere in $\mathbb{R}^3$) cannot be described. Therefore, the use of the explicit form is limited in applications.

Parametric Form

The parametric form is the most general representation of free-form curves and surfaces. In terms of manifolds, a *regular parametrization* is the inclusion map $f : U \rightarrow \mathbb{R}^n$ of a k-dimensional *regular submanifold*[1] $\mathcal{S} \subseteq \mathbb{R}^n$ ($k \leqslant n$) in the Euclidean space $\mathbb{R}^n$ for some connected open set $U \subseteq \mathbb{R}^k$ [Bolle & Vemuri, 1991; Velho et al., 2002].

A vector $\mathbf{x} \in \mathbb{R}^n$ whose coordinate's depend on a parameter t traces out a *parametric curve*. Similarly, $\mathbf{x}$ traces out a *parametric surface* if it depends on two parameters s and t. A parametric curve in the plane ($k = 1$ and $n = 2$) as well as a parametric surface in 3-D space ($k = 2$ and $n = 3$) are shown in the last row of Table 2.1.

2.4.2 B-Spline Curves and Surfaces

Polynomials are attractive as they give a smooth representation for a discrete set of points and there exists a unique polynomial of degree $n - 1$ or less which passes through any ordered set of n points. The drawbacks are the nonintuitive variation of the fitting curve between the interpolating points and at the borders with variation of a single point. Also, an increased numerical instability is observed with a higher degree.

B-splines share the nice properties of polynomials and at the same time overcome their limitations. They are very commonly used for object representation in computer graphics. This section gives an outline of the mathematical basics and summarizes the most important properties. For a deeper insight the reader is referred to the text books on B-splines and their applications [de Boor, 2001; Hoschek et al., 1993; Piegl & Tiller, 1997; Farin, 2001; Prautzsch et al., 2002]. In general, splines are parametric curves defined as affine combination

$$\mathcal{C}(t) = \sum_{i=0}^{n} \mathbf{c}_i f_i(t) \tag{2.17}$$

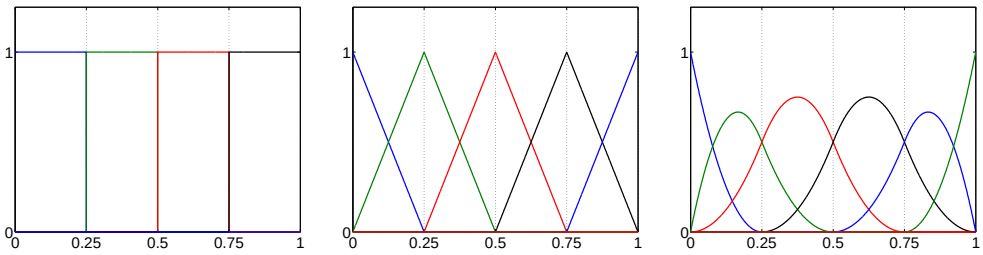
of some *control points* $\{\mathbf{c}_i\}_{i=0}^n$ and *piecewise polynomials* $\{f_i(t)\}_{i=0}^n$ of desired degree and continuity. Following the definition of [Prautzsch et al., 2002], a curve $\mathcal{C}(t)$ is called a spline of *degree* p with *knot vector*

$$T = [t_0, \ldots, t_m],$$

where $t_i \leqslant t_{i+1}$ and $t_i < t_{i+p+1}$ for all possible i, if $\mathbf{x}(t)$ is $p-k$ times differentiable at any k-fold knot[2], and $\mathbf{x}(t)$ is a polynomial of degree $\leqslant p$ over each knot interval $[t_i, t_{i+1}]$, for $i = 0, \ldots, m-1$. A spline of degree p is also referred to as spline of *order* $p + 1$.

[1] f is a smooth embedding and the derivative of f is everywhere injective

[2] A knot t_{i+1} is called k-fold if $t_i < t_{i+1} = \ldots = t_{i+k} < t_{i+k+1}$.



(a) $p = 0$ (b) $p = 1$ (c) $p = 2$

Figure 2.8: B-spline basis functions of different degree defined over the knot vector $T = [0, ..., 0, 0.25, 0.5, 0.75, 1, ..., 1]$ with $(p + 1)$-fold knots at its beginning and its end.

B-Spline Basis Functions

B-splines are splines in terms of the definition above and form a basis for the vector space of all piecewise polynomial functions. The name of B-splines has been introduced by [Schoenberg, 1967] and the following recurrence formula, most useful for computer implementation, is due to [de Boor, 1972]. The ith *B-spline basis function* of degree p, denoted by $N_{i,p}$, over the non-decreasing knot vector

$$T = [t_0, \ldots, t_m], \; t_i \leqslant t_{i+1} \text{ and } t_i < t_{i+p+1}$$

is defined as

$$N_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \leqslant t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t) \tag{2.19}$$

The shapes of the basis functions are determined entirely by the relative spacing of the knots. Scaling ($t'_i = \alpha t_i, \forall i$) or translating ($t'_i = t_i + \Delta t, \forall i$) the knot vector has no effect on the shapes of the $N_{i,p}$. The quotient of Equation (2.19) may yield $0/0$ for k-fold knots with $k > 1$. In that case, the quotient is defined to be zero. The shape of B-spline basis functions of different degree is visualized in Figure 2.8. The basis functions of lowest possible degree $p = 0$ are local constants and the number of basis functions grows with the degree due to the repetition of the first and last item in the knot vector.

Knot Vectors

The knot vector of a B-spline is given in general by

$$T = [t_0, \ldots, t_m] .$$

Since the knot vector determines the shape of the B-spline basis functions, knot vectors of particular form are referred to by special terms. A knot vector is called *uniform* (or *periodic*) if all knots are equally spaced, i.e. $t_{i+1} - t_i = \text{constant}, \forall i$. The basis functions on a uniform knot vector are simply shifted versions of one another. The knot vector is called *clamped* (or *nonperiodic*), if it has the form

$$T = [\underbrace{a, \ldots, a}_{p+1}, t_{p+1}, \ldots, t_{m-p-1}, \underbrace{b, \ldots, b}_{p+1}]$$

and the knots $t_p, \ldots, t_{m-p}$ are called *interior knots*. All knot vectors that are not clamped are called *unclamped*. A clamped knot vector is called uniform if its interior knots are equally spaced. All knot vectors that are not uniform are called *non-uniform*.

B-Spline Curves

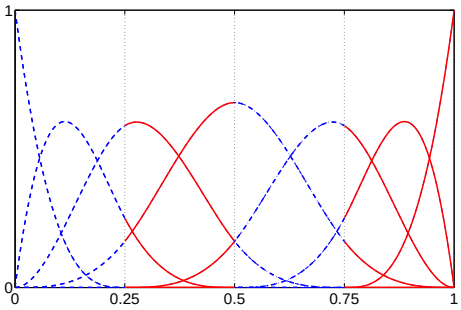
A pth-degree *B-spline curve* is defined by

$$\mathcal{C}(t) = \sum_{i=0}^{n} N_{i,p}(t)\, \mathbf{c}_i, \qquad a \leqslant t \leqslant b \tag{2.20}$$

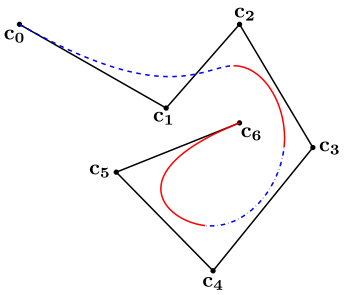
with control points $\{\mathbf{c}_i\}_{i=0}^{n}$ and B-spline basis functions $N_{i,p}$ of degree p defined over the knot vector

$$T = [a = t_0, \ldots, t_m = b] .$$

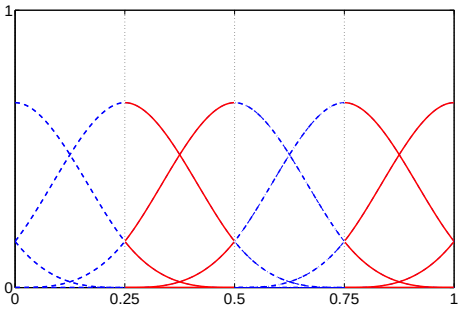
The polygon formed by the control points is called the *control polygon*. The shape of the B-spline curves depends on the knot vector and the control points. The terms defined above for the knot vectors are easily understood when looking at the B-spline curves formed by it. The end points of a B-spline curve over a clamped knot vector are identical with the first and the last control point, i.e. the curve is clamped. This is not the case for B-spline curves with unclamped knot vectors. A B-spline curve of given degree p over an unclamped and uniform knot vector ties a closed loop with $\mathcal{C}^{p-1}$ continuity if the last p control points are a repetition of the first p control points. Such a curve is called *cyclic*. All curves that are not closed are called open. The cubic B-spline basis functions ($p = 3$) with clamped and unclamped uniform knot vectors and examples for each, a clamped open, an unclamped open and a closed B-spline curve are shown in Figure 2.9.



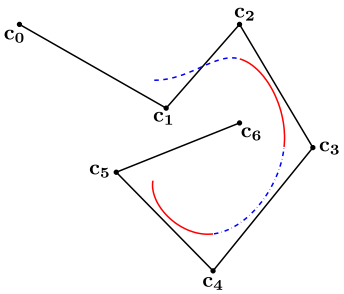
(a) Clamped uniform knot vector



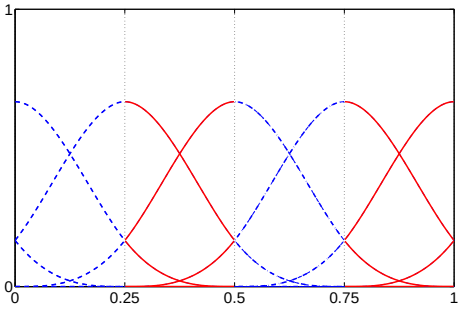
(b) Open B-spline curve



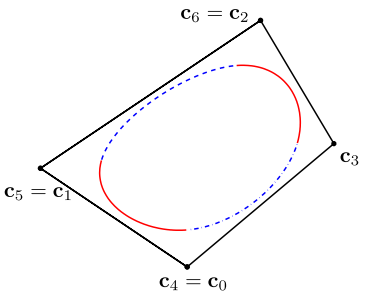
(c) Unclamped uniform knot vector



(d) Open B-spline curve



(e) Unclamped uniform knot vector



(f) Cyclic B-spline curve

Figure 2.9: Cubic B-spline basis functions for different knot vectors and example B-spline curves. The knot vectors in the left column together with the control polygon yield the B-spline curve in the right column. Corresponding B-spline basis functions and intervals of the B-spline curve are plotted in the same color and style.

B-Spline Properties

In the following list a choice of basic properties of B-splines are summarized.

- A B-spline curve with clamped knot vector interpolates the first and last control point: $\mathcal{C}(0) = \mathbf{c}_0$ and $\mathcal{C}(1) = \mathbf{c}_L$.

- B-splines have *local support*

  $$\operatorname{supp} N_{i,p} = [t_i, t_{i+p+1}]$$

  and are positive over the interior of their support

  $$N_{i,p}(t) > 0 \qquad \text{for } t \in (t_i, t_{i+p+1}).$$

  Thus, moving a control point $\mathbf{c}_i$ of a B-spline curve, affects the curve shape only on the subintervals (t_i, t_{i+p+1}).

- B-splines of degree p with a given knot sequence that do not vanish over some knot interval are *linearly independent* over this interval.

- The degree p, the number of control points $n+1$, and the number of knots $m+1$ are related by the equation

  $$m = n + p + 1.$$

- The control polygon represents a piecewise linear approximation to the curve. In general, the lower the degree, the closer a B-spline curve follows its control polygon.

- An affine transformation is applied to a B-spline curve by applying it to the control points.

- The *derivative* of a B-spline basis function is given by

  $$\frac{d}{dt} N_{i,p}(t) = \frac{p}{t_{i+p} - t_i} N_{i,p-1}(t) - \frac{p}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t) \qquad (2.21)$$

- Any segment $[\mathcal{C}(t_i), \mathcal{C}(t_{i+1}))$ of a pth degree B-spline lies within the convex hull of its $p+1$ control points $\mathbf{c}_{i-p}, \ldots, \mathbf{c}_i$

B-Spline Surfaces

A B-spline surface is obtained by taking a bidirectional net of control points $\mathbf{c_{i,j}}$, two knot vectors

$$S = [s_0, \ldots, s_{m+p+1}] \quad \text{and} \quad T = [t_0, \ldots, t_{n+q+1}],$$

and the products of the univariate B-spline basis functions

$$\mathcal{S}(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(s)\, N_{j,q}(t)\, \mathbf{c}_{i,j}. \tag{2.22}$$

A curve on $\mathcal{S}$ given by $\mathcal{S}(s,t)$ for fixed s or fixed t is called an *isoparametric curve*. B-spline surfaces can be open in both directions, closed in one direction, or closed in both directions, depending on the knot vectors and control points.

2.4.3 Orthogonal Functions

Orthogonal functions provide a simple means of parameter estimation for equally distributed data and may thus be used for object representation. The *inner product* in the vector space of complex or real valued square-integrable functions $\mathbf{\Phi}(I) = \{f : I \to \mathbb{F}\}$, with $\mathbb{F} = \mathbb{C}$ or $\mathbb{F} = \mathbb{R}$, on an interval $I = [a,b]$ is defined by

$$\langle f, g \rangle = \int_a^b \bar{f}(x)\, g(x)\, dx. \tag{2.23}$$

for $f, g \in \mathbf{\Phi}(I)$ and $\bar{f}$ is the complex conjugation of f, ignored for real valued functions. The inner product with a *weight function* $w(x)$ is defined by

$$\langle f, g \rangle_w = \int_a^b \bar{f}(x)\, g(x)\, w(x)\, dx. \tag{2.24}$$

The latter notation will be used as the first definition is only a special case of the latter for $w(x) \equiv 1$. The functions f and g are called *orthogonal* with respect to the weight function w, if

$$\langle f, g \rangle_w = 0. \tag{2.25}$$

If additionally

$$\langle f, f \rangle_w = 1, \qquad \text{and} \qquad \langle g, g \rangle_w = 1 \tag{2.26}$$

the functions are called *orthonormal*.

A system of pairwise orthogonal functions $\{\phi_n(x)\}$ is called an *orthogonal system*. It is called an *orthonormal system* if all functions have *length* $\sqrt{\langle \phi_n, \phi_n \rangle_w} = 1$, $\forall n$. The system is *complete* in I, if for an arbitrary function $f \in \mathbf{\Phi}(I)$ some $\{\gamma_n\}$ exist such that the *minimum square error* converges to zero as n becomes infinite:

$$\lim_{N\to\infty} \int_a^b \left[f(x) - \sum_{n=1}^{N} \gamma_n \phi_n(x) \right]^2 w(x)\, dx = 0. \tag{2.27}$$

Equation (2.27) takes its minimum for $\gamma_n = \langle \phi_n, f \rangle_w$, $\forall n$. A complete orthogonal system is a basis of $\mathbf{\Phi}(I)$ and the expression of a square-integrable function $f \in \mathbf{\Phi}(I)$ by this basis $\{\phi_n(x)\}$

$$f(x) \cong \sum_{n=1}^{\infty} \gamma_n \phi_n(x) \tag{2.28}$$

is called the *generalized Fourier series* with *Fourier coefficients*

$$\gamma_n = \frac{\langle \phi_n, f \rangle_w}{\langle \phi_n, \phi_n \rangle_w}. \tag{2.29}$$

It is desirable to express the shape of an object, given by a real or complex valued function f on an interval I, in terms of the generalized Fourier series expansion of a complete orthogonal system in I. The parameters that represent the shape are the coefficients γ_n independently computed by the inner product given in Equation (2.29).

In real applications the function f is usually a discrete signal. With equally spaced samples at points $\{t_k\}_{k=1}^{N}$, the coefficients can be computed by

$$\gamma_n = \frac{1}{N} \sum_{k=1}^{N} f(t_k)\, \phi_n(t_k) \tag{2.30}$$

A drawback of the orthogonal functions for shape representation is their limitation to equally spaced samples of the objects boundary. This is usually not satisfied by scattered data, e.g. acquired by optical 3D scanners. The coefficients of the generalized Fourier expansion does not give an insight into the geometry of the object represented. Local shape variations and discontinuities are hardly represented and their approximation involves basis functions of high degree.

Fourier Descriptors

Fourier descriptors are a parametric representation of a cyclic or periodic boundary of two-dimensional shape. A boundary curve, represented in parametric form

$$\mathcal{C}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \tag{2.31}$$

of a continuous parameter t, is combined into a curve with the complex function $z(t) = x(t) + iy(t)$. The coefficients of the *Fourier series expansion*

$$\hat{z}_n = \frac{1}{T} \int_0^T z(t)\, e^{-\frac{2\pi i n t}{T}}\, dt \tag{2.32}$$

with T the cycle period and $n \in \mathbb{Z}$ are also known as *Fourier descriptors*. In computer vision, one usually deals with discretized signals. The Fourier descriptors of a sampled boundary $\{z_n\}_{k=1}^{N}$ can be computed by the discrete Fourier transform

$$\hat{z}_n = \frac{1}{N} \sum_{k=0}^{N-1} z_k e^{-\frac{2\pi i k n}{N}} \tag{2.33}$$

with coefficients $\{\hat{z}_n\}_{n=0}^{N-1}$. The boundary is reconstructed by the inverse transformation

$$z(t) = \sum_{n=0}^{N-1} \hat{z}_n e^{\frac{2\pi i n t}{T}}. \tag{2.34}$$

Typically, the boundary points are parameterized by the arc length or by the angle between the radius drawn from the objects centroid to a point on the boundary and the x-axis. For the use of the *arc length parametrization* the boundary is sampled at equal distances P/N with P the perimeter of the curve. The *angle parametrization* is the representation of the boundary as a function of the angle with $P = 2\pi$ and samples $\{z_n\}_{n=0}^{N-1}$ at angles $2\pi n/N$. This parametrization limits the application to boundaries that are radial deformations of a circle. Scale and rotation invariant shape parameters can be derived from Fourier descriptors by scale normalization and phase shift [Jähne, 2002].

Two-dimensional Fourier descriptors have been applied to airplane silhouette classification [Arbter et al., 1990], image segmentation [Staib & Duncan, 1992], character recognition [Granlund, 1972] and content based image retrieval [Zhang & Lu, 2002; Folkers & Samet, 2002]. The use of Fourier descriptors for three-dimensional object representation has been suggested by [Lin & Jungthirapanich, 1990; Wu & Sheu, 1998].

Polynomials	Equation	Weight Function	Interval
Chebyshev	$T_n(x) = \cos(n \arccos(x))$	$\frac{1}{\sqrt{1-x^2}}$	$[-1, 1]$
Laguerre	$L_n(x) = e^x \frac{d^n}{dx^n}(x^n e^{-x})$	e^{-x}	$[0, \infty)$
Hermite	$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}\left(e^{-x^2}\right)$	e^{-x^2}	$(-\infty, \infty)$
Legendre	$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n}\left(\left(x^2-1\right)^n\right)$	1	$[-1, 1]$

Table 2.2: Orthogonal polynomials

Orthogonal Polynomials

Orthogonal polynomials are a class of orthogonal systems with real valued polynomial functions of a single variable x defined over an interval $I = [a, b]$. The formulas, definition intervals and the weight function of four popular orthogonal polynomial systems are listed in Table 2.2. Further polynomials can be found in [Abramowitz & Stegun, 1965].

2.4.4 Superellipses and Superquadrics

Superellipses are special cases of *Lamé curves* that are defined by the equation

$$\left(\frac{x}{a}\right)^m + \left(\frac{y}{b}\right)^m = 1 \tag{2.35}$$

where $a > 0$ and $b > 0$ are real numbers and m is any rational number. In the case of superellipses the exponent m is given by

$$m = \frac{2}{\epsilon} > 0, \qquad \text{where } \epsilon \in \mathbb{R} \tag{2.36}$$

and a and b are the size of the major and minor axes. For $\epsilon = a = b = 1$ this is the equation of a unit circle. For $m \to 0$ the curve takes up the shape of a cross and for $m \to \infty$ the shape of a rectangle [Jaklic et al., 2000]. Analogous to a circle a superellipse can be written in parametric form

$$\mathcal{C}(\theta) = \begin{bmatrix} a\cos^{\epsilon}\theta \\ b\sin^{\epsilon}\theta \end{bmatrix}, -\pi \leqslant \theta \leqslant \pi \tag{2.37}$$

where the exponentiation with ϵ is a signed power function $\cos^{\epsilon}\theta = \text{sign}\{\cos\theta\}\,|\cos\theta|^{\epsilon}$. The term of a *superquadric* has been defined by [Barr, 1981] and comprises a family of shapes including superellipsoids, superhyperboloids and supertoroids. *Superquadrics* can be obtained by the *spherical product*

$$\mathcal{S}(\theta,\varphi) = \mathcal{C}_m(\theta) \otimes \mathcal{C}_h(\varphi) = \begin{bmatrix} m_1(\theta)\,h_1(\varphi) \\ m_1(\theta)\,h_2(\varphi) \\ m_2(\theta) \end{bmatrix} \tag{2.38}$$

of two 2-D curves

$$\mathcal{C}_h(\varphi) = \begin{bmatrix} h_1(\varphi) \\ h_2(\varphi) \end{bmatrix} \quad \text{and} \quad \mathcal{C}_m(\theta) = \begin{bmatrix} m_1(\theta) \\ m_2(\theta) \end{bmatrix} \tag{2.39}$$

defined in the intervals $\varphi_0 \leqslant \varphi \leqslant \varphi_1$ and $\theta_0 \leqslant \theta \leqslant \eta_1$. Geometrically, $\mathcal{C}_h$ is a horizontal curve vertically modulated by $\mathbf{C}_m$. *Superellipsoids* are the spherical product of two superellipses

$$\mathbf{S}(\theta,\varphi) = \mathbf{C}_1(\theta) \otimes \mathbf{C}_2(\varphi) \tag{2.40}$$

$$= \begin{bmatrix} a_1\cos^{\epsilon_1}\theta \\ b_1\sin^{\epsilon_1}\theta \end{bmatrix} \otimes \begin{bmatrix} a_2\cos^{\epsilon_2}\varphi \\ b_2\sin^{\epsilon_2}\varphi \end{bmatrix} \tag{2.41}$$

$$= \begin{bmatrix} a_x\cos^{\epsilon_1}\theta\cos^{\epsilon_2}\varphi \\ a_y\cos^{\epsilon_1}\theta\sin^{\epsilon_2}\varphi \\ a_z\sin^{\epsilon_1}\theta \end{bmatrix}, \quad \begin{matrix} -\pi/2 \leqslant \theta \leqslant \pi/2, \\ -\pi \leqslant \varphi \leqslant \pi \end{matrix} \tag{2.42}$$

with $a_x = a_1 a_2$, $a_y = a_1 b_2$, and $a_z = b_1$ the scale factors along the three coordinate axes. The implicit form

$$F(x,y,z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}}. \tag{2.43}$$

can be derived using the equality $\cos^2 \alpha + \sin^2 \alpha = 1$ [Jaklic et al., 2000]. This equation is called the *inside-outside function* because it provides a simple test whether a given point (x, y, z) lies inside ($F(x, y, z) < 1$), outside ($F(x, y, z) > 1$) or on the surface ($F(x, y, z) = 1$) of the superquadric.

Supertoroids are defined by the equation

$$\mathbf{x}(\theta, \varphi) = \begin{bmatrix} a_x (r_0 + \cos^{\epsilon_1} \theta) \cos^{\epsilon_2} \varphi \\ a_y (r_0 + \cos^{\epsilon_1} \theta) \sin^{\epsilon_2} \varphi \\ a_z \sin^{\epsilon_1} \theta \end{bmatrix}, \quad \begin{matrix} -\pi \leqslant \theta \leqslant \pi, \\ -\pi \leqslant \varphi \leqslant \pi \end{matrix} \tag{2.44}$$

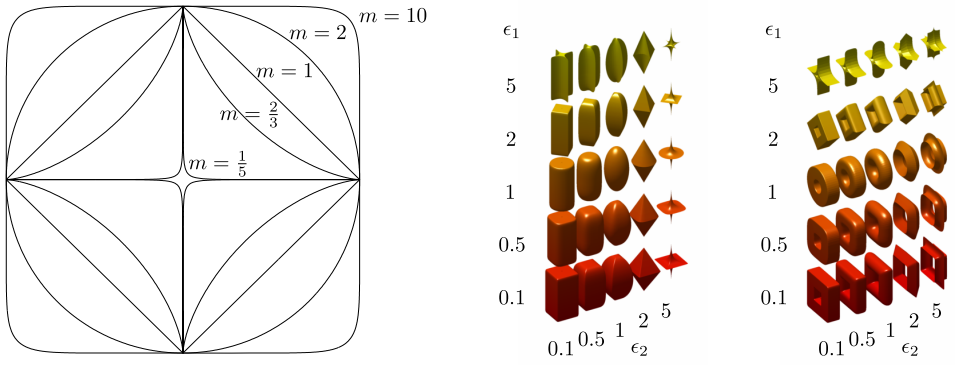
where r_0 is a positive real offset value which is related to the radius R of the supertoroid in the following way

$$r_0 = \frac{R}{\sqrt{a_x^2 + a_y^2}}. \tag{2.45}$$

The implicit form of the supertoroid is

$$F(x,y,z) = \left(\left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{2}} - r_0 \right)^{\frac{2}{\epsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}}. \tag{2.46}$$

In the computer vision literature, it is common to refer to superellipsoids by the more generic term of superquadrics. Superquadrics have been used to build complex objects by composite parts [Pentland, 1986; Gupta & Bajcsy, 1993], e.g. human body modeling in 3D motion tracking applications [Kehl & Gool, 2006]. By addition of tapering, twisting, and bending deformations to superquadrics a variety of free form objects can be modeled [Pentland, 1986; Solina & Bajcsy, 1990]. [Terzopoulos & Metaxas, 1991] used local finite element basis functions to adapt the model to local deformations. However, such models capture the fine detail that differentiates objects like human faces only with great difficulty. The ability of superquadrics to determine the coarse shape of an object has been used to determine geometric classes (*Geons*) [Raja & Jain, 1992; Dickinson et al., 1997]. Superellipses, superquadrics and supertoroids of different parameterization are shown in Figure 2.10.



(a) (b) (c)

Figure 2.10: (a) Superellipses, (b) superellipsoids and (c) supertoroids for some values of m, ϵ_1 and ϵ_2.

2.4.5 Common Surfaces

The term *common surfaces* refers to surfaces that can be obtained by translation or rotation of a planar curve in three-dimensional space. In particular, the swept surface and the surface of revolution are introduced.

Swept Surfaces

A *swept surface* is generated by sweeping an arbitrary *section curve* $\mathcal{C}_s$ along an arbitrary *trajectory curve* $\mathcal{C}_t$. The swept surface is given by the form

$$\mathcal{S}(s,t) = \mathcal{C}_t(t) + \mathbf{M}(t)\,\mathcal{C}_s(s) \tag{2.47}$$

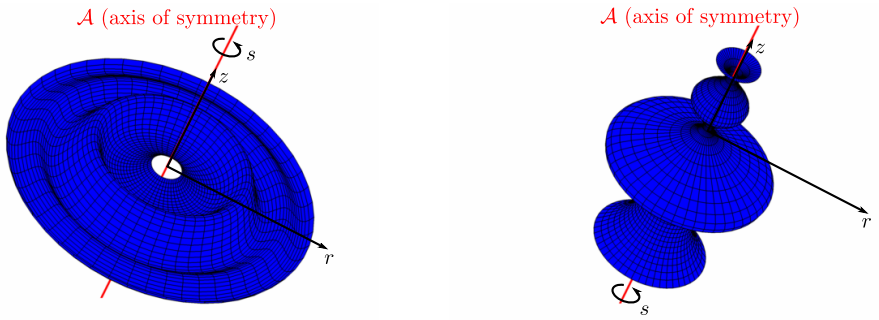
where $\mathbf{M}$ is a matrix incorporating rotation and scaling of $\mathcal{C}_s$ as a function of t. Two specific types of swept surfaces appear in practice:

1. $\mathbf{M}$ is the identity matrix
2. $\mathbf{M}$ is not the identity matrix.

The *generalized cylinders* is a special case, with $\mathcal{C}_t$ being a straight line and $\mathbf{M}$ the identity matrix.

Surface of Revolution

A surface of revolution (SoR) $\mathcal{S}$ is defined by an axis of symmetry $\mathcal{A}$ and a planar generating curve $\mathcal{C}$ rotated around $\mathcal{A}$. The planar curve $\mathcal{C}$ is called the *generatrix* and



(a) (b)

Figure 2.11: Surface of revolution with generatrix (a) orthogonal and (b) parallel to axis of symmetry

may be given in parametric form by

$$\mathcal{C}(t) = \begin{bmatrix} r(t) \\ h(t) \end{bmatrix}, \tag{2.48}$$

where r is the distance from the axis and h is the height relative to a reference along the axis. The pose of a surface of revolution has five degrees of freedom encoded in the axis of symmetry $\mathcal{A} = \{\mathbf{x}_C, \mathbf{a}\}$. The center $\mathbf{x}_C$ may be chosen arbitrarily along the axis as it only causes an offset of the generating curve. The direction $\mathbf{a}$ is uniquely defined by two parameters and with $\mathbf{e}_z = (0,0,1)^T$ and $|\mathbf{a}| = 1$, the rotation matrix representation of $\mathbf{a}$ is given by

$$\mathbf{R} = \mathbf{I}_{3\times 3} + 2\mathbf{e}_z\mathbf{a}^T - \frac{1}{1+\mathbf{a}^T\mathbf{e}_z}(\mathbf{a}+\mathbf{e}_z)(\mathbf{a}+\mathbf{e}_z)^T. \tag{2.49}$$

A SoR is thus given in parametric form by

$$\mathcal{S}(s,t) = \mathbf{R}\begin{bmatrix} r(t)\cos(s) \\ r(t)\sin(s) \\ h(t) \end{bmatrix} + \mathbf{x}_C \tag{2.50}$$

with $s \in [-\pi, \pi]$. Each isoparametric curve for fixed t traces out a circle of radius $r(t)$ and is called a *meridian*. A generalization of surfaces of revolution are *swung surfaces*, where the meridian is an arbitrary trajectory curve. Figure 2.11 shows two examples of surfaces of revolution with identical generatrix. Compared to Figure 2.11(a) the axis of symmetry in Figure 2.11(b) has been rotated by $90°$.

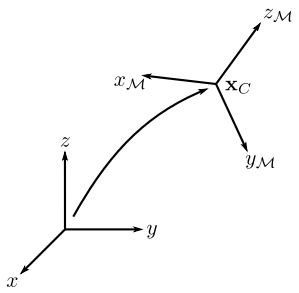


Figure 2.12: Machine and model coordinate system

2.5 Fitting Models to Point Clouds

Fitting a model of arbitrary shape and pose to an unorganized point cloud is an analytically and computationally complex problem that is often approached hierarchically by solving a sequence of less complex sub-problems. The first stages usually involve the identification of initial model parameters by appropriate direct methods which provide an approximation to the true values. The final result is usually found by means of an iterative non-linear optimization algorithm starting with the initial guess. This section gives an overview of the state of the art in parametric model fitting for scattered point clouds such as acquired by the 3-D measurement device introduced in Chapter 3. The focus is set particularly on surfaces of revolution and the related problems.

In Section 2.5.1 the terminology is introduced and the definition of a model is given. The mathematical fundamentals of model fitting are presented in Section 2.5.2 by a review of the linear and the non-linear least squares problem. An introduction to the general concept of a distance functions as error of fit measure in non-linear least squares and distance functions used in particular surface fitting problems are covered in Section 2.5.3. The closed form solutions to orthogonal distance fitting of lines and planes are introduced in Section 2.5.4 and a direct method for parameter estimating of a surface of revolution from surface normals is treated in Section 2.5.5. The problem of B-spline interpolation is discussed in Section 2.5.6 and the B-spline curve approximation in Section 2.5.7.

All the algorithms introduced in this section are inherently able to deal with equally distributed measurement errors such as random noise. In practice the observations may be subject to irregular noise and outliers due to errors in the data acquisition process. The concept of *robust statistics* provides fitting algorithms which are able to deal with such errors by construction of a framework on top of the methods presented in this section. A selection of the major algorithms developed in robust statistics are introduced in the next section.

2.5.1 Model Definition

In the context of the methods presented in this chapter, a *model* is defined as the composition $\mathcal{M} = \{\mathcal{S}, \mathcal{P}\}$ of a surface $\mathcal{S} \subset \mathbb{R}^3$ in the coordinate system defined by a *pose* $\mathcal{P} = \{\mathbf{R}, \mathbf{x}_C\}$. The coordinate system of the measurement device is called the *machine coordinate system* and the coordinate system defined by the pose is referred to as *model coordinate system*. The surface is represented by a parametric shape description, e.g. a B-spline surface. The pose is expressed in terms of a rotation matrix $\mathbf{R} = (\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z)^T$ and the *model center* $\mathbf{x}_C = (x_C, y_C, z_C)^T$, where the rotation matrix $\mathbf{R}$ and the translation $\mathbf{x}_C$ describe the transformation from the machine coordinate system to the model coordinate system. If necessary, constraints are introduced, such that $\mathcal{M}$ is described by a unique set of surface and pose parameters. Let $\mathbf{x} = (x, y, z)^T$, be a point in machine coordinates. A point $\mathbf{x}_\mathcal{M} = (x_\mathcal{M}, y_\mathcal{M}, z_\mathcal{M})^T$ in model coordinates is computed by applying the backward transformation

$$\mathbf{x}_\mathcal{M} = \mathbf{R}^{-1} \cdot (\mathbf{x} - \mathbf{x}_C) \,. \tag{2.51}$$

The process of fitting a model to a 3D point cloud involves the identification of the pose parameters $\mathcal{P}$ and the parameters that describe the surface $\mathcal{S}$ given a particular shape representation (see Section 2.4).

2.5.2 Least Squares

The method of *least squares*

$$\min_{\mathbf{p}} \|\mathbf{r}\|^2 \tag{2.52}$$

is utilized in model fitting for computing an approximation $\hat{\mathbf{p}}$ to the true parameters $\mathbf{p} = (p_1, \ldots, p_n)^T$ of a model $\mathcal{M}$ from perturbed observations $\{\mathbf{x}_i\}_{i=1}^m$ such that the squared Euclidean norm of the *residuals* $\mathbf{r} = (r_1, \ldots, r_m)^T$ is minimized. The residual r_i of an observation $\mathbf{x}_i$ with the model $\mathcal{M}$ is computed according to

$$r_i = d\,(\mathbf{p}, \mathbf{x}_i) \tag{2.53}$$

called the *distance measure*. Distance measures are introduced in general in Section 2.5.3 and presented in particular for B-spline approximation in Section 2.5.7. Depending on the distance measure and the scale of a least squares problem the solution can be found directly or an approximation is computed by an iterative algorithm.

Linear Least Squares

If the distance measure of a parameter fitting problem is an equation in explicit form which is linear in the parameters, the least squares problem is given by the form

$$\mathbf{X}\mathbf{p} = \mathbf{y}, \tag{2.54}$$

where the ith row in $\mathbf{X}$ and the ith element in $\mathbf{y}$ together make the measurement vector $\mathbf{x}_i$. Assuming that $\mathbf{X}$ has full column rank and the observations represent independent random draws from their population, the method of linear least squares leads to the unique solution

$$\hat{\mathbf{p}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}. \tag{2.55}$$

The solution can be found using Gauss elimination of $\mathbf{X}^T\mathbf{X}$ for well conditioned matrices. A numerically more stable and faster algorithm is the Cholesky decomposition [Golub & Loan, 1996]. For very large systems with sparse matrices the conjugate gradient method finds a well approximation very efficiently [Nocedal et al., 2000]. In linear least squares the *hat matrix* $\mathbf{H}$ is the matrix

$$\mathbf{H} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T \tag{2.56}$$

with $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$ are the predictors of the fitted model.

If the assumption of equal variance is violated the weighted residuals are minimized. The above expression is transformed into the weighted linear least squares formulation

$$\hat{\mathbf{p}} = \left(\mathbf{X}^T\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{W}\mathbf{y} \tag{2.57}$$

where $\mathbf{W}$ is a square diagonal weight matrix with ith element $w_{ii} = 1/\sigma_i^2$ the reciprocal of the variance of the ith observation. For correlated errors the weight matrix is the inverse of the covariance matrix of the observations.

For ill-posed problem, a solution is found introducing additional information to the linear least squares equation by means of a *regularization* term $\|\mathbf{R}\mathbf{p}\|^2$. The solution of a regularized linear least squares problem is given by

$$\hat{\mathbf{p}} = \left(\mathbf{X}^T\mathbf{X} + \mathbf{R}^T\mathbf{R}\right)^{-1}\mathbf{X}^T\mathbf{y}. \tag{2.58}$$

In statistics, this method is referred to as *ridge regression*.

A related problem is that of *total least squares* (TLS) where the orthogonal error of the observations with the model is minimized. The application of TLS to line and plane fitting is discussed in Section 2.5.4.

Non-Linear Least Squares

In non-linear least squares the error of fit function $d(\mathbf{x}_i, \mathbf{p})$ is non-linear in the model parameters $\mathbf{p}$ and an approximation to the solution is numerically found by a sequence of linear least squares steps. Starting with an initial estimate $\mathbf{p}^{(0)}$ of the parameters, the *Gauss-Newton algorithm* proceeds by iteratively computing a new estimate

$$\mathbf{p}^{(k)} = \mathbf{p}^{(k-1)} + \Delta\mathbf{p}, \qquad k > 0 \tag{2.59}$$

using the first-order Taylor series expansion of $\mathbf{r}$ about the previous estimate $\mathbf{p}^{(k-1)}$. The increment $\Delta\mathbf{p}$ is computed to satisfy the normal equations

$$\mathbf{J}^T\mathbf{J}\Delta\mathbf{p}^{(k-1)} = \mathbf{J}^T\mathbf{r}^{(k-1)}, \tag{2.60}$$

where $\mathbf{J}$ is the Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \partial d(x_1)/\partial p_1 & \dots & \partial d(x_1)/\partial p_n \\ \vdots & & \vdots \\ \partial d(x_m)/\partial p_1 & \dots & \partial d(x_m)/\partial p_n \end{bmatrix}. \tag{2.61}$$

This linear least squares problem is solved by means of the algorithms given in the previous paragraph. The iteration is aborted on some stopping criteria, e.g. the change in the parameters. If the Jacobian is rank-deficient or nearly so, a regularization term is introduced in order to change the direction of search. This leads to the *Levenberg-Marquardt algorithm* where Equation (2.60) is replaced by

$$\left(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{D}^T\mathbf{D}\right)\Delta\mathbf{p}^{(k-1)} = \mathbf{J}^T\mathbf{r}^{(k-1)}, \tag{2.62}$$

and $\mathbf{D}$ is a diagonal matrix taking into account of different scale in the observations. The value of the parameter λ in the algorithm is chosen such as to decrease the error of fit function.

In large scale problems the Jacobi matrix is often not available and a variant of the conjugate gradient method in combination with Newton line search or a trust region algorithm is used instead [Nocedal et al., 2000]. The result of any non-linear least squares algorithm significantly depends on the initial estimate and the eligibility of the local linear model. The algorithm may converge to an undesired local minimum if the initial estimate is too far away from the true solution or the error of fit function is not convex close to the solution.

2.5.3 Distance Measure

All iterative fitting algorithms use an *error of fit* measure F that is successively decreased until a minimum is found. The error of fit is computed from the distance of the data points and the underlying model. The *distance measure* $d(\mathbf{x})$ between a point $\mathbf{x}$ and the surface defined by a model $\mathcal{M}$ can be defined in various ways, depending on the representation of the surface. The distance is zero $d(\mathbf{x}) = 0$ if the point lies on the surface.

For curves and surfaces in implicit form $\mathcal{S} = \{\mathbf{x} : f(\mathbf{x}) = 0\}$, the *algebraic distance* is defined as the absolute function value $d_a(\mathbf{x}) = f(\mathbf{x})$. Model fitting using the algebraic distance as error of fit measure is called *algebraic fitting*. Unfortunately, the algebraic distance does not have an intrinsic geometric significance [Sullivan et al., 1994]. Using the algebraic distance for fitting, may introduce bias or artifacts such as holes or unbounded surface components [Gross & Boult, 1988; Taubin et al., 1992].

The *geometric distance* $d_o(\mathbf{x})$between a point $\mathbf{x}$ and a curve or surface $\mathcal{S}$ is defined as the minimum Euclidean distance between $\mathbf{x}$ and any of the points belonging to $\mathcal{S}$. Since connection of $\mathbf{x}$ and the point on $\mathcal{S}$ with minimum distance is the surface normal, the geometric distance is also known as *orthogonal distance*. Orthogonal distance fitting is the fitting of models using the orthogonal distance as error of fit measure. A drawback of the orthogonal distance measure is that in general it cannot be computed in closed form. The approximation

$$\hat{d}_o(\mathbf{x}) = \frac{f^2(\mathbf{x})}{\|\nabla f(\mathbf{x})\|^2} \tag{2.63}$$

to the orthogonal distance for implicit surfaces is due to [Sampson, 1982; Taubin, 1991]. Several approximations for orthogonal distance measures of superquadrics have been proposed by [Bajcsy & Solina, 1987; Gross & Boult, 1988; Gupta et al., 1989; Yokoya et al., 1992]. A good approximation to the orthogonal distance of a point $\mathbf{x}$ and a superquadric $\mathcal{S}$ is the distance between $\mathbf{x}$ and the point $\mathbf{x}_r$ on the surface of $\mathcal{S}$ in radial direction from the center of $\mathcal{S}$ to $\mathbf{x}$. The point $\mathbf{x}_o$ with minimal Euclidean distance can be determined in an iterative refinement procedure using $\mathbf{x}_r$ as initial value [Gupta et al., 1989].

2.5.4 Lines and Planes

Even though the orthogonal distance cannot in general be computed in closed form, orthogonal distance fitting remains a linear problem for lines and planes. Due to the fact that the solution can be found in closed form at low computation cost, parameters of complex objects should be initialized by the parameters obtained from fitting lines and planes where ever possible [Ahn, 2004].

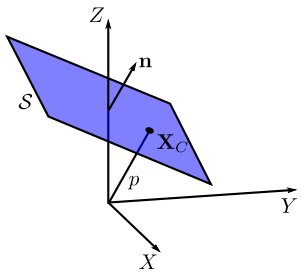


Figure 2.13: Plane

Planes

The orthogonal distance fit of a plane to points $\{\mathbf{x}_i\}_{i=1}^n$ in 3D space is equivalent to the determination of the two-dimensional subspace $\mathcal{S} \subset \mathbb{R}^3$, such that the variance of the orthogonal distance of all data points with $\mathcal{S}$ is minimal. Following the principal component analysis (PCA), the first and second principal component are a basis of the subspace that satisfies this condition (see e.g. [Bishop, 2006]). With

$$\mathbf{x}_C = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i, \tag{2.64}$$

the mean of the data set and

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{x}_C) (\mathbf{x}_i - \mathbf{x}_C)^T \tag{2.65}$$

the data covariance matrix, the first and the second principal components are the eigenvectors of Σ corresponding to the largest and second largest eigenvalues. It follows, that the normal vector of the plane in question is the third eigenvector of Σ corresponding to the smallest eigenvalue. The data mean $\mathbf{x}_C$ is regarded as the model center.

A plane is represented in Hessian normal form

$$\mathbf{n}^T \mathbf{x} + d = 0 \tag{2.66}$$

with $\mathbf{n}$, the normalized third principal component, is the unit normal vector of the plane ($\|\mathbf{n}\| = 1$), and d is the distance of the plane from the origin computed by

$$d = -\mathbf{n}^T \mathbf{x}_C. \tag{2.67}$$

Lines

Analogous to the plane, the orthogonal distance fit of a three-dimensional line to points in $\mathbb{R}^3$ can be found by the PCA. The line in question is the one-dimensional subspace $\mathcal{L} \subset \mathbb{R}^3$, such that the variance of the orthogonal distance of all data points $\{\mathbf{x}_i\}_{i=1}^n$ with $\mathcal{L}$ is minimal. The basis of this linear subspace is therefore the eigenvector of Σ corresponding to the largest eigenvalue.

A line is represented in parametric form

$$\mathbf{x}(t) = \mathbf{x}_0 + t \cdot \mathbf{l}, \tag{2.68}$$

with $t \in \mathbb{R}$ is a parameter, $\mathbf{x}_0 = \mathbf{x}_C$ is the data mean, and $\mathbf{l}$ is the first principal component, i.e. the direction of the line.

A straight line $\mathcal{L}$ can also be represented by a normalized direction vector $\mathbf{l}$, $\|\mathbf{l}\| = 1$ and the moment vector $\bar{\mathbf{l}} = \mathbf{x} \times \mathbf{l}$, with $\mathbf{x}$ is an arbitrary point on $\mathcal{L}$. The moment $\bar{\mathbf{l}}$ is independent of the choice of $\mathbf{x}$. The six coordinates $\left(\mathbf{l}, \bar{\mathbf{l}}\right)$ are called the *normalized Plücker coordinates* of $\mathcal{L}$. They satisfy the Plücker relation $\mathbf{l}^T\bar{\mathbf{l}} = 0$. Any 6-tupel $\left(\mathbf{l}, \bar{\mathbf{l}}\right) \in \mathbb{R}^6$ with $\|\mathbf{l}\| = 1$ and $\mathbf{l}^T\bar{\mathbf{l}} = 0$ represents a line, where $\left(\mathbf{l}, \bar{\mathbf{l}}\right)$ and $\left(-\mathbf{l}, -\bar{\mathbf{l}}\right)$ describe the same line.

2.5.5 Surface of Revolution

A special geometric property of a SoR is that all surface normals are contained in a linear complex [Pottmann & Wallner, 2001]. A linear complex $\mathcal{X} = (\mathbf{x}, \bar{\mathbf{x}})$ is the set of all lines $\mathcal{L} = \left(\mathbf{l}, \bar{\mathbf{l}}\right)$ whose Plücker coordinates satisfy the equation

$$\bar{\mathbf{x}}^T\mathbf{l} + \mathbf{x}^T\bar{\mathbf{l}} = 0. \tag{2.69}$$

In the case of a surface of revoltion, $\mathcal{X}$ represents the symmetric axis $\mathcal{A}$ in Plücker coodinates. The moment of a line $\mathcal{L} = \left(\mathbf{l}, \bar{\mathbf{l}}\right)$ with respect to the linear complex $\mathcal{X} = (\mathbf{x}, \bar{\mathbf{x}})$ is defined as

$$m\left(\mathcal{L}, \mathcal{X}\right) = \frac{1}{\|\mathbf{x}\|\,\|\mathbf{l}\|}\left|\bar{\mathbf{x}}^T\mathbf{l} + \mathbf{x}^T\bar{\mathbf{l}}\right|. \tag{2.70}$$

The moment becomes

$$m\left(\mathcal{L}, \mathcal{X}\right) = \left|\bar{\mathbf{x}} \cdot \mathbf{l} + \mathbf{x} \cdot \bar{\mathbf{l}}\right| \tag{2.71}$$

if normalized Plücker coordinates are used.

Let $\{\mathbf{x}_i\}_{i=1}^n$ be a sampling of a surface of revolution $\mathcal{S}$ and $\{\mathbf{n}_i\}_{i=1}^n$ the corresponding surface normals with $\|\mathbf{n}_i\| = 1$. With $\bar{\mathbf{n}}_i = \mathbf{n}_i \times \mathbf{x}_i$

$$\mathcal{N}_i = (\mathbf{n}_i, \bar{\mathbf{n}}_i) \tag{2.72}$$

are the lines in Plücker coordinates defined by the surface normals. The axis of a surface of revolution can be reconstructed from the $\mathcal{N}_i$ using (2.71). Therefore the sum of squared moments

$$F\left(\mathbf{x}, \bar{\mathbf{x}}\right) = \sum_{i=1}^{n} m\left(\mathcal{L}, \mathcal{X}\right)^2 = \sum_{i=1}^{n}\left(\bar{\mathbf{x}}^T\mathbf{n}_i + \mathbf{x}^T\bar{\mathbf{n}}_i\right)^2 \tag{2.73}$$

is used as a measure of deviation of a linear complex $\mathcal{X}$ from the set of lines $\{\mathcal{N}_i\}_{i=1}^n$. Since the product $(a \cdot b)\,(c \cdot d)$ is in matrix notation written as $b^T \cdot \left(a \cdot c^T\right) \cdot d$, F can be expressed in the form

$$F\left(\mathbf{x}, \bar{\mathbf{x}}\right) = \left(\mathbf{x}, \bar{\mathbf{x}}\right)^T \cdot \mathbf{M} \cdot \left(\mathbf{x}, \bar{\mathbf{x}}\right) \tag{2.74}$$

with

$$M = \sum_{i=1}^{n}\begin{bmatrix} \bar{\mathbf{n}}_i\bar{\mathbf{n}}_i^T & \mathbf{n}_i\bar{\mathbf{n}}_i^T \\ \\ \bar{\mathbf{n}}_i\mathbf{n}_i^T & \mathbf{n}_i\mathbf{n}_i^T \end{bmatrix}. \tag{2.75}$$

The linear complex that approximates the surface normals is computed looking for the normalized Plücker coordinates $(\mathbf{x}, \bar{\mathbf{x}})$, $\mathbf{x}^T\mathbf{x} = 1$ that minimize F. Since $\mathcal{X}$ itself represents a line, the minimization is subject to the side condition $\mathbf{x}^T\bar{\mathbf{x}} = 0$. Both constraints

$$G(\mathbf{x}, \bar{\mathbf{x}}) = \mathbf{x}^T\mathbf{x} = 1 \quad \text{and} \quad H(\mathbf{x}, \bar{\mathbf{x}}) = \mathbf{x}^T\bar{\mathbf{x}} = 0 \tag{2.76}$$

can be written in matrix notation by

$$G(\mathbf{x}, \bar{\mathbf{x}}) = (\mathbf{x}, \bar{\mathbf{x}})^T \cdot D \cdot (\mathbf{x}, \bar{\mathbf{x}}) = 1 \quad \text{and} \tag{2.77}$$

$$H(\mathbf{x}, \bar{\mathbf{x}}) = (\mathbf{x}, \bar{\mathbf{x}})^T \cdot K \cdot (\mathbf{x}, \bar{\mathbf{x}}) = 0 \tag{2.78}$$

with

$$D = \operatorname{diag}(1, 1, 1, 0, 0, 0) \quad \text{and} \quad K = \begin{bmatrix} 0 & \operatorname{diag}(1, 1, 1) \\ \operatorname{diag}(1, 1, 1) & 0 \end{bmatrix}$$

With the Lagrangian multipliers λ and μ the solution is found by solving

$$(\nabla F - \lambda \nabla G - \mu \nabla H)(\mathbf{x}, \bar{\mathbf{x}}) = 0,\ G(\mathbf{x}, \bar{\mathbf{x}}) = 1,\ H(\mathbf{x}, \bar{\mathbf{x}}) = 0. \tag{2.79}$$

This simplifies to

$$(M - \lambda D - \mu K) \cdot (\mathbf{x}, \bar{\mathbf{x}}) = \mathbf{0}, \quad \mathbf{x}^T\mathbf{x} = 1, \quad \mathbf{x}^T\bar{\mathbf{x}} = 0. \tag{2.80}$$

The vector $(\mathbf{x}, \bar{\mathbf{x}})$ is contained in the kernel of $M - \lambda D - \mu K$. A non-zero solution obviously is possible if and only if $M - \lambda D - \mu K$ is singular. This implies

$$\det(M - \lambda D - \mu K) = 0. \tag{2.81}$$

From Equation (2.80) it follows, that $F(\mathbf{x}, \bar{\mathbf{x}}) = \lambda$. This shows that in order to minimize F, a pair (λ, μ) with smallest possible λ has to be chosen. However, the solution of this problem is not straightforward and best computed numerically. An approximation to the symmetric axis is given by the linear complex $(\mathbf{x}, \bar{\mathbf{x}})$ that minimizes F, ignoring the side condition $\mathbf{x}^T\bar{\mathbf{x}} = 0$. Then, Equation (2.81) reduces to

$$\det(M - \lambda D) = 0, \tag{2.82}$$

that is a cubic equation in λ. The solution is the general eigenvector $(\mathbf{x}, \bar{\mathbf{x}})$, $\|\mathbf{x}\| = 1$ corresponding to the smallest root of Equation (2.82). This approximation makes sense only for $k > 5$, because there is a unique linear complex which contains five lines in general position and for $k < 5$ there is at least a one-parameter family of linear complexes which contains all given lines.

2.5.6 B-Spline Interpolation

First, the problem of computing a interpolating B-spline curve $\mathcal{C}$ of given degree p that passes through a given set of ordered points $\{\mathbf{x}_k\}_{k=0}^{n}$ is considered. Assigning a parameter value $\hat{t}_k$ to each $\mathbf{x}_k$ and selecting an appropriate knot vector

$$T = \{t_0, \ldots, t_m\},$$

leads to the $(n+1) \times (n+1)$ system

$$\mathbf{x}_k = \mathcal{C}\left(\hat{t}_k\right) = \sum_{i=0}^{n} N_{i,p}\left(\hat{t}_k\right) \mathbf{c}_i \tag{2.83}$$

of linear equations, where the $n+1$ control points $\mathbf{c}_i$ are the unknowns. Let r be the dimension of the data points. The linear system of Equation (2.83) has a single coefficient matrix and r right hand sides and r solution sets for the r coordinates of the $\mathbf{c}_i$.

Parametrization and Knot Vector Selection

The problem of choosing the $\hat{t}_k$ and the knot vector U remains, and their choice affects the shape and the parametrization of the curve. Throughout this section the parameter is assumed to lie in the range $t \in [0, 1]$. This is possible because scaling the knot vector does not affect the shape of the B-spline basis functions (see Section 2.4.2). The following four methods of choosing the $\hat{t}_k$ can be used [Hoschek et al., 1993; Piegl & Tiller, 1997].

- The *uniform* parametrization assigns equally spaced parameters

$$\hat{t}_0 = 0, \quad \hat{t}_n = 1, \quad \text{and} \quad \hat{t}_k = \frac{k}{n}, \quad k = 1, \ldots, n-1 \tag{2.84}$$

  to each point. This method is not recommended when the points are unevenly spaced, as it can produce unwanted loops or knobs.

- Let d_k be the distance $|\mathbf{x}_k - \mathbf{x}_{k-1}|$, $k = 1, \ldots, n$, with $d_0 = 0$, and the total length $s = \sum_{k=0}^{n} d_k$. The *chord length* parametrization assigns the parameters

$$\hat{t}_0 = 0, \quad \hat{t}_n = 1, \quad \hat{t}_k = \hat{t}_{k-1} + \frac{d_k}{s}, \quad k = 1, \ldots, n-1. \tag{2.85}$$

  This is the most widely used method. Geometrically, this parametrization can be interpreted as an approximation of the arc length.

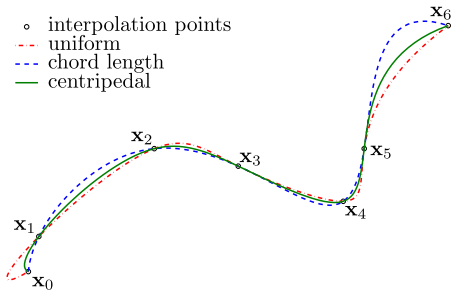


Figure 2.14: B-spline curve interpolation with different parametrizations

- The *centripetal* parametrization [Lee, 1989] works in analogy to the chord length parametrization, with $d_k = \sqrt{|\mathbf{x}_k - \mathbf{x}_{k-1}|}$ and s computed from these d_k. This method gives better results than the chord length method when the data takes sharp turns.

- The previous parametrization methods are invariant under rotations, translations and equal scaling in all coordinates. Motivated by character shape representation in two dimensions, [Foley & Nielson, 1989] introduced a parametrization method where the shape of the interpolated curve is consistent if the defining data points are rotated, scaled differently in x and y, translated, or sheared.

  The *affine invariant distance* of two points $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ is defined by

$$d_k = \sqrt{(\mathbf{x}_k - \mathbf{x}_{k-1}) \frac{\Sigma}{\det(\Sigma)} (\mathbf{x}_k - \mathbf{x}_{k-1})^T} \tag{2.86}$$

  with Σ is the covariance matrix of the points $\{\mathbf{x}_k\}_{k=1}^m$. The *affine invariant chord* parametrization is a generalization of the chord length parametrization using the affine invariant distance.

  The *affine invariant angle* parametrization also takes in account of the change of the angles between the interpolation points. The definition of this measure can be found in [Foley & Nielson, 1989].

- [Zhang et al., 1998] developed an algorithm for parameterization of data that has been taken from a parametric quadratic polynomial. The *new 2-spline* method performs better than the above if the convexity of the data points does not change sign.

The entries in the knot vector may be equally spaced

$$t_0 = \ldots = t_p = 0 \quad \text{and} \quad t_{m-p} = \ldots = t_m = 1, \quad \text{with}$$
$$t_{j+p} = \frac{j}{n-p+1} \qquad j = 1, \ldots, n-p. \tag{2.87}$$

This method is not recommended, as it may result in a singular system of equations if the *Schönberg-Withney condition* is violated. This condition is satisfied if there is at least one $\hat{t}_k$ in each knot span $[t_j, t_{j+1}]$.

[Piegl, 1991] recommends the following technique of *averaging*

$$t_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} \hat{t}_i \qquad j = 1, \ldots, n-p. \tag{2.88}$$

This method of selecting the knots leads to a coefficient matrix that is totally positive and banded with semi-bandwidth less than p, that is, $N_{i,p}\left(\hat{t}_k\right) = 0$, if $|i-k| \geqslant p$ [de Boor, 2001].

Surface Interpolation

In B-spline surface interpolation a (p, q)th-degree B-spline surface $\mathcal{S}$ is constructed, which passes through a grid of $(n+1) \times (m+1)$ data points $\{\mathbf{x}_{k,l}\}$, $k = 0, \ldots, n$ and $l = 0, \ldots, m$. First, reasonable values for the $\left(\hat{s}_k, \hat{t}_l\right)$ and the knot vectors

$$S = [s_0, \ldots, s_{n+p+1}] \quad \text{and} \quad T = [t_0, \ldots, t_{m+q+1}]$$

have to be found. The parameters $\hat{s}_k$ and $\hat{t}_l$ are computed by

$$\hat{s}_k = \frac{1}{m+1} \sum_{l=0}^{m} \hat{s}_k^l \quad \text{and} \quad \hat{t}_l = \frac{1}{n+1} \sum_{k=0}^{n} \hat{t}_l^k \tag{2.89}$$

where the $\hat{s}_k^l$ and the $\hat{t}_l^k$ are computed from the control points $\mathbf{c}_{k,l}$ with fixed l and fixed k, respectively, by Equation (2.85) using the Euclidean distance, the centripetal distance, or the affine invariant distance of consecutive control points. Once the $\left(\hat{s}_k, \hat{t}_l\right)$ are computed, the knot vectors S and T can be obtained by Equation (2.88). The system

$$\mathbf{x}_{k,l} = \mathcal{S}\left(\hat{s}_k, \hat{t}_l\right) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}\left(\hat{s}_k\right) N_{j,q}\left(\hat{t}_l\right) \mathbf{c}_{i,j} \tag{2.90}$$

with $(n+1) \times (m+1)$ linear equations in the unknown $\mathbf{c}_{i,j}$ can be arranged, such that the coefficient matrix has a banded structure. The solution can be computed efficiently as a sequence of $(n+1) + (m+1)$ curve interpolations [Piegl & Tiller, 1997].

2.5.7 B-Spline Approximation

In B-spline approximation a B-spline curve or surface is computed that approximates the given data points in the least squares sense, that is, an objective function on the distances between the data points and the B-spline curve or surface is minimized. In general, the resulting curve or surface does not pass precisely through the data points. An approximation problem is given when the number of data points is larger than the desired number of control points.

The focus in this section is on approximation of a set of given data points $\mathbf{X} = \{\mathbf{x}_k\}_{k=1}^m$ by a B-spline curve $\mathcal{C}$ of pth-degree with $n+1$ control points $\{\mathbf{c}_j\}_{j=0}^n$, data point parametrization $\hat{T} = \left\{\hat{t}_k\right\}_{k=1}^m$ and a knot vector $T = [t_0, \ldots, t_{n+p+1}]$. The general B-spline curve approximation problem is thus given by

$$\min_{n,T,\hat{T},\mathbf{C}} \sum_{k=0}^{m} d_o^2\left(\mathbf{x}_k, \mathcal{C}\left(\hat{t}_k\right)\right), \tag{2.91}$$

where d_o is the orthogonal distance between a data point and the B-spline curve $\mathcal{C}$ given a data parametrization. In practice, n is determined by starting with a minimum or maximum number and iteratively increasing or decreasing the number of control points until a desired accuracy is obtained [Piegl & Tiller, 1997]. The point $\mathcal{C}\left(\hat{t}_k\right)$ is called the *foot point* of $\mathbf{x}_k$ on $\mathcal{C}$. The computation of the foot point with minimum Euclidean distance (orthogonal distance)

$$d_o\left(\mathbf{x}_k, \mathcal{C}\right) = \min_t \left\|\mathbf{x}_k - \mathcal{C}\left(t\right)\right\| \tag{2.92}$$

is a non-linear problem in the parameter t for each data point as is the selection of the optimal knot vector. The knot vector is usually precomputed using the data parameterization (see below). What is left is the parametrization of the data points. The problem

$$\min_{\hat{T},\mathbf{C}} \sum_{k=0}^{m} d_o^2\left(\mathbf{x}_k, \mathcal{C}\left(\hat{t}_k\right)\right), \tag{2.93}$$

has been considered as a global non-linear optimization problem by [Speer et al., 1998]. In order to avoid nonlinear optimization, for each $\mathbf{x}_k$ a parameter $\hat{t}_k$ can be precomputed. The curve is then fitted by minimizing an approximate to the squared orthogonal distance

$$\hat{d}_o^2\left(\mathbf{x}_k, \mathcal{C}\left(\hat{t}_k\right)\right) \approx d_o^2\left(\mathbf{x}_k, \mathcal{C}\left(\hat{t}_k\right)\right)$$

between the data point and the foot point $\mathcal{C}\left(\hat{t}_k\right)$. A linear system of equations in the control points is obtained by setting the partial derivatives of the distance measure

equal to zero. Given a proper initial curve the approximating curve is found in an iterative process with alternating parameter correction and curve fitting. The iteration finishes if either the distance vectors reach a desired degree of orthogonality or the change in the control points is less than a given threshold.

1. Compute initial parameters $\hat{t}_{k,0}$ and an initial curve $\mathcal{C}_0$
2. Compute the corrected parameters $\hat{t}_{k,i+1}$ based on $\mathcal{C}_i$
3. Compute an updated curve $\mathcal{C}_{i+1}$ by solving a linear system of equations based on $\hat{d}_o^2\left(\mathbf{x}_k, \mathcal{C}_{i+1}\left(\hat{t}_{k,i+1}\right)\right)$

The details for each of the three steps are discussed in the following paragraphs.

Initial Parametrization

The iterative process outlined above requires a reasonable initial parametrization of the data points. The parametrization is straightforward in case of an *ordered* set of data points. Equation (2.85) or one of the other methods introduced in the previous section can be used to obtain the initial parametrization.

The case of an *unordered* point set is somewhat more complex. The adjacency relation between the unordered data points can be easily reconstructed if the distribution of the points is sufficiently thin. Then, points can be ordered by connecting nearest neighbors with respect to the Euclidean distance. However, this approach is not applicable for dense and noisy point sets. The moving least squares approach [Levin, 1998] computes a point $\hat{\mathbf{x}}_k$ for each point $\mathbf{x}_k$ in the set by a weighted regression method that fits a local polynomial curve to $\mathbf{x}_k$ and the points in its neighborhood. The weights are determined by a *kernel* function on the distance between $\mathbf{x}_k$ and the neighbors. Common choices for the kernel function are the Gaussian, the Tri-cube and the Epanechnikov kernel [Hastie et al., 2001]. The procedure may be repeated iteratively until a sufficiently thin point arrangement has been determined. The effect of the neighborhood size on the thinning process has been studied by [Lee, 2000]. Additionally, a method for reducing the effect of unwanted points in the local neighborhood has been given by conducting a data structure based on the *Euclidean minimum spanning tree* (see e.g. [Cormen et al., 1990]) of the point set. Once a sufficiently thin arrangement of data points $\{\hat{\mathbf{x}}_k\}_{k=0}^m$ has been found, the points can be ordered as suggested above.

Knot Vector Selection

As in B-spline interpolation the shape of the resulting curve in B-spline approximation considerably depends on the knot vector

$$T = [t_0, \ldots, t_{n+p+1}]$$

to be chosen. According to [Piegl & Tiller, 1997] a knot vector that reflects the distribution of the data points and satisfies the Schönberg-Withney condition can be found by assigning

$$\begin{aligned} t_0 = \ldots = t_p = 0 \quad \text{and} \quad & t_n = \ldots = t_{n+p+1} = 1, \\ t_{p+j} = (1-\alpha)\,\hat{t}_{i-1} + \alpha \hat{t}_i \qquad & j = 1, \ldots, n-p \end{aligned} \tag{2.94}$$

with

$$r = \frac{m+1}{n-p+1}, \quad i = \lfloor jr \rfloor, \quad \alpha = jr - i.$$

given the initial parametrization $\left\{\hat{t}_k\right\}_{k=1}^{m}$ of the data points. However, this knot placement technique does not reflect the shape of the curve to be approximated and is thus not optimal. Various methods that place the knots based on shape of the (ordered) data points have been proposed [Razdan, 1999; Li et al., 2005; Park & Lee, 2007]. The authors of the most recent paper proposed to select points of high curvature, called *dominant points*, and compute the knot vector according to the averaging method in B-spline interpolation using the parameterization of the dominant points.

Adjusting the knot vector by nonlinear optimization has been considered by [Hoschek & Schneider, 1990; Hoschek, 1992; Schwetlick & Schütze, 1995].

Orthogonal Distance Error

The objective function in B-spline curve approximation is the sum of the squared orthogonal distances

$$F = \sum_{k=1}^{m} d_o^2 \left(\mathbf{x}_k, \mathcal{C} \left(\hat{t}_k \right) \right), \tag{2.95}$$

between the data points $\{\mathbf{x}_k\}_{k=1}^m$ and the pth-degree B-spline curve $\mathcal{C}$. Since in general no closed-form solution to the correct orthogonal distance exists, various approximations $\hat{d}^2 \left(\mathbf{x}_k, \mathcal{C} \left(\hat{t}_k \right) \right)$ to the squared orthogonal distance have been proposed. Independent of the distance error term, applying the standard technique of linear least squares fitting leads to a linear system of equations by setting the partial derivatives equal to zero.

The *point distance error term*

$$\hat{d}_P^2 \left(\mathbf{x}_k, \mathcal{C} \right) = \left(\mathbf{x}_k - \mathcal{C} \left(\hat{t}_k \right) \right)^2 \tag{2.96}$$

is the most common measure used in curve fitting [Plass & Stone, 1983; Hoschek, 1988; Goshtasby, 2000; Saux & Daniel, 2003] which yields the linear system of equations

$$\mathbf{N}^T \mathbf{N} \mathbf{C} = \mathbf{P} \tag{2.97}$$

with coefficient matrix

$$\mathbf{N}_P = \begin{bmatrix} N_{0,p} \left(\hat{t}_1 \right) & \dots & N_{n,p} \left(\hat{t}_1 \right) \\ \vdots & \ddots & \vdots \\ N_{0,p} \left(\hat{t}_m \right) & \dots & N_{n,p} \left(\hat{t}_m \right) \end{bmatrix}, \tag{2.98}$$

the matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_0^T \\ \vdots \\ \mathbf{c}_n^T \end{bmatrix} \tag{2.99}$$

containing the unknown control points in its rows, and the right hand side

$$\mathbf{P} = \begin{bmatrix} \sum_{k=0}^{m} N_{0,p} \left(\hat{t}_k \right) \mathbf{x}_k^T \\ \vdots \\ \sum_{k=0}^{m} N_{n,p} \left(\hat{t}_k \right) \mathbf{x}_k^T \end{bmatrix}. \tag{2.100}$$

The notation in (2.97) is used as a short form for multiple linear system of equations. For each column in $\mathbf{P}$ a single linear system of equations has to be solved. The number of systems equals the dimension of the control points $\mathbf{c}_k$.

The *tangent distance error term* has been used in curve fitting by [Blake & Isard, 2000] and is defined by

$$\hat{d}_T^2\left(\mathbf{x}_k, \mathcal{C}\right) = \left(\left(\mathcal{C}\left(\hat{t}_k\right) - \mathbf{x}_k\right)^T \mathbf{n}_k\right)^2, \tag{2.101}$$

where each $\mathbf{n}_k$ is the unit normal vector of the current fitting curve at the point $\mathcal{C}\left(\hat{t}_k\right)$. A third error term that has been recently introduced to B-spline curve fitting is *squared distance error term* [Wang et al., 2006]. The authors use the distance

$$|d_k| = \left\|\mathbf{x}_k - \mathcal{C}\left(\hat{t}_k\right)\right\|$$

and the curvature radius

$$\rho_k = \frac{\left|\mathcal{C}'\left(\hat{t}_k\right)\right|^3}{\left|\mathcal{C}'\left(\hat{t}_k\right) \times \mathcal{C}''\left(\hat{t}_k\right)\right|}$$

of $\mathcal{C}\left(t\right)$ at $t = \hat{t}_k$. The sign of the distance d_k is defined negative if the center of the curvature radius and $\mathbf{x}_k$ are on opposite sides of $\mathcal{C}\left(t\right)$. The squared distance error term is defined as

$$\hat{d}_S^2\left(\mathbf{x}_k, \mathcal{C}\right) = \frac{d_k}{d_k - \rho_k}\left(\left(\mathcal{C}\left(\hat{t}_k\right) - \mathbf{x}_k\right)^T \mathbf{g}_k\right)^2 + \left(\left(\mathcal{C}\left(\hat{t}_k\right) - \mathbf{x}_k\right)^T \mathbf{n}_k\right)^2 \tag{2.102}$$

for $d < 0$ and

$$\hat{d}_S^2\left(\mathbf{x}_k, \mathcal{C}\right) = \left(\left(\mathcal{C}\left(\hat{t}_k\right) - \mathbf{x}_k\right)^T \mathbf{n}_k\right)^2 \tag{2.103}$$

for $0 \leqslant d_k < \rho_k$. Note that for $d_k > 0$ there is always $d_k < \rho_k$, otherwise the $\mathcal{C}\left(\hat{t}_k\right)$ is not the closest point on the curve to $\mathbf{x}_k$. The squared error minimization comes close to the full Newton method even though it makes use of an quadratic approximation of the Hessian.

By use of the TDM and SDM error term, the control points are computed in a single linear system of equations. The linear system thus grows with the dimension of the control points. In case of a two-dimensional curve, the length of the parameter vector and the number of columns in $\mathbf{N}$ double compared to the size in PDM. These methods may thus not be feasible in curve fitting with many control points.

Parameter Correction

In parameter correction the current fit curve $\mathcal{C}$ is fixed and an updated parametrization $\hat{T}_{i+1}$ is computed to obtain new foot points with distance vectors which are closer to orthogonal to the curve. The updated parameters are computed by

$$\hat{t}_{k,i+1} = \hat{t}_{k,i} + \Delta\hat{t}_{k,i}, \quad k = 1, \ldots, m \tag{2.104}$$

using the correction $\Delta\hat{t}_{k,i}$. Three methods for computing an updated data parametrization are common [Hoschek et al., 1993]. The *intrinsic parametrization* [Hoschek, 1988] updates the parameters by the projection of the distance vector to the tangent at the current foot point

$$\Delta\hat{t}_{k,i} = \frac{1}{L}\left(\mathbf{x}_k - \mathcal{C}\left(\hat{t}_{k,i}\right)\right)^T \frac{\mathcal{C}'\left(\hat{t}_k\right)}{\left\|\mathcal{C}'\left(\hat{t}_k\right)\right\|} \tag{2.105}$$

where L is an approximation to the curve length, as computed from polygonal approximation to the curve. The second method was suggested by [Rogers & Fog, 1989] and uses the series expansion

$$d\left(\mathbf{x}_k, \mathcal{C}\right) = \mathbf{x}_k - \mathcal{C}\left(\hat{t}_{k,i}\right) \approx \mathbf{x}_k - \mathcal{C}\left(\hat{t}_{k,i}\right) - \mathcal{C}'\left(\hat{t}_k\right)\Delta\hat{t}_{k,i}.$$

Taking the absolute value and differentiation yields the correction

$$\Delta\hat{t}_{k,i} = \left(\mathbf{x}_k - \mathcal{C}'\left(\hat{t}_k\right)\mathcal{C}\left(\hat{t}_{k,i}\right)\right)^T \frac{\mathcal{C}'\left(\hat{t}_k\right)}{\left\|\mathcal{C}'\left(\hat{t}_k\right)\right\|^2}. \tag{2.106}$$

A third method for minimizing the local error vector has been proposed by [Plass & Stone, 1983]. The idea is to minimize the squared distance which after differentiation gives

$$f = \left(\mathbf{x}_k - \mathcal{C}\left(\hat{t}_{k,i}\right)\right)^T \mathcal{C}'\left(\hat{t}_k\right).$$

Using the well-known Newton iteration formula to compute a zero of f leads to the correction formula

$$\Delta\hat{t}_{k,i} = -\frac{\left(\mathbf{x}_k - \mathcal{C}\left(\hat{t}_{k,i}\right)\right)^T \mathcal{C}'\left(\hat{t}_{k,i}\right)}{\left(\mathbf{x}_k - \mathcal{C}\left(\hat{t}_{k,i}\right)\right)^T \mathcal{C}''\left(\hat{t}_k\right) - \mathcal{C}'\left(\hat{t}_k\right)^T \mathcal{C}'\left(\hat{t}_k\right)} \tag{2.107}$$

Nonlinear optimization methods have been also considered for parameter correction by [Saux & Daniel, 2003].

Endpoint Considerations

For cyclic curves the endpoint constraints are straight forward as previously outlined in Section 2.4.2 of Chapter 2. For non-cyclic B-spline curves however, the treatment of the curves endpoints needs further considerations. In the generic case of Equation (2.95) no constraints are applied to the curve. In this case of free endpoints it is possible to use both, a clamped or an unclamped knot vector. In case the start point $\mathbf{x}_1$ and the end point $\mathbf{x}_m$ of the fitting curve $\mathcal{C}$ are known in advance, a clamped knot vector is chosen and the first and the last control points are set

$$\mathbf{c}_0 = \mathbf{x}_1 \text{ and } \mathbf{c}_n = \mathbf{x}_m. \tag{2.108}$$

The remaining $m-2$ data points are substituted

$$\tilde{\mathbf{x}}_k = \mathbf{x}_k - N_{0,p}\left(\hat{t}_k\right)\mathbf{x}_1 - N_{m,p}\left(\hat{t}_k\right)\mathbf{x}_m, \quad k = 2, \ldots, m-1 \tag{2.109}$$

and the curve fitting error defined in Equation (2.95) is adapted

$$F = \sum_{k=2}^{m-1} \hat{d}_o^2\left(\tilde{\mathbf{x}}_k, \mathcal{C}\left(\hat{t}_k\right)\right) \tag{2.110}$$

where the number of equations obviously reduces by two [Piegl & Tiller, 1997].

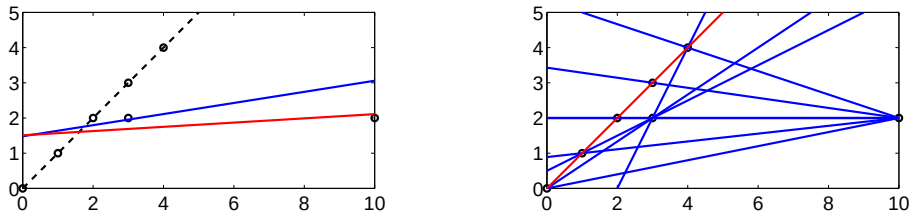


Figure 2.15: A data set from the line $y = x$ with an outlier at $(10, 2)$. (a) The initial regression result (blue) and the result after four iterations using the 'drop out largest residual' heuristic (red). (b) Lines computed for all pairs (blue) and the result using a robust algorithm (red).

2.6 Robust Estimation

The standard least squares fitting algorithms presented in Section 2.5 assume that the error in the data is relatively small and underlies a Gaussian distribution. These algorithms may fail in case the assumptions are violated and some data items are subject to gross errors, called *outliers*. The focus in this section is on algorithms that establish a framework where the previously presented algorithms may be patched into, such that the package is insensitive to deviations from the assumptions up to some degree. This degree of robustness is usually measured by the *breakdown point*, that is, the minimum fraction of outliers in a data set that can cause the algorithm to diverge arbitrarily far from the true estimate [Rousseeuw & Leroy, 2003]. The breakdown point of least squares is 0 as a single outlier can be placed such that the fit moves arbitrarily far from the true estimate. A line regression example that failed using a least squares fit is shown in Figure 2.15. Obviously, the theoretical maximum value of a breakdown point is 0.5. If there are more outliers, they can be arranged so that a fit through them will minimize any objective function of an estimator.

Robust parameter estimation algorithms are usually embedded in a two-stage procedure. The first stage utilizes an algorithm designed to be particularly robust against outliers to find a good initial estimate and simultaneously detect outliers. In the second stage such an initial estimate is assumed to be given and a refinement of the estimate is computed iteratively. The available algorithms for both stages are covered in that order by the two parts of this section.

2.6.1 Outlier Rejection

A number of heuristic rejection methods have been proposed to deal with outliers. An example is the method of dropping out the item with largest residual. However, it can be easily shown that even a single outlier can cause this heuristic to fail [Fischler & Bolles, 1981]. In Figure 2.15(a) a data set representing a line including an outlier at $(10, 2)$ is shown. The dashed line is the true underlying model and the blue line the initial least squares fit. Starting with an initial least squares fit using all data items results in a residual that is less for the outlier than it is for the valid data item $(4, 4)$. Applying the 'drop out largest residual' method iteratively for three times and estimating the line by least squares converges with the red line.

RANSAC is the abbreviation for *RANdom SAmple Consensus* and refers to an iterative parameter estimation method that is able to robustly fit a model to experimental data in the presence of a relatively high number of outliers [Fischler & Bolles, 1981]. Let n be the number of observations and s the number of data items needed to fit a model, usually the number of parameters in the model. The algorithm randomly picks subsets of size s from the input data and fits a model to each subset. For each fitted model $\mathcal{M}_i$, $i = 1, 2, \ldots$ the consensus set $\mathcal{C}_i$ of data items that support $\mathcal{M}_i$ is determined using a threshold t on the residuals. Finally, a model is fitted using the data items of the largest consensus set $\mathcal{C}_j$ that support the model $\mathcal{M}_j$.

The algorithms needs the number s, a threshold t and the number m of trials as input. Assuming the data contains $\epsilon \cdot n$ outliers ($\epsilon \in [0, 1]$) the number m of trials necessary to include at least one outlier-free sample with probability p is given as

$$m = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)}. \tag{2.111}$$

Rather than running the whole number m of trials the process may be stopped if a consensus set of size $(1-\epsilon) \cdot n$ is found. In case ϵ is not known in advance, it is possible to use a worst case assumption of 0.5 and iteratively correct it using the size of a consensus set.

Another related method is the *least median of squares (LMedS)* technique. The minimization problem

$$\min \underset{i}{\text{median}} (r_i) \tag{2.112}$$

is solved using m random subsets drawn from the original data set as in RANSAC. Each subset is fitted by least squares and the model with minimum median of the residuals is the best solution of the algorithm and a good solution with probability p. The LMedS method is more general than RANSAC since no threshold parameter needs to be specified.

M-estimator	$\rho(x)$	$\psi(x)$	$w(x)$
Huber $\begin{cases} \lvert x\rvert \leqslant c \\ \lvert x\rvert > c \end{cases}$	$\begin{cases} x^2/2 \\ c(\lvert x\rvert - c/2) \end{cases}$	$\begin{cases} x \\ c\,\mathrm{sign}(x) \end{cases}$	$\begin{cases} 1 \\ c/\lvert x\rvert \end{cases}$
Cauchy	$\frac{c^2}{2}\log\left(1+(x/c)^2\right)$	$\frac{x}{1+(x/c)^2}$	$\frac{1}{1+(x/c)^2}$
Tukey $\begin{cases} \lvert x\rvert \leqslant c \\ \lvert x\rvert > c \end{cases}$	$\begin{cases} \frac{c^2}{6}(1-(1-\frac{x}{c}^2)^3) \\ (c^2/6) \end{cases}$	$\begin{cases} x(1-\frac{x}{c}^2)^2 \\ 0 \end{cases}$	$\begin{cases} (1-\frac{x}{c}^2)^2 \\ 0 \end{cases}$

Table 2.3: Functions of commonly used M-estimators.

2.6.2 M-Estimators

M-Estimators are a robust regression technique in the presence of outliers given a good initial estimate of the underlying model. The effects of outliers are reduced introducing an objective function

$$\min \sum_i \rho(\hat{r}_i) \tag{2.113}$$

that replaces the squared sum of residuals. The residuals r_i are required to be standardized according to

$$\hat{r}_i = \frac{r_i}{\sigma} \tag{2.114}$$

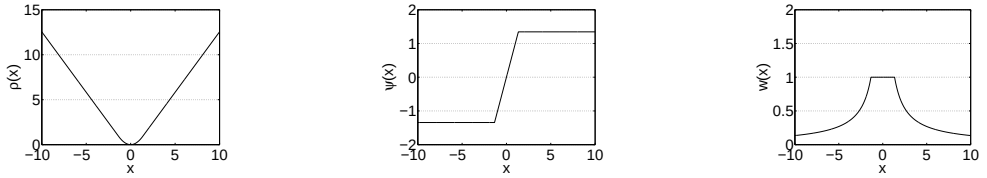
where σ is the scale of the residuals. If σ is not known in advance it can be computed via the robust estimate

$$\hat{\sigma} = 1.4826\left(1+\frac{5}{n-m}\right)\underset{i}{\mathrm{median}}\,|r_i| \tag{2.115}$$

where n is the size of the data set and m is the dimension of parameter vector. The constant 1.4826 is a coefficient to achieve the same efficiency as least squares in the presence of only Gaussian noise and $\frac{5}{n-m}$ is to compensate the effect of small data sets. The details on these magic numbers can be found in [Rousseeuw & Leroy, 2003].

The loss function ρ is symmetric, positive-definite with a unique minimum at zero and is chosen to be less increasing than square [Zhang, 1997]. The M-estimator of the parameter vector $\mathbf{p} = (p_1, \ldots, p_m)^T$ based on $\rho(\hat{r}_i)$ is the solution to the m equations

$$\sum_i \psi(\hat{r}_i)\frac{\partial \hat{r}_i}{\partial p_j} = 0, \quad j = 1, \ldots, m \tag{2.116}$$



(a) (b) (c)

Figure 2.16: The loss function $\rho(x)$, the influence function $\psi(x)$ and the weight function $w(x)$ of the Huber M-estimator [Huber, 1981].

where the derivative

$$\psi(\hat{r}_i) = \frac{d\rho(x)}{dx} \tag{2.117}$$

is called the *influence function*. By definition of the *weight function*

$$w(x) = \frac{\psi(x)}{x} \tag{2.118}$$

Equation (2.116) becomes

$$\sum_i w(\hat{r}_i)\,\hat{r}_i \frac{\partial \hat{r}_i}{\partial p_j} = 0, \quad j = 1, \ldots, m. \tag{2.119}$$

The system of equations (2.119) is solved by the *iteratively reweighted least-squares* (IRLS) method which alternates steps of (1) calculating weights and (2) computing the weighted least squares estimate

$$\min_{\mathbf{p}} \sum_i w\left(\hat{r}^{(k-1)}\right) r_i^2 \tag{2.120}$$

where $^{(k)}$ indicates the iteration number. This process is repeated iteratively until convergence of a criteria on the change of the parameters or the weights. The scale may be updated in the beginning but needs to be fixed in order to guarantee convergence.

Table 2.3 lists the loss function $\rho(x)$, the influence function $\psi(x)$ and the weight function $w(x)$ for three commonly used M-estimators. The tuning parameter c are usually chosen to obtain 95% (Huber: $c = 1.345$, Cauchy: $c = 2.3849$, Tukey: $c = 4.6851$). Here efficiency means the ratio of minimum possible variance of the M-estimate relative to the actual variance, assuming that the underlying error distribution is in fact normal (Gaussian) [Stewart, 1999].

Kernel	Equation
Nearest-Neigbhor	$D(t) = \begin{cases} 1 & \text{if } \lvert t\rvert \leqslant 1 \\ 0 & \text{otherwise} \end{cases}$
Epanechnikov	$D(t) = \begin{cases} \frac{3}{4}\left(1-t^2\right) & \text{if } \lvert t\rvert \leqslant 1 \\ 0 & \text{otherwise} \end{cases}$
Tri-cube	$D(t) = \begin{cases} \left(1-t^3\right)^3 & \text{if } \lvert t\rvert \leqslant 1 \\ 0 & \text{otherwise} \end{cases}$
Gaussian	$D(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2}$

Table 2.4: Kernels for local smoothing

2.7 Data Smoothing

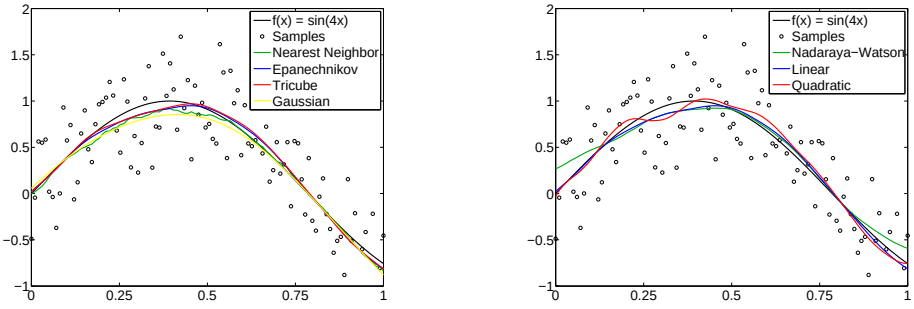
Data smoothing plays an important role in statistics where a real valued non-linear function f over the domain $\mathbb{R}^p$ is to be estimated from noisy observations $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$ in such a way that the resulting function $\hat{f}(\mathbf{x})$ is smooth in $\mathbb{R}^p$. Kernel regression smoothing introduced in Section 2.7.1 are non-parametric. Two parametric models for smoothing based on piecewise polynomials (B-splines) are introduced in Section 2.7.2 and Section 2.7.3. Other parametric models such as wavelets are covered in [Hastie et al., 2001].

2.7.1 Kernel Smoothing

Kernel smoothing is a non-parametric estimation technique. The idea of kernel smoothing is to fit a simple model separately at each query point $\mathbf{x}_0$ using only those observations close to $\mathbf{x}_0$. The localization is established by a weighting function $K_\lambda(\mathbf{x}_0, \mathbf{x}_i)$ called the *kernel* which assigns the weight

$$K_\lambda(\mathbf{x}_0, \mathbf{x}_i) = D\left(\frac{\|\mathbf{x}_i - \mathbf{x}_0\|}{\lambda}\right) \tag{2.121}$$

to each $\mathbf{x}_i$ based on its distance from x_0. As the Euclidean norm depends on the unit in each coordinate, the predictors should be normalized in advance. In the case $p = 2$, the normalization can be integrated into (2.121) by replacing the denominator with $(\max x_i - \min x_i)\,\lambda$. A selection of popular kernel functions are listed in Table 2.4.



(a) (b)

Figure 2.17: Applying different kernel smoothers to the random sampling of the function $f(x) = \sin(4x)$. (a) Kernel functions with local linear regression and (b) kernel smoothing methods using Epanechnikov kernel.

The Epanechnikov and the Tri-cube kernel have compact support, that is, the window size is bounded. The Gaussian density function is a noncompact kernel with the standard deviation playing the role of the window size.

The *Nadaraya-Watson* kernel regression method is the locally weighted average

$$\hat{f}_\lambda(\mathbf{x}_0) = \frac{\sum_{i=1}^{n} K_\lambda(\mathbf{x}_0, \mathbf{x}_i) y_i}{\sum_{i=1}^{n} K_\lambda(\mathbf{x}_0, \mathbf{x}_i)}. \tag{2.122}$$

This locally weighted average may, however, result in biased estimates at the boundaries of the domain because of the asymmetry of the kernel in that region. This effect can be observed in Figure 2.17(b). An improved smoother can be achieved by local linear regression rather than fitting constants. Define $\mathbf{b}(\mathbf{x}) = \left(\mathbf{x}^T, 1\right)$. Let $\mathbf{B}$ the $n \times p$ matrix with ith row $\mathbf{b}(\mathbf{x}_i)$ and $\mathbf{W}(\mathbf{x}_0)$ the $n \times n$ diagonal matrix with ith diagonal element $K_\lambda(\mathbf{x}_0, \mathbf{x}_i)$. The explicit expression for the local linear regression estimate of $\mathbf{x}_0$ is then given by

$$\hat{f}_\lambda(\mathbf{x}_0) = \mathbf{b}(\mathbf{x_0}) \left(\mathbf{B}^T \mathbf{W}(\mathbf{x}_0) \mathbf{B}\right)^{-1} \mathbf{B}^T \mathbf{W}(\mathbf{x}_0) \mathbf{y}. \tag{2.123}$$

Further improvement give local polynomial models, which reduce the bias in regions of curvature in the underlying function. The disadvantage of local polynomial regression is a potentially increased variance. The result of the local quadratic smoother in Figure 2.17(b) possesses this behavior. All above variants of kernel smoothers may be embedded in a robust environment (see Section 2.6). The regression is performed using the product of the kernel weights and the M-estimator weights.

2.7.2 Smoothing Splines

In the scope of the B-spline basis, *smoothing splines* are B-spline curves which are fitted by approximation with an additional regularization term that penalizes the curvature in the function and thus establishes a smoothing of the data with the curve fitted. The objective function to be minimized in smoothing spline approximation is defined as

$$F = \sum_{k=0}^{m} d_o^2 \left(\mathbf{x}_k, \mathcal{C}\right) + \lambda \int \mathcal{C}'' \left(t\right)^2 dt \tag{2.124}$$

where the first term is one of the error terms defined previously and the second term is the square of the second derivative of the fitted curve, commonly used as smoothing penalty since the seminal work of [Reinsch, 1967]. In order to avoid the problem of choosing a feasible knot vector, a relatively large number of equally spaced knots is used that would lead to data overfitting and might also violate the Whitney-Schoenberg condition with conventional B-spline fitting. The smoothing term restricts the flexibility of the fitted curve and thus prevents overfitting. The *smoothing parameter* λ is a positive constant that establishes the degree of smoothing. For $\lambda \to 0$ the obtained curve is the same as in B-spline approximation and for $\lambda \to \infty$ the curve becomes a least squares line.

2.7.3 P-Splines

Very similar to smoothing splines, *P-splines* have been introduced by [Eilers & Marx, 1996] as penalized B-spline curves which are computed by approximation with a penalty term based on higher order finite differences of the coefficients of adjacent B-spline basis functions. With

$$\Delta \mathbf{c}_j = \mathbf{c}_j - \mathbf{c}_{j-1}$$

the lth-order finite difference is defined by

$$\Delta^l \mathbf{c}_j = \underbrace{\Delta \ldots \Delta}_{l} \mathbf{c}_j. \tag{2.125}$$

The objective function of an lth-order P-spline is thus defined as

$$F = \sum_{k=0}^{m} d_o^2 \left(\mathbf{x}_k, \mathcal{C}\right) + \lambda \sum_{j=l}^{n} \Delta^l \mathbf{c}_j. \tag{2.126}$$

As proposed for smoothing splines a high number of equally spaced knots is chosen and an unclamped knot vector is used to minimize border effects.

2.8 Model Selection and Assessment

In many parameter estimation problems some degrees of freedom of the model in question are not known in advance. For instance, in B-spline approximation the optimal number of intervals is usually unknown and in data smoothing it is the value of the smoothing parameter λ. In machine learning and statistics *model selection* methods are used to find the optimal value of such *tuning parameters* automatically. The performance of a predefined number of models, each with the tuning parameter assigned a different value, is validated and the best model is chosen. To follow the notation in machine learning, the data used for model extraction will be referred to as *training data* and the data set used for validation will be called *test data*. The performance of the extracted models is measured by the *test error*, the expected error of a test data set that is independent from the training data. It is emphasized that training data and test data need to be disjoint because with growing complexity a model will adopt more and more to the training data and the *training error* will underestimate the true test error. If 'enough' observational data is available, it is possible to split the data into training data and a test data in advance. In the case where it is not possible to split the data into parts there are various methods that approximate the test error either analytically or by an efficient sample reuse [Hastie et al., 2001]. These methods may be also applied to *model assessment* where the error of the finally chosen model is estimated. Let the training error be defined as the sum of squared residuals with a model

$$F = \frac{1}{n}\sum_{i=1}^{n} r_i^2. \tag{2.127}$$

where n is the number of data items in the data set.

2.8.1 Akaike Information Criterion

The *Akaike Information Criterion* (AIC) [Akaike, 1974] computes the test error analytically from the training error by adding a term of test error underestimation. Applied to model selection the model that minimizes the AIC has to be chosen. For the Gaussian model with variance $\hat{\sigma}^2$ assumed to be known, the AIC is defined as

$$AIC = F + 2\frac{m}{n}\hat{\sigma}. \tag{2.128}$$

where F is the training error and m the number of parameters in the model. For nonlinear and other complex models, m needs to be replaced by some measure of model complexity such as used in 2.8.3.

2.8.2 Cross Validation

In *cross validation* (CV) the entire data set of size n is split up into k parts of size n/k. The residuals of the data items in each part are computed with the model that was fitted to all the other $(k-1)$ parts. All residuals $\hat{r}_i$ are computed by fitting k models. The cross validation estimate of the test error is

$$CV = \frac{1}{n}\sum_{i=1}^{n}\hat{r}_i^2. \tag{2.129}$$

The value of k needs to be chosen depending on the size of the input data set and on how the performance of the algorithm varies with the size of the training data set. Typical choices for k are 5 or 10. Choosing $k = n$ is known as *leave-one-out cross validation* and can be computed quickly in linear fitting as shown in the next section.

2.8.3 Generalized Cross Validation

In linear fitting (see 2.5.2) the leave-one-out cross validation can be approximated using the trace of the hat matrix $\mathbf{H}$. The estimate

$$GCV = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{r_i}{1-\operatorname{trace}(\mathbf{H})/n}\right)^2 \tag{2.130}$$

is called the *generalized cross validation* (GCV) where the quantity $\operatorname{trace}(\mathbf{H})$ is the effective number of parameters in the model [Hastie & Tibshirani, 1990]. It has been shown by [Stone, 1977] that AIC and GCV are asymptotically equivalent.

2.8.4 Bootstrap Methods

The *non-parametric bootstrap* is a resampling method used to assess the statistics of observational data. It randomly draws *bootstrap samples* with replacement from the training data with the same size as the training data set. A model is fitted to each bootstrap sample and the resulting models are examined. In *parametric bootstrap* the items in a bootstrap sample are perturbed by additive Gaussian noise or any other parametric model before the model is fitted.

Let $\bar{\mathcal{B}}_i$ be the set of all bootstrap samples that do not contain the ith datum of the training data set and $\left|\bar{\mathcal{B}}_i\right|$ the number of these samples. With $r_{i,B}$ the residual of the ith data item with the model fitted to the bootstrap sample B the leave-one-out bootstrap estimate of the test error is defined by

$$F_{\mathcal{B}} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\left|\bar{\mathcal{B}}_i\right|}\sum_{B\in\bar{\mathcal{B}}_i} r_{i,}. \tag{2.131}$$

3 Calibration of a Laser Projector

Active vision systems utilizing a laser point pattern have found a large interest in recent years [Lequellec & Lerasle, 2000; Woo & Dipanda, 2000; Clabian et al., 2001; Marzani et al., 2002; Dipanda et al., 2003; Dipanda & Woo, 2005; Popescu et al., 2006; Aoki et al., 2007; Lubeley, 2007] since bright laser spots allow for applications in dynamic processes and in inhomogeneous ambient illumination environments. These measurement devices consist of a laser projector generating a beam of laser rays and a digital camera with known intrinsic parameters. Commonly lens distortion is considered and described by the parameters. The point pattern can be generated by a diffraction grating [Lequellec & Lerasle, 2000; Marzani et al., 2002; Dipanda et al., 2003; Dipanda & Woo, 2005; Popescu et al., 2006] or a Liquid Crystal on Silicon (LCoS) microdisplay [Lubeley, 2007] installed in front of a laser module. Typical extensions of the laser point pattern are grids with sizes of 7×7 [Clabian et al., 2001; Popescu et al., 2006], 15×15 [Clabian et al., 2001] or 19×19 [Clabian et al., 2001; Marzani et al., 2002; Dipanda et al., 2003; Dipanda & Woo, 2005] beams. Each laser ray is projected to the surface of a target object and the reflected light appears as a bright spot in the images of the stereo camera.

A big limitation of systems applying a single laser projector is the trade-off that has to be made among the density of the acquired point cloud and the extension of the field of view, due to the pyramidal shape of the beam. The field of view can be extended by increasing the distance between the illumination unit and the target object or by extension of the beam fan angle whereas in both cases the laser spot density on the surface of the target object shrinks.

However, in many applications a high density of the depth map over a wide lateral extension is required. In this Chapter a new measurement device bridging this gap by means of multiple laser projectors and a stereo camera is introduced and a new method for the reconstruction of a laser projector's beam geometry is presented. In Section 3.1 the experimental setup is introduced and a rough outline of the calibration strategy is given. The focus in Section 3.2 is on the detection of the laser spots in the images of the stereo camera. The reconstruction of the point pattern matrix needed for labeling of the laser spots is treated in Section 3.3. The reconstruction of the laser beam geometry is covered in Section 3.4. Finally, experimental results are presented and the accuracy of the measurement device is analyzed in Section 3.5.

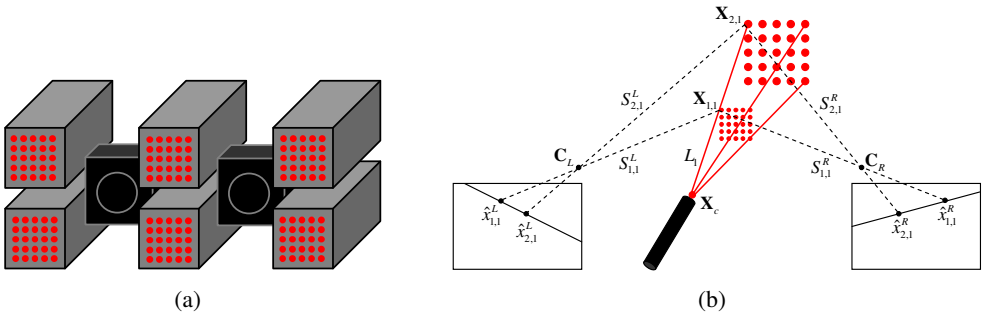


Figure 3.1: (a) Experimental setup and (b) calibration procedure for a laser module

3.1 Experimental Setup and Calibration Strategy

In this section the measurement device is introduced and an outline of the calibration procedure is given. An illustration of the proposed new measurement device is shown in Figure 3.1(a). It consists of a stereo camera and six laser projectors arranged in a matrix of two by three projectors in vertical and horizontal direction, respectively. Each laser projector consists of a laser module with a mounted diffraction grating generating a 15×15 point pattern with inter-ray angle of $\approx 2.34°$. A housing of each projector prevents the projection of higher order maxima of the diffraction grating. The stereo rig consists of two digital cameras with a resolution of 1392×1040 pel and a wide angle lens ($f = 4.8mm$) is used in order to achieve a wide field of view. The cameras are placed within the vertical gaps in between the projectors with a stereo basis of $b = 377.31$ mm. By joining all laser projectors into a single illumination unit, this system can be regarded as a trifocal measurement device. From this point a *laser ray* refers to a single ray that generates a single spot on the surface it is projected to. The term of a *laser beam* either refers to the set of rays of a single projector or the set of rays of all projectors, depending on the context.

The trifocal measurement device is subject to an elaborate calibration procedure that involves the calibration of the stereo camera and the reconstruction of the laser beam geometry. The calibration of the stereo camera is carried out initially using a test-field placed in multiple positions. A bundle adjustment algorithm is used in order to compute the extrinsic and intrinsic parameters including image distortion [Abraham & Hau, 1997; Abraham, 1999]. The mapping

$$f_x(x, y) \quad \text{and} \quad f_y(x, y) \tag{3.1}$$

combining image undistortion and rectification and its inverse

$$f_x^{-1}(\hat{x}, \hat{y}) \quad \text{and} \quad f_y^{-1}(\hat{x}, \hat{y}) \tag{3.2}$$

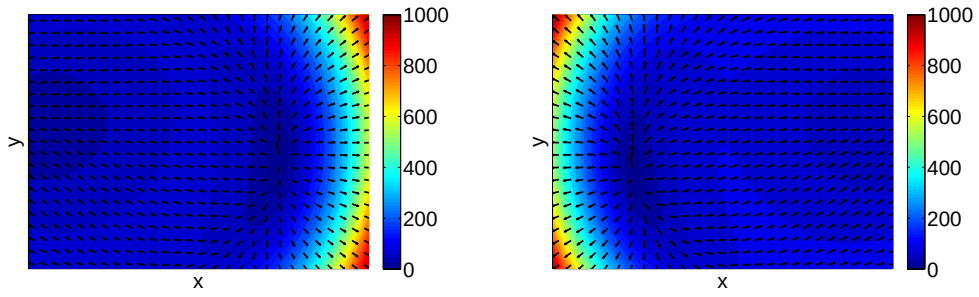
Algorithm 1: Laser calibration

Data: images of the laser point pattern
Result: laser beam geometry
begin
  laser spot detection
  pattern reconstruction and spot labeling
  reconstruction of 3-D points
  reconstruction of the laser beam geometry
end

are computed for both cameras and stored in look-up-tables of image size. Figure 3.2 visualizes $[f_x, f_y]^T$ for the stereo camera in the presented device. The color encodes the amount of distortion in pixels and the vectors illustrate the distortion direction.

The laser calibration procedure is carried out similar to the methodology introduced by [Marzani et al., 2002]. A planar surface is successively positioned in multiple depths within the field of view of the stereo camera and the laser projector. For each such constellation a stereo image pair is acquired while laser points are projected on the planar surface. In order to eliminate background texture in the feature extraction process a second pair of images without laser illumination is acquired with minimum possible delay. Algorithm 1 gives an outline of the major stages of the laser calibration procedure. In the first and the second stage the images of the left and the right camera are independently preprocessed. These stages involve the detection of the laser spots, treated in Section 3.2, and the assignment of the generating laser ray to each spot, covered in Section 3.3. The third stage uses the labeling information to find corresponding laser spots. The three-dimensional coordinates of the laser ray's intersection with the planar surfaces are reconstructed by triangulation using the known geometry of the stereo camera. Finally, the laser beam geometry is reconstructed using a global minimization scheme. The details of this stage of the algorithm are presented in Section 3.4. The whole procedure is repeated independently for each of the six laser modules. The relation of image features and the laser geometry is shown in Figure 3.1(b) for a planar surface placed in two depths. Each stereo image pair acquired during the calibration procedure for the planar surface in one position is referred to as a *measurement*.

The reason for the use of a stereo camera is the structured light pattern generated by the laser projector. It would hardly be possible to identify the correspondences between the spots in the image of a single camera and the generating laser ray. The trifocal epipolar constraints allow for efficient identification of correspondences as will be shown in Chapter 4. The major advantage compared to the mono camera



(a) Left camera (b) Right camera

Figure 3.2: Look-up-table for camera undistortion and rectification.

approach is the elimination of the extensive alignment process of the planar surface in front of the camera. Instead, the orientation of the plane is reconstructed from 3-D points computed by triangulation of corresponding points in the stereo images using the parameters of the epipolar geometry.

There are various reasons for the separation of stereo camera and laser calibration. The main reason is the strong image distortion introduced by the wide angle lens. The pattern reconstruction algorithm introduced in Section 3.3 assumes a rectified camera and is not able to deal with strong distortions in the point pattern such as caused by the utilized wide angle lens. Second, the laser projector alone does not provide absolute Euclidean information. Thus, it is not possible to compute absolute coordinates. Also, an exact alignment of an object is not feasible and thus no Euclidean information will be available from the laser spots. The used test-field provides the needed absolute Euclidean information. A third point is the limitation of the image area where the laser spots are imaged. Particularly for the estimation of the lens distortion data in the image border area is required. However, laser spots in these areas of the images are hardly available.

Algorithm 2: Laser spot detection

Data: image with laser spots and background image
Result: rectified coordinates of the laser spots
begin
  separate laser spots from background structures
  find local minima in scale space representation using DoG
  fit paraboloid to the intensity landscape in the neighborhood of each response
  find the maximum of the paraboloid with sub-pixel accuracy
  compute the undistorted and rectified coordinate
end

3.2 Laser Spot Detection

The algorithm presented in this section aims for detection of all laser spots in an image whilst erroneous responses should be avoided. The main error sources for the laser spot detection procedure are (1) background structures, (2) varying surface reflectivity, and (3) camera noise. Varying surface reflectivity results in spots of varying size and brightness. Spots of low contrast may vanish and an error in the position may be introduced for spots of large extent. The background structures may induce erroneous responses outside the area of interest. Strong noise in the images may also introduce erroneous responses.

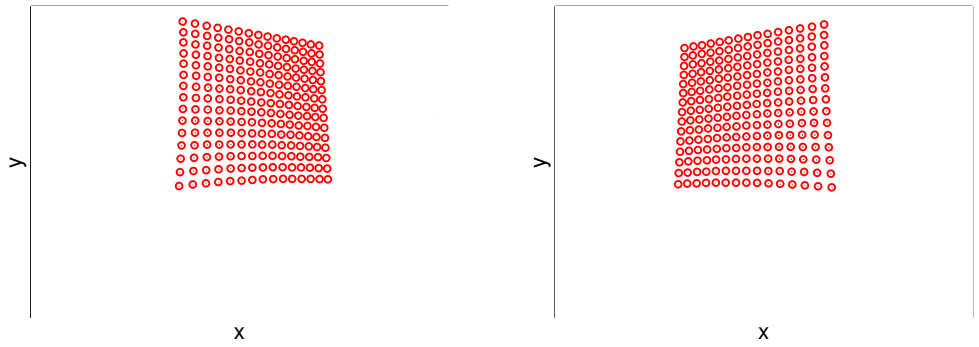
The background structures are initially eliminated. The laser calibration procedure starts with the acquisition of images with and without the planar surface illuminated by the laser projector. Let I_S be the image with laser illumination applied at acquisition time and I_B the image without laser illumination. The laser spots are separated from the background applying the image difference

$$I = I_S - I_B. \tag{3.3}$$

The separation of the laser spots from the background almost eliminates the effects of error source (1). Since the images I_S and I_B are acquired sequentially, a change in the background structure in the meanwhile can still cause minor effects.

In the second step the scale space representation D is constructed for I using the difference of Gaussian method with scales $\sigma_k = 2^{k/10}$, $k = 1, \ldots, 10$. The laser spots are detected at local minima in D according to Section 2.2.2. The detected laser spots for an image of the left and the right camera used in the calibration procedure are shown in Figure 3.3. The images have been inverted for the plot.

The discrete image coordinates (x, y) are corrected by means of local intensity interpolation according to the approach presented in [Nister, 2001, pg. 19]. A second degree polynomial is independently fitted to the intensities of the pixel at (x, y) and



(a) Left camera (b) Right camera

Figure 3.3: Laser spots detected in the original images (inverted for plot).

its neighbors in vertical and horizontal direction. The corrected coordinates are the positions of both parabola's vertex.

Finally, the corrected coordinates are transformed to the undistorted and rectified camera. Since the mapping (3.1) is available at integer positions in the image, the coordinates are computed by bilinear interpolation, according to

$$\hat{x} = g_x\left(\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4\right) \tag{3.4}$$

$$\hat{y} = g_y\left(\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4\right) \tag{3.5}$$

with the coordinates

$$\hat{x}_1 = f_x\left(\lfloor x \rceil, \lfloor y \rceil\right) \quad \text{and} \quad \hat{y}_1 = f_y\left(\lfloor x \rceil, \lfloor y \rceil\right) \tag{3.6}$$

$$\hat{x}_2 = f_x\left(\lfloor x \rceil, \lceil y \rceil\right) \quad \text{and} \quad \hat{y}_2 = f_y\left(\lfloor x \rceil, \lceil y \rceil\right) \tag{3.7}$$

$$\hat{x}_3 = f_x\left(\lceil x \rceil, \lfloor y \rceil\right) \quad \text{and} \quad \hat{y}_3 = f_y\left(\lceil x \rceil, \lfloor y \rceil\right) \tag{3.8}$$

$$\hat{x}_4 = f_x\left(\lceil x \rceil, \lceil y \rceil\right) \quad \text{and} \quad \hat{y}_4 = f_y\left(\lceil x \rceil, \lceil y \rceil\right). \tag{3.9}$$

An alternative to the final bilinear interpolation of the coordinates would be to initially apply the rectification and undistortion transform to the whole image. However, the laser spots located in the boundary area of the images may be lost by applying this transformation. This may lead to an increased error of the beam geometry or even cause the algorithm to fail due to unsufficient data. Furthermore, the transformation requires additional runtime and the feature detection algorithm may yield perturbed coordinates due to the deformation of the spots. The expected difference in the coordinates is estimated in Section 3.5.2. The expected error in the coordinates introduced by the proposed laser spot detection algorithm is given in Section 3.5.1 as a result of the calibration procedure.

3.3 Labeling

For the calibration of a laser projector the point correspondences of features observed in the images of the left and the right camera as well as correspondences of the features generated by the same laser ray in all measurements are needed. The correspondence problem reduces to a look up operation if the generating laser ray of each feature is known. In the *labeling* procedure each feature is assigned its corresponding laser ray. The input to the labeling algorithm is the set of features detected in a laser projector calibration image. A neighborhood relation $\mathcal{N}$ of the abstract set $\mathcal{V} = \{v_1, ..., v_n\}$ representing the features is established by means of the Delaunay triangulation $\mathcal{T}_D$ of $\mathcal{V}$. The labeling is carried out by reconstruction of the matrix pattern that is formed by the laser spots of a single projector in the camera images. The matrix reconstruction process starts with a classification of the edges in $\mathcal{T}_D$ in order to identify edges which belong to vertical and horizontal connections in the matrix. In a further step each feature is assigned its most likely vertical and horizontal neighbors. The feasibility of the neighborhood of a feature is used as the feature's quality. Finally, the matrix is reconstructed by a breadth-first search algorithm based on the order and the quality of the features. The algorithm is robust for a small number of extraneous points detected within or outside the pattern.

3.3.1 Edge Classification

The purpose of edge classification is the estimation of the likelihood of the edges in $\mathcal{T}_D$ to be a horizontal or vertical connection of features in the matrix pattern of the laser projector. Let $\mathcal{V}$ represent a set of features arranged in a regular grid with square cells. The edges in the Delaunay triangulation $\mathcal{T}_D$ of $\mathcal{V}$ are the vertical and horizontal connections in the grid plus a set of diagonal edges, each dividing a cell in two congruent triangles. In the case of a regular grid the Delaunay triangulation $\mathcal{T}_D$ is not unique since the boundary points of each cell in the grid are on a circle. Let w be the number of cells in the horizontal direction and h the number of cells in vertical direction and therefore the number of vertices in $\mathcal{V}$ is $n = (w + 1) \cdot (h + 1)$ and the number of vertices on the convex hull of $\mathcal{V}$ is $b = 2w + 2h$. With l_γ the length of all vertical and horizontal connections in the grid, the length of the edges in $\mathcal{T}_D$ dividing a square cell in the grid is $l_\delta = \sqrt{2} \cdot l_\gamma$. The number of edges with length l_γ is $n_\gamma = (w + 1) \cdot h + w \cdot (h + 1)$ and the number of edges of length l_δ is $n_\delta = w \cdot h$. Since $n_\gamma > 2n_\delta$, it follows

$$l_\gamma = \text{median}\left\{l_{ij} : (v_i, v_j) \in \mathcal{T}_D\right\}. \tag{3.10}$$

The Delaunay triangulation of a regular grid with $w = h = 14$ is shown in Figure 3.4(a) and the edge length histogram with peaks for l_γ and l_δ in Figure 3.4(b).

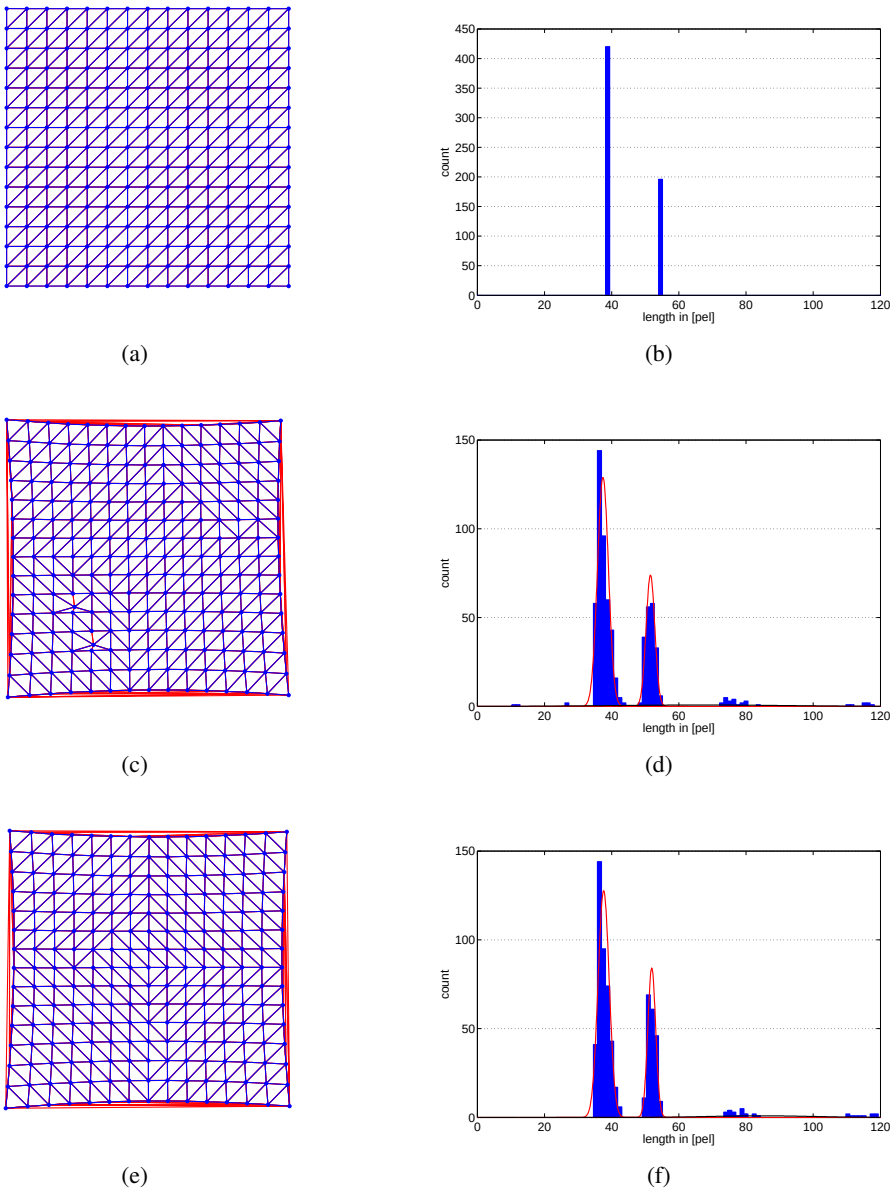


Figure 3.4: Delaunay triangulation and the edge length histogram of (a),(b) a regular grid; (c),(d) laser pattern imaged by the left camera; (e),(f) laser pattern imaged by the right camera.

The image of the laser point pattern projected to a planar surface for the calibration of the laser projector, is not a regular grid due to the laser beam geometry and the perspective mapping of the point pattern by the camera. However, for small pattern deformations the special properties of the Delaunay triangulation of a regular grid still remain true to some degree. Though the length of the edges is not constant anymore it will vary only in a narrow range around the original length. A regular grid is convex and thus has maximum number $b = 2w + 2h$ of vertices on the convex hull. In case of a deformed grid, the number of vertices on the convex hull may be less. However, the minimum number of vertices that form a convex hull of $\mathcal{V}$ is $b = 3$ and thus the maximum number of edges added by the triangulation of the convex hull is $n_C = 2w + 2h - 3$. With

$$n_\gamma = 2wh + w + h > wh + 2w + 2h - 3 \tag{3.11}$$

the approximation $\hat{l}_\gamma$ to l_γ can still be computed by (3.10) and $\hat{l}_\delta = \sqrt{2}\hat{l}_\gamma$ is the approximation to l_δ. The minimum length of edges in $\mathcal{T}_D$ which are introduced by the convex hull of $\mathcal{V}$ is $2l_\gamma$. The mean of the length of edges in the convex hull is denoted by $\bar{l}_\beta$.

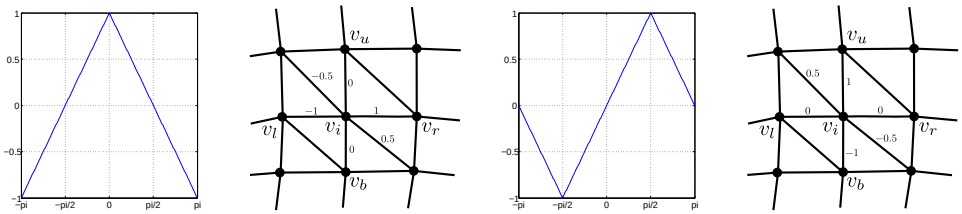
The length l_{ij} of all edges e_{ij} in $\mathcal{T}_D$ has been computed and the edge length distribution is approximated by a Gaussian mixture model with the three Gaussians $G\left(\rho_\gamma, \bar{l}_\gamma, \sigma_\gamma, \rho_\delta, \bar{l}_\delta, \sigma_\delta, \rho_\beta, \bar{l}_\beta, \sigma_\beta\right)$. The mixture proportions ρ_γ, ρ_δ, and ρ_β are estimated from n_C, n_γ, and the overall number of edges in the Delaunay triangulation. The probability of an edge e_{ij} to be a vertical or horizontal connection in the grid is the a-posteriori probability

$$P\left(\gamma|e_{ij}\right) = \frac{\rho_\gamma}{P\left(e_{ij}\right)} \frac{1}{2\pi\sigma_\gamma^2} e^{-\frac{\left(l_{ij}-\bar{l}_\gamma\right)^2}{2\sigma_\gamma}} \tag{3.12}$$

and the probability to be an edge which splits a cell is

$$P\left(\delta|e_{ij}\right) = \frac{\rho_\delta}{P\left(e_{ij}\right)} \frac{1}{2\pi\sigma_\delta^2} e^{-\frac{\left(l_{ij}-\bar{l}_\delta\right)^2}{2\sigma_\delta}}. \tag{3.13}$$

A topological filter based on the edge length distribution function is applied and edges e_{ij} with $P\left(\gamma|e_{ij}\right) + P\left(\delta|e_{ij}\right) < P_{min}$ are discarded. P_{min} is chosen such that only edges of a Delaunay triangulation of the corresponding regular grid are left. The laser point pattern imaged by the left and right camera and its Delaunay triangulations are shown in Figure 3.4(c) and Figure 3.4(e). The corresponding edge length distribution and the Gaussians of the mixture model are plotted in Figure 3.4(d) and Figure 3.4(f). All red edges have been discarded by the topological filter with $P_{min} = 0.1$.



(a) (b) (c) (d)

Figure 3.5: The perpendicularity function and perpendicularity of edges incident to v_i in Delaunay triangulation of (a),(b) p_H and (c),(d) p_V

3.3.2 Feature Ranking

In order to determine the most likely left, right, bottom and upper neighbor of v_i among all vertices $v_j \in \mathcal{V}$ adjacent to v_i in $\mathcal{T}_D$, the *perpendicularity measures* p_H, p_V and $p_\mathcal{N}$ are introduced. For the computation of the perpendicularity, the Delaunay triangulation $\mathcal{T}_D$ is regarded a directed graph with arcs a_{ij} and a_{ji} for each edge e_{ij} in $\mathcal{T}_D$. Let Δx_{ij} and Δy_{ij} be the extension of a_{ij} along the x- and y-axis starting at v_i and $l_{ij} = \sqrt{\Delta x_{ij}^2 + \Delta y_{ij}^2}$ be the length of e_{ij}. The *horizontal perpendicularity* of a_{ij} is defined as

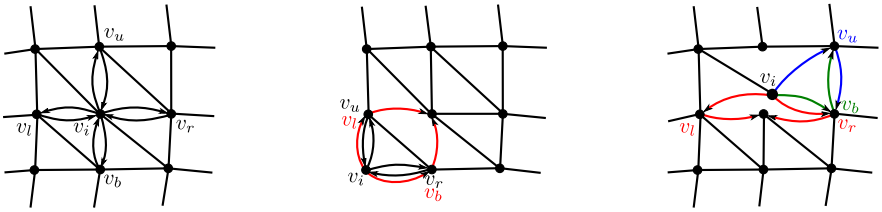
$$p_H\left(a_{ij}\right) = \frac{2}{\pi} \sin^{-1}\left(\frac{\Delta x_{ij}}{l_{ij}}\right) \tag{3.14}$$

and the *vertical perpendicularity* of a_{ij} as

$$p_V\left(a_{ij}\right) = \frac{2}{\pi} \sin^{-1}\left(\frac{\Delta y_{ij}}{l_{ij}}\right). \tag{3.15}$$

Obviously, $p_H\left(a_{ji}\right) = -p_H\left(a_{ij}\right)$ and $p_V\left(a_{ji}\right) = -p_V\left(a_{ij}\right)$. It follows by the definition, that in a regular grid the left/right and bottom/upper neighbors minimize/maximize p_H and p_V. Let v_l, v_r, v_b, and v_u be the the left, right, bottom and upper neighbors of v_i, respectively. The perpendicularity function and an example in the Delaunay triangulation of a slightly deformed grid for both, p_H and p_V is shown in Figure 3.5.

The minimum/maximum value of p_H and p_V is a necessary but not a sufficient condition to be a left/right or upper/bottom neighbor. For example, the left/right and upper/bottom neighbors of the vertices forming the boundary of $\mathcal{T}_D$, are chosen from the minimum/maximum of p_H and p_V even though no such neighbors actually exist. The choice for a vertice v_i is verified by means of a consistency check with all neighbors v_j of v_i. Let v_j be the left neighbor of v_i with minimal p_H. This relation is



(a) (b) (c)

Figure 3.6: Neighborhood of v_i is (a) consistent, (b) inconsistent at the boundary (c) inconsistent due to erroneous feature

consistent if and only if it is symmetric, i.e. v_i is the right neighbor of v_j with maximal p_H. This consistency analysis is performed for the right, the bottom and the upper neighbors of v_i as well. Examples for a consistent neighborhood of v_i, an inconsistent constellation for v_i on the boundary of the grid, and an inconsistent constellation for an erroneous v_i, is illustrated in Figure 3.6.

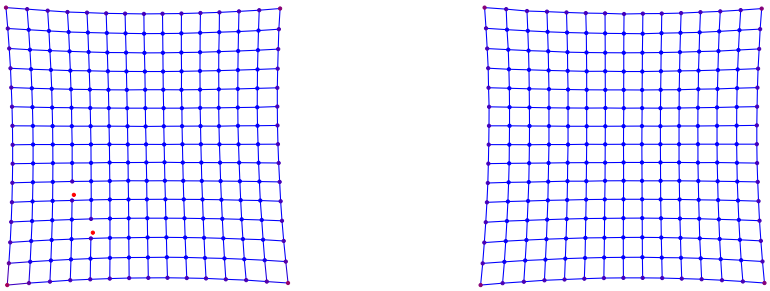
For the consistent neighborhood of v_i and v_j by e_{ij}, the *neighborhood related perpendicularity*

$$p_{\mathcal{N}}(e_{ij}) = \begin{cases} -\min(p_H(a_{ij}), 0) & v_j \text{ is left neighbor of } v_i \\ \max(p_H(a_{ij}), 0) & v_j \text{ is right neighbor of } v_i \\ -\min(p_V(a_{ij}), 0) & v_j \text{ is bottom neighbor of } v_i \\ \max(p_V(a_{ij}), 0) & v_j \text{ is upper neighbor of } v_i. \end{cases} \tag{3.16}$$

defines the degree of validity. It follows, that $0 \leqslant p_{\mathcal{N}} \leqslant 1$. With $\mathcal{N}_i$ the set of valid neighbors of v_i, the quality of v_i is computed as

$$q_v(v_i) = \frac{1}{4} \sum_{j \in \mathcal{N}_i} P(\gamma | e_{ij}) \cdot p_{\mathcal{N}}(e_{ij}) \tag{3.17}$$

combining the validity of all valid neighbors and the probability (3.12) of all corresponding edges. In Figure 3.7 the quality of all features and the probability of the valid edges is drawn color coded. The quality of features on the boundary is lower than the quality of interior features because the lower number of valid neighbors.

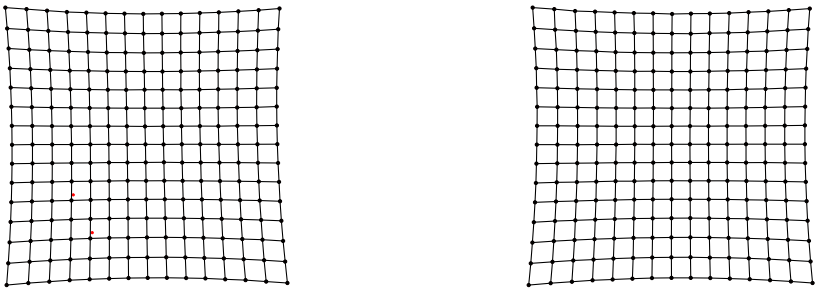


(a) (b)

Figure 3.7: Vertices quality q_v and probability P of valid edges in image of (a) left and (b) right camera; a red vertice/edge means $q_v = 0/P = 0$ and a blue vertice/edge means $q_v = 1/P = 1$

3.3.3 Matrix Reconstruction

The next step is to map each feature to its corresponding ray of the 15×15 pattern. The pattern is represented by a matrix M and the algorithm starts with a matrix $M_{1\times1}$ containing the feature with best quality q_v. Given each features horizontal and vertical neighbors, features are successively added by expansion of M in a breadth-first search fashion. The order of visit is determined by the quality q_v and the unexplored neighbors of a feature added to M are inserted into a priority queue. A proper data structure for priority queues is the *heap*, see e.g. [Cormen et al., 1990, ch. 20,21]. If the final size of M is 15×15, the labeling is straight forward. In all other cases an interactive user input is necessary, specifying the actual boundary of the matrix. In Figure 3.8 the connections of the reconstructed matrices in a stereo image pair acquired during the calibration procedure of a laser projector are shown.



(a) Left camera (b) Right camera

Figure 3.8: Reconstructed pattern matrix.

3.4 Laser Beam Geometry Reconstruction

The actual laser calibration procedure is carried out by the reconstruction of the projector's beam geometry. This requires the representation of the beam by a feasible model and sufficient information in order to extract the model parameters. The representation of the 15×15 laser spot pattern by means of matrix as a result of the labeling algorithm allows for identification of stereo correspondences as well as correspondences among measurements by simple look up operations. The stereo correspondences provide access to the 3-D coordinates of the intersection of a laser ray with a planar surface and the inter-measurement correspondences together with the 3-D coordinates can be evaluated to determine the lines in space which are prescribed by the laser rays. The terminology and the parametrization of the laser beam reconstruction problem is given in Section 3.4.1. An incremental approach to the inital parametrization is presented in Section 3.4.2. The problem is expressed in terms of a global optimization problem in Section 3.4.3 and solved by means of bundle adjustment in Section 3.4.4. The necessary information needed by a point correspondence algorithm in an application of the device is extracted in Section 3.4.5.

With begin of this section a few minor changes in the terminology have to be introduced. Throughout this section upper case letters represent lines and planes, upper case bold letters denote 3-space vectors, and lower case bold letters 2-space vectors, unless specified other. Lines in space are parametrized in terms of Plücker coordinates $L = \left(\mathbf{l}, \bar{\mathbf{l}}\right)$ with $\mathbf{l}$, the direction and $\bar{\mathbf{l}}$, the moment of the line (see Section 2.5.4).

3.4.1 Parameters and Constraints

The geometry of a laser projector is represented by a bundle of lines in 3-space

$$L_i = \left(\mathbf{l}_i, \bar{\mathbf{l}}_i\right),\ i = 1, \ldots, n$$

with $n = 15 \cdot 15$ and the three-dimensional center of projection $\mathbf{X}_c$ where all L_i intersect. The additional unknowns in the calibration procedure are the planes

$$P_j = \left(\mathbf{n}_j, d_j\right),\ j = 1, \ldots, p,$$

each representing the planar surface placed in another depth. The intersection of the laser ray L_i with the plane P_j is denoted by $\mathbf{X}_{j,i}$. The images of $\mathbf{X}_{j,i}$ in the rectified and undistorted left and the right camera are denoted by

$$\mathbf{x}^L_{j,i} \quad \text{and} \quad \mathbf{x}^R_{j,i},$$

respectively. The cameras lines of sight of features $\mathbf{x}^L_{j,i}$ and $\mathbf{x}^R_{j,i}$ are denoted by

$$S^L_{j,i} = \left(\mathbf{s}^L_{j,i}, \bar{\mathbf{s}}^L_{j,i}\right) \quad \text{and} \quad S^R_{j,i} = \left(\mathbf{s}^R_{j,i}, \bar{\mathbf{s}}^R_{j,i}\right).$$

Through the setup of the calibration and the geometric properties of the laser projector, the parameters underlie a number of constraints provided below.

- the camera's lines of sight $S^L_{j,i}$ and $S^R_{j,i}$ of corresponding features $\mathbf{x}^L_{j,i}$ and $\mathbf{x}^R_{j,i}$ intersect in $\mathbf{X}_{j,i}$.

$$\mathbf{s}^L_{j,i} \times \mathbf{X}_{j,i} = \bar{\mathbf{s}}^L_{j,i} \tag{3.18}$$
$$\mathbf{s}^R_{j,i} \times \mathbf{X}_{j,i} = \bar{\mathbf{s}}^R_{j,i} \tag{3.19}$$

- All points $\mathbf{X}_{j,i}$, $i = 1, \ldots, n$, of measurement j form the plane $P_j = (\mathbf{n}_j, d_j)$

$$\forall i\ \mathbf{n}_j \cdot \mathbf{X}_{j,i} = d_j \tag{3.20}$$

- All points $\mathbf{X}_{j,i}$, $j = 1, \ldots, p$ generated by the laser line L_i form a line in space

$$\forall j\ \mathbf{l}_i \times \mathbf{X}_{j,i} = \bar{\mathbf{l}}_i \tag{3.21}$$

- All laser lines L_i, $i = 1, \ldots, n$ intersect in the beam's center of projection $\mathbf{X}_c$

$$\forall i\ \mathbf{l}_i \times \mathbf{X}_c = \bar{\mathbf{l}}_i \tag{3.22}$$

Algorithm 3: Parameter initialization

Data: coordinates and labels of laser spots and camera geometry
Result: rectified coordinates of the laser spots
begin
  compute 3-D points by triangulation of corresponding spots
  compute regression plane through 3-D points of same measurement
  compute intersection of lines of sight with regression planes
  compute lines through 3-D points of same laser ray
  compute point of intersection of all lines
end

3.4.2 Parameter Initialization

The input data of the laser calibration procedure are the coordinates $\hat{\mathbf{x}}^L_{j,i}$ and $\hat{\mathbf{x}}^R_{j,i}$ of the laser spots which have been observed in the images of the stereo camera. For each spot the generating laser ray is known from the labeling. Utilizing the constraints from equations (3.18) – (3.22) the parameters are initially estimated by an incremental algorithm using direct methods outlined in Algorithm 3.

The algorithm starts with the reconstruction of three-dimensional points $\mathbf{X}_{j,i}$ from corresponding laser spots $\hat{\mathbf{x}}^L_{j,i}$ and $\hat{\mathbf{x}}^R_{j,i}$ by triangulation according to (4.11). This involves computing the camera's lines of sight for all features.

All 3-D points which belong to the same measurement form a plane in space as they have been generated by the intersection of the laser beam with a planar surface. The regression plane P_j through all $\mathbf{X}_{j,i}$, with fixed j, is computed using the method introduced in Section 2.5.4.

Further 3-D points $\mathbf{X}^L_{j,i}$ and $\mathbf{X}^R_{j,i}$ are reconstructed by intersection of the lines of sight $S^L_{j,i}$ and $S^R_{j,i}$ with the regression planes P_j. The aim of this step is to project the uncertainty of the observed laser spots into 3-space. Furthermore, this allows for reconstruction of 3-D points from single laser spots where the corresponding spot in the other image does not exist.

The 3-D lines L_i representing each laser ray are approximated using the methodology of Section 2.5.4 and the points $\mathbf{X}^L_{j,i}$ and $\mathbf{X}^R_{j,i}$. Finally, the center of projection $\mathbf{X}_c$ is approximated by the point of intersection of all lines L_i using constraint (3.22).

3.4.3 Projection Model and Prediction Error

The model parameters

$$P_j = (\mathbf{n}_j, d_j) \quad \text{and} \quad L_i = \left(\mathbf{l}_i, \bar{\mathbf{l}}_i\right) \quad \text{and} \quad \mathbf{X}_c,$$

introduced in Section 3.4.1, are combined into the *intersection function* g which computes the intersection

$$\mathbf{X}_{j,i} = g\left(j, i\right), \tag{3.23}$$

of line L_i with plane P_j. The *projection function* f is based on the camera geometry and computes the images

$$\left(\mathbf{x}_{j,i}^L, \mathbf{x}_{j,i}^R\right) = f\left(\mathbf{X}_{j,i}\right) \tag{3.24}$$

of a point $\mathbf{X}_{j,i}$ in 3-space in the left and right camera.

Merging both functions into a single term yields the *projection model*

$$\left(\mathbf{x}_{j,i}^L, \mathbf{x}_{j,i}^R\right) = f\left(g\left(j, i\right)\right), \tag{3.25}$$

which predicts points $\mathbf{x}_{j,i}^{\{L,R\}}$ in the image domain based on the model parameters.

The *prediction error* is defined as displacement

$$\Delta\mathbf{x}_{j,i}^L = \hat{\mathbf{x}}_{j,i}^L - \mathbf{x}_{j,i}^L \quad \text{and} \quad \Delta\mathbf{x}_{j,i}^R = \hat{\mathbf{x}}_{j,i}^R - \mathbf{x}_{j,i}^R \tag{3.26}$$

between observed features coordinates

$$\hat{\mathbf{x}}_{j,i}^L \quad \text{and} \quad \hat{\mathbf{x}}_{j,i}^R$$

and the predicted points

$$\mathbf{x}_{j,i}^L \quad \text{and} \quad \mathbf{x}_{j,i}^R.$$

The prediction error is also referred to as reprojection error.

3.4.4 Bundle Adjustment

Since a direction vector $(\Delta x, \Delta y, \Delta z)$ of fixed length r is uniquely defined by two parameters, all normal vectors $\mathbf{n}_j$ of the planes P_j and all direction vectors $\mathbf{l}_i$ of the laser lines L_i are parametrized in spherical coordinates (r, ϕ, θ) according to the formula

$$\begin{pmatrix} r \\ \phi \\ \theta \end{pmatrix} = \begin{pmatrix} \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \\ \cos^{-1}\left(\left|\frac{\Delta z}{r}\right|\right) \\ \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right). \end{pmatrix} \tag{3.27}$$

Without loss of generality it can be assumed that $r = 1$, and the parameter space can be limited to the northern hemisphere with the zenith $0 \leqslant \phi \leqslant \pi/2$. The arctangent takes into account of the correct quadrant of $\Delta y/\Delta x$ and thus $-\pi < \theta \leqslant \pi$. For computation of the cost function the inverse transformation from spherical coordinates $(1, \phi, \theta)$ to Cartesian coordinates $(\Delta x, \Delta y, \Delta z)$ is computed as

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} \cos(\theta)\sin(\phi) \\ \sin(\theta)\sin(\phi) \\ \cos(\phi) \end{pmatrix} \tag{3.28}$$

Let θ_j^n and ϕ_j^n be the parameters of plane normal $\mathbf{n}_j$ and θ_i^l and ϕ_i^l the parameters of the direction $\mathbf{l}_i$ of laser ray L_i. Then, the state vector is defined by

$$\mathbf{v} = \left(\theta_1^n, \phi_1^n, d_1, \ldots, \theta_p^n, \phi_p^n, d_p, \theta_1^l, \phi_1^l, \ldots, \theta_n^l, \phi_n^l, \mathbf{X}_c\right). \tag{3.29}$$

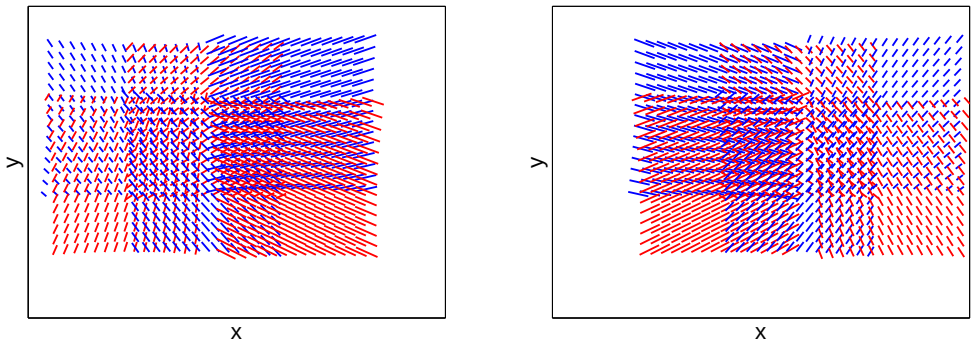
The start values in $\mathbf{v}$ are computed from the initial approximations yielded in Section 3.4.2 and the optimal solution is found by means of the minimization

$$\min_{\mathbf{v}} \|F(\mathbf{v})\|^2 \tag{3.30}$$

with the covariance-weighted sum of squared errors

$$F(\mathbf{v}) = \sum_{j=1}^{p}\sum_{i=1}^{n} \Delta\mathbf{x}_{j,i}^{L\,T} W^L \Delta\mathbf{x}_{j,i}^L + \Delta\mathbf{x}_{j,i}^{R\,T} W^R \Delta\mathbf{x}_{j,i}^R, \tag{3.31}$$

as the error of fit function. The matrices W^L and W^R are the inverse error covariance matrices of the observations $\hat{x}^L$ and $\hat{x}^R$, respectively. The non-linear minimization has been carried out by means of a trust region approach based on the conjugate gradient algorithm [Steihaug, 1983; Nocedal et al., 2000] because of the large number of variables $(2 \cdot n + 3 \cdot p + 3)$ and equations $(2 \cdot n \cdot p)$, and the sparsity of the Jacobian. The minimization algorithms is embedded in a robust framework in order to deal with outliers in the input data. It is assumed that the observations are subject to very few outliers of small scale. The robust framework has been established by means of the Huber M-estimator (see Section 2.6.2).

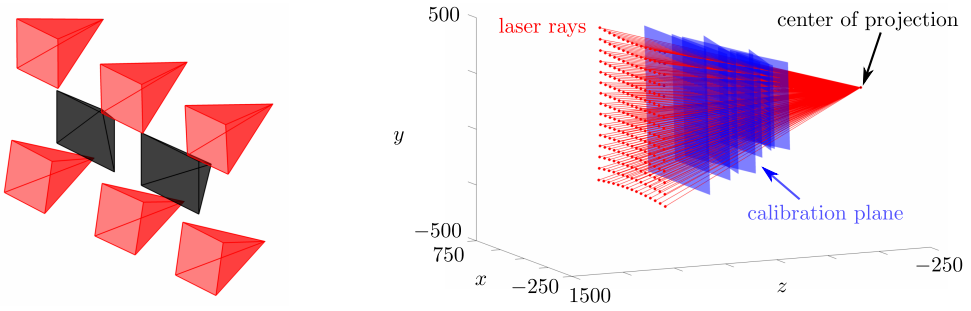


(a) Left camera (b) Right camera

Figure 3.9: Laser calibration segments of all laser projectors.

3.4.5 Calibration Segments

A *calibration segment* C_i is the projection of a depth bounded laser ray L_i to its image in the observing camera. In case of a stereo camera each laser ray L_i belongs the two calibration segments C_i^L and C_i^R. The *depth range* $[z_{min}, z_{max}]$ is input to the calibration procedure and limits the depth where a target object may be positioned such that three-dimensional data of its surface can be acquired. The minimal depth z_{min} and the maximal depth z_{max} refer to the stereo basis, which is equivalent to the x-axis. In an undistorted camera a line in 3D space is imaged to a line in the image domain and a calibration segment is thus uniquely defined by two points. Given the projectors geometry and the depth constraint, the two points $\mathbf{X}_{min,i}$ and $\mathbf{X}_{max,i}$ are computed by intersection of each L_i with a plane perpendicular to the cameras principal axis in distances d_{min} and d_{max}. Then, C_i^L and C_i^R are the segments defined by the projection of $\mathbf{X}_{min,i}$ and $\mathbf{X}_{max,i}$ to the stereo cameras image planes.



(a) Projectors and cameras

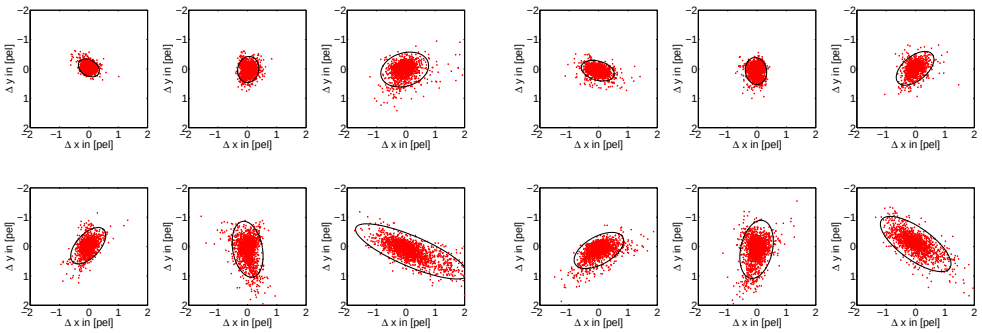
(b) Geometry of single projector in detail

Figure 3.10: Laser projector geometry.

3.5 Experimental Results

The calibration procedure has been carried out for each of the six laser projectors of the illumination unit. The planar surface has been placed in eight positions with increasing depth. The geometry of the whole measurement device and the beam geometry of a single projector (center projector in upper row) together with the calibration plane's positions are shown in Figure 3.10. The red pyramids each represent a laser projector with its center of projection and the boundary of the beam. The black pyramids represent the stereo camera. The center of the world coordinate system is located in the center of projection of the left camera. The geometry of the measurement system defines a minimum distance between an observed object and the stereo basis where the field of view of cameras and projectors begin to overlap. The minimum distance of an object such that the laser spots of all projectors are imaged by both cameras is about 500 mm. The measurement distance of an application developed in Chapter 5 is in the range $[700, 750]$ mm. The calibration segments for this range have been computed and plotted in Figure 3.9. The calibration segments have been plotted in different colors in order to allow for discrimination of overlapping segments.

The uncertainties of the measurement device have been estimated using the observations and the results of the calibration procedure. The errors in the coordinates of the laser spot have been evaluated in Section 3.5.1, and the expected error introduced by initial undistortion and rectification of the image is analyzed in Section 3.5.2. In Section 3.5.3 the beam geometry of all laser projectors is mutually analyzed and the true inter-ray angle of the laser beam is estimated.



(a) Left camera (b) Right camera

Figure 3.11: Reprojection error and 95% error ellipse for each combination of laser projector and camera.

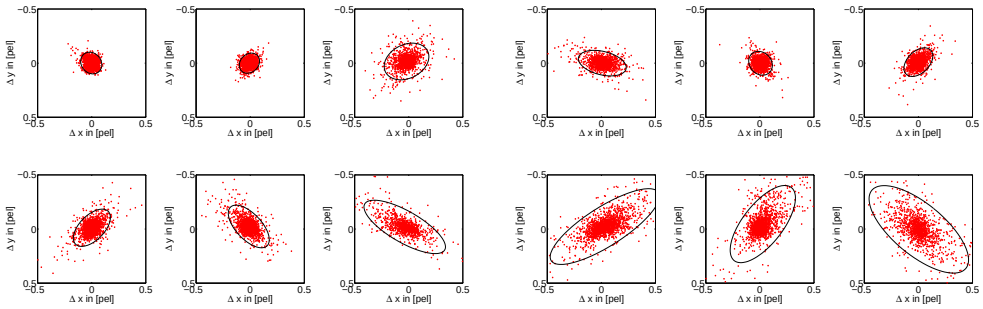
3.5.1 Reprojection Error

The final reprojection error (3.26) that minimizes the error of fit (3.31) is shown in Figure 3.11. In each of the 12 plots the error of the laser spots generated by a single projector in one of the camera images is shown together with the 95% error ellipse of the covariance matrix. A correlation of the first principal component of the error and the direction of the laser calibration segments is due to the deformation of the laser spots in direction of their respective laser calibration segments in the image domain (see Figure 3.9). A higher variance of the error in the spots generated by the projectors in the bottom row is due to a stronger deformation of these spots. This deformation is caused by an increased elevation angle of the laser rays with respect to the calibration plane due to the bent of the projectors in order to overlap with the camera's field of view.

Apart from the analysis of the reprojection errors for the individual projectors, the overall error has been determined by means of three different measures. First, the norm of the overall reprojection error, second the orthogonal distance $d_o\left(\mathbf{x}, C\right)$ between a detected laser spot $\mathbf{x}$ and the corresponding calibration segment C and third, the vertical distance of corresponding laser spots $\left|y^L - y^R\right|$. The errors are specified by their standard deviation, the median error and the maximum value in the 95% confidence interval. All quantities are given in Table 3.1 and the unit is pel. These error measures are utilized by the algorithms in Chapter 4 for deriving an optimal parametrization and for prediction of the expected error in the 3-D coordinates.

Error	σ	median	max in 95% confidence interval
$\|\Delta\mathbf{x}\|$	$\sigma = 0.33$	0.22	0.88
$d_o(\mathbf{x}, C)$	$\sigma_l = 0.21$	0.02	0.38
$\|y^L - y^R\|$	$\sigma_y = 0.31$	0.15	0.65

Table 3.1: Expected errors in the image coordinates of the laser spots. The unit of all values is pel.

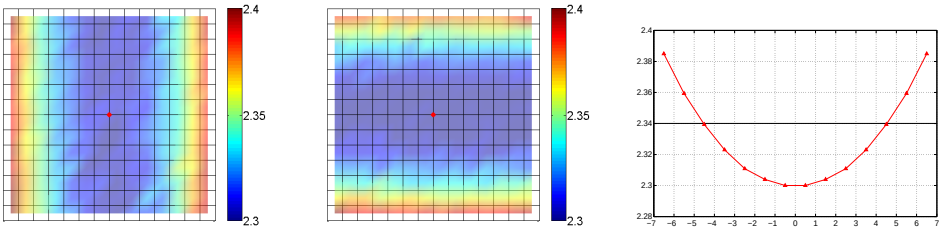


(a) Left camera (b) Right camera

Figure 3.12: Error caused by image rectification and 95% error ellipse for each combination of laser projector and camera.

3.5.2 Image Rectification Error

The coordinates of the laser spots are identified in the original images and subject to the rectification and undistortion transform. The alternative approach of an initial image transformation has been discussed in Section 3.2. The difference between the coordinates of the detected laser spots by the two variants is illustrated in Figure 3.12. The meaning of each individual plots is the same as those depicted in Figure 3.11. The amount of the error is about $1/4$ of the error caused by the laser spot detection algorithm itself and may thus be ignored. However, the other drawbacks of the alternative approach dominate and the proposed variant is preferred.



(a) (b) (c)

Figure 3.13: Interbeam angle function in (a) horizontal direction, (b) vertical direction for a single laser projector; the grid represents the laser pattern. (c) Interbeam angle function averaged over all projectors combining both, horizontal and vertical direction.

3.5.3 Laser Beam Geometry

A further result of the calibration procedure is the actual value of the interbeam angle for each pair of consecutive laser rays in both, vertical and horizontal direction. The interbeam angle for each pair of consecutive laser rays of a single projector has been computed from the laser ray directions that is output of the calibration procedure and the result is shown in Figure 3.13(a) and Figure 3.13(b). Obviously, the angle is not constant over all pairs, though all laser beams show almost constant vertical interbeam angles along each row and constant horizontal interbeam angles along each column. Moreover, due to the symmetry of the generating diffractive grid, the interbeam angle function in horizontal and vertical direction is the same. The function has been computed as average over all vertical and all horizontal interbeam angles and all projectors and is shown in Figure 3.13(c). The average interbeam angle over all pairs of consecutive rays in the presented device is 2.33°.

4 3-D Measurement with a Laser Projector

With the methods presented in the previous chapter the epipolar geometry and the uncertainties of the trifocal measurement device are made available to a superior application interested in three-dimensional data acquisition. The input to such an application is a sequence of image pairs acquired by the stereo camera while the surface of the object of interest is illuminated with the laser point pattern. In order to allow for the separation of the illuminated pattern from the image background structures the sequence should provide image pairs with and without laser illumination in an alternating manner.

The aim of this chapter is to develop and evaluate the algorithms which are required for gaining three-dimensional point data from the stereo images acquired by the measurement device introduced in the previous chapter. A rough outline of the procedure that is carried out for each image pair is outlined in Algorithm 4 which consists of three major steps. Each section of this chapter is devoted to one of these steps. Two different approaches for detection of the laser spots in an image are introduced in Section 4.1. The different settings also led to specific requirements for the point correspondence algorithm. Both variants are covered in Section 4.2. The point correspondences are the input to the actual computation of three-dimensional points. The triangulation for the undistorted and rectified stereo camera and the feature error propagation is treated in Section 4.3. Finally, simulation results and the derived theoretical performance of the algorithms are discussed in Section 4.4.

Algorithm 4: Compute 3-D point cloud from stereo image pair

Data: image pair $\left(I_i^L, I_i^R\right)$
Result: set of 3-D points $\{\mathbf{x}_k\}$
begin
  laser spot detection in both images
  identify point correspondences
  three-dimensional triangulation
end

4.1 Laser Spot Detection

The first step of the reconstruction of a three-dimensional point cloud from a stereo image pair acquired by the stereo camera is the localization of the bright spots (features). These have been generated in the camera images by the light of the laser projector reflected on the surface of the observed object. Two different strategies are pursued in parallel and their particular properties are discussed in the two paragraphs of this section. Both strategies have their respective advantages and a final conclusion is given in Chapter 5. The feature extraction algorithms are independently applied to the images of both, the left camera and the right camera. Hence, the indices $_L$ and $_R$ are dropped throughout this section.

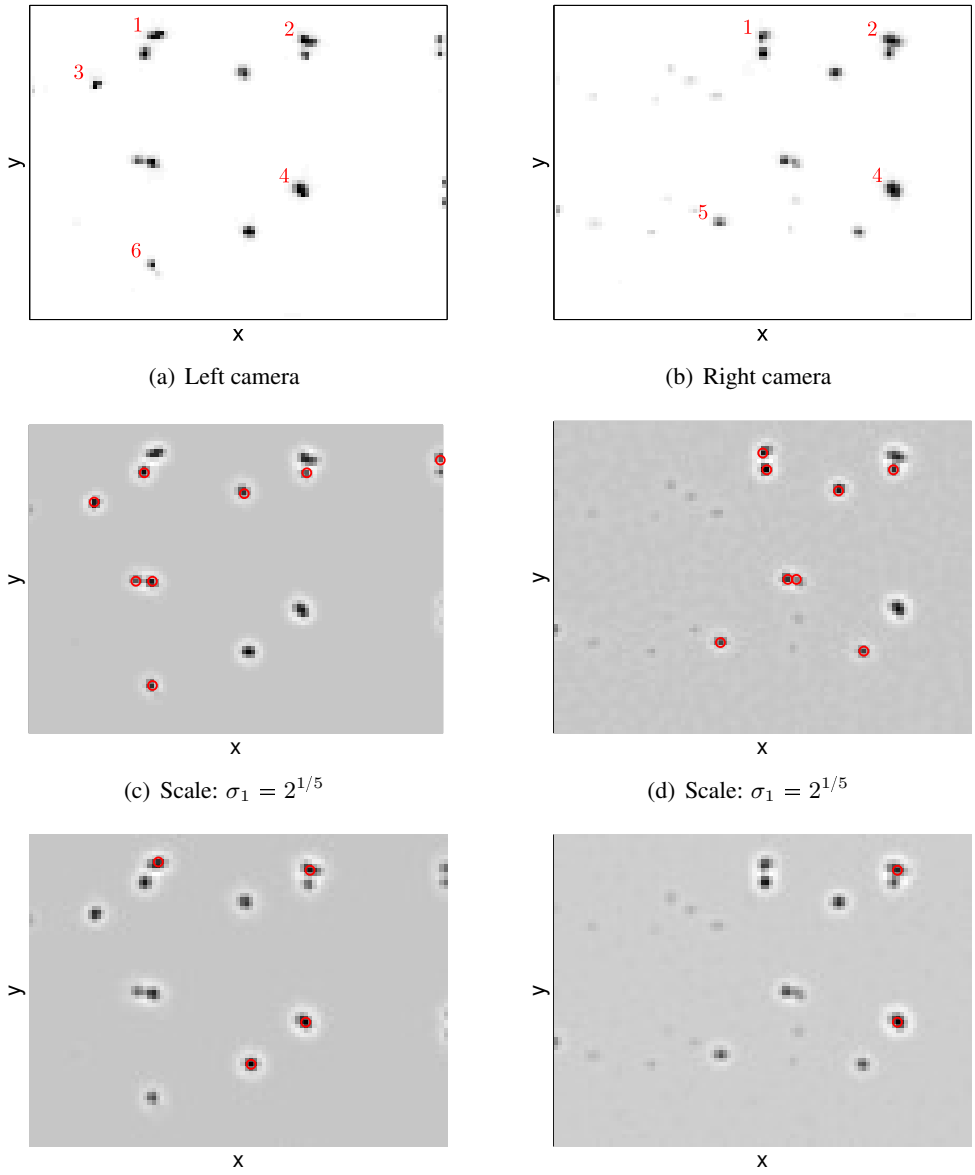
Initially, the images are preprocessed in order to separate the laser spots from the background structures using the difference image approach as previously introduced in Section 3.2 in connection with the laser calibration. Any image I_i in the sequence with laser illumination applied at acquisition time is assumed to be preceded and succeeded by images I_{i-1} and I_{i+1} without laser illumination applied at acquisition time. The input image to the feature detection algorithm is computed as

$$\hat{I}_i = 2 \cdot I_i - I_{i+1} - I_{i-1} \tag{4.1}$$

This operation may introduce effects at the objects boundary in case the observed object is subject to motion. Bright background structures close to the object boundary may cause suppression of laser spots in the region that has changed between two consecutive images due to the objects motion. This effect can be ignored if the objects motion is slow compared to the frequency of image acquisition.

4.1.1 Image Domain

In the first approach, the laser spots are detected in the two-dimensional image domain (ID) according to Algorithm 2 in Section 3.2. The difference of Gaussian blob detector (see Section 2.2) is applied to the image $\hat{I}_i$. The blob centers are detected at local intensity minima in the image stack $D(x, y, \sigma)$ which consists of five DoG response images where scales $\sigma_k = 2^{k/5}$, $k = 1, \ldots, 5$ have been chosen. The results for the corresponding cropped image regions acquired by the left and the right camera are shown in Figure 4.1. All features were found in the scales σ_1 and σ_2. Some features in the original images have been labeled by numbers. The features with number 1, 2, and 4 have in common that they are all spot clusters which have been generated by multiple laser rays due to the overlapping laser beams of the projection unit. This limitation of the ID approach leads to loss of correspondences as will be shown later. The features labeled with number 3, 5, and 6 do not have a counterpart in the other image. Correspondences for such points are lost as well.



(a) Left camera

(b) Right camera

(c) Scale: $\sigma_1 = 2^{1/5}$

(d) Scale: $\sigma_1 = 2^{1/5}$

(e) Scale: $\sigma_2 = 2^{2/5}$

(f) Scale: $\sigma_2 = 2^{2/5}$

Figure 4.1: Cropped original images (inverse) and features detected at different scales.

4.1.2 Calibration Segment Domain

The limitation of the spot detection in the image domain and the availability of the laser calibration segments (see Section 3.4.5) motivates the search for features solely in a calibration segment's domain (CD). A similar approach introduced by [Popescu et al., 2006] is improved by applying the difference of Gaussian approach to the one-dimensional profile image which is extracted from the intensities in the neighborhood of a laser calibration segment.

For each laser calibration segment L, the one dimensional profile image I_L is extracted from the input image I along L. The discretization of I_L is chosen uniformly with size $l = 2 \cdot \lceil \|L\| \rceil$, in order to reduce aliasing effects. The gray value $g(x)$ for each pixel at position x within I_L is found by bilinear interpolation from the four closest pixels to the position $L(x/l)$ on L in the input image I. The profile image I_L can be extracted either directly from the rectified image, or it is generated from the original image by applying the inverse rectification transform to the coordinates of the discretized calibration segment L.

In order to detect laser spots of varying size, for each profile image I_L a 1-D version of the scale invariant difference of Gaussian approach as introduced in Section 2.2 is applied. The profile image is successively convolved with Gaussians g_σ with increasing scale σ to

$$G_L(x, \sigma) = (g_\sigma * I_L)(x). \tag{4.2}$$

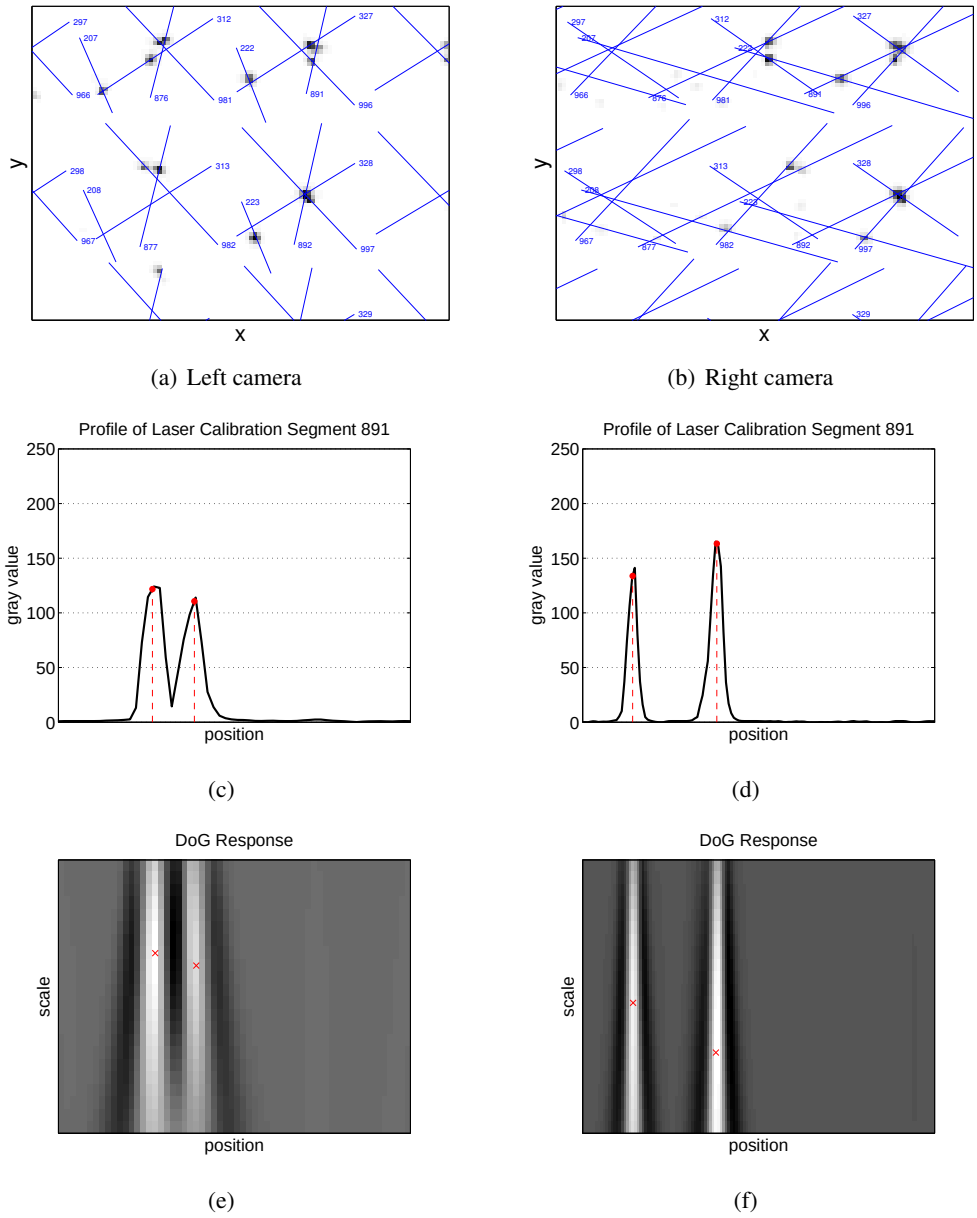
The difference of Gaussian $D(x, \sigma(i))$ is computed as finite differences of consecutive response images

$$D(x, \sigma(i)) = G_L(x, \sigma(i)) - G_L(x, \sigma(i+1)) \tag{4.3}$$

with $-1 \leqslant i \leqslant s+1$. This results in a two dimensional response image

$$R = \begin{pmatrix} D(1, \sigma(-1)) & \cdots & D(l, \sigma(-1)) \\ \vdots & & \vdots \\ D(1, \sigma(s)) & \cdots & D(l, \sigma(s)) \end{pmatrix} \tag{4.4}$$

with size $(s+2) \times l$. The features are found at local maxima of R with respect to the 8-neighborhood. A subpixel correction is carried out using the vertex position of a parabola fitted to each response in R. Finally, the position of the features in the original image I are determined from the position of L. In Figure 4.2 the detail of an image with laser calibration segments, the profile image for a laser calibration segment, and the response of the difference of Gaussian function is shown.



(a) Left camera

(b) Right camera

(c)

(d)

(e)

(f)

Figure 4.2: (a), (b): Cropped original images (inverse) and laser calibration segments labeled with numbers. (c), (d): Intensity profile along calibration segment 891 and local maxima. (e), (f): Response image R of difference of Gaussian function with local maxima.

4.2 Feature Correspondences

The identification of corresponding laser spots detected in a stereo image pair is the key to the reconstruction of three-dimensional point data. The different reflectivity of the surface with respect to the left and the right camera results in varying intensity, shape and size of corresponding laser spots. This gives rise to an approach that solely relies on the features coordinates and the epipolar geometry. It has been mentioned in the previous section that missing and erroneous features will lead to an incomplete matching. For this reason and since the mapping between the two point sets in the left and right camera image is in general not affine the PPM algorithms of Section 2.3.1 do not apply.

The two different feature detection algorithms presented in the previous section also lead to two variants of the correspondence algorithm. Both algorithms have in common that the orthogonal distance between the features and the epipolar geometry is evaluated and correspondences are found within a narrow tolerance band near the epipolar lines in the camera images. The identification of point correspondences for features detected in the image domain and calibration segment domain is treated in Section 4.2.1 and Section 4.2.2, respectively.

4.2.1 Image Domain

The search for correspondences between features detected in the image domain is split into five major steps listed in Algorithm 5. The input to the algorithm are the feature coordinates, the epipolar geometry of the stereo camera, and the calibration segments of the laser rays. Let

$$\left\{\mathbf{x}_i^L\right\}_{i=1}^m \quad \text{and} \quad \left\{\mathbf{x}_j^R\right\}_{j=1}^n$$

with

$$\mathbf{x}_i^L = \left(x_i^L, y_i^L\right) \quad \text{and} \quad \mathbf{x}_j^R = \left(x_j^R, y_j^R\right)$$

denote the coordinates of the features detected in the left and right camera image, respectively. The depth bounded calibration segments C_k^L and C_k^R are represented by their first and last points

$$\left\{\mathbf{c}_{k,0}^L, \mathbf{c}_{k,1}^L\right\} \quad \text{and} \quad \left\{\mathbf{c}_{k,0}^R, \mathbf{c}_{k,1}^R\right\}$$

for $k = 1, \ldots, 1350$. The normals $\mathbf{n}_k^L$ and $\mathbf{n}_k^R$ with unit length $\|\mathbf{n}_k\| = 1$ are initially computed for all calibration segments.

Algorithm 5: Point correspondences

Data: features in left and right camera image and epipolar geometry
Result: pairs of corresponding features
begin
  feature labeling
  identify feasible correspondences
  weighted bipartite graph representation
  weighted bipartite matching
end

Feature Labeling

In the first step of the correspondence algorithm the correspondences between features and calibration segments are established. Such correspondences are pairs (i, k) and (j, k) which satisfy the inequality relation

$$d_o\left(\mathbf{x}_i^L, C_k^L\right) < d_l \quad \text{and} \quad d_o\left(\mathbf{x}_j^R, C_k^R\right) < d_l \tag{4.5}$$

where $d_o\left(\mathbf{x}, C_k\right)$ is the orthogonal distance of a point $\mathbf{x}$ with the calibration segment C_k. The orthogonal distance is given by

$$d_o\left(\mathbf{x}, C_k\right) = \left|\mathbf{n}_k \cdot \left(\mathbf{x} - \mathbf{c}_{k,0}\right)\right| \tag{4.6}$$

The *labeling distance* d_l is a threshold that limits the complexity of the correspondence algorithm and simultaneously allows for small errors in the feature coordinates. The optimal value for d_l is determined in Section 4.4.1. The distance to the closest endpoint is used if the orthogonal projection of $\mathbf{x}$ is outside the calibration segment's domain.

Feasible Correspondences

In the second step feasible point correspondences are identified using the previous labeling information and the camera epipolar geometry constraints. As the feature coordinates refer to their position in the rectified and undistorted camera, the epipolar lines are parallel to the x-axis. A pair $(i, j)_k$ is thus picked as feasible point correspondence if both pairs, (i, k) and (j, k) exist for a k and the vertical distance between the features

$$\left|y_i^L - y_j^R\right| < 2d_l \tag{4.7}$$

is less than twice the labeling distance. If several such k exist for a pair (i, j), k is chosen to minimize the sum of all distances between features and calibration segments.

Weighted Bipartite Graph Representation

The point correspondence problem is now transformed into a bipartite matching problem as introduced in Section 2.3.2. Let the features be represented by an abstract set $\mathcal{V} = \mathcal{V}_L \cup \mathcal{V}_R$ with disjoint subsets

$$\mathcal{V}_L = \{v_i\}_{i=1}^{m} \quad \text{and} \quad \mathcal{V}_R = \{v_j\}_{j=1}^{n}\,.$$

The bipartite graph representation of the correspondence problem is given by

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \tag{4.8}$$

with edges $\mathcal{E} = \{e_{ij}\}$ defined for each feasible point correspondence $(i,j)_k$.

If each $v_i \in \mathcal{V}_L$ and each $v_j \in \mathcal{V}_R$ is incident to at most one edge in $\mathcal{E}$, an unambiguous set of point correspondences is found given by $\mathcal{E}$. Due to the orientation and the position of the calibration segments, many features will appear in pairs (i,k) and (j,k) for more than a unique k. This leads to ambiguous correspondences. However, the vertices which are interconnected by such edges may be limited to a small subset. The graph $\mathcal{G}$ is split into its connected components by means of a breadth-first search. All unambiguous components are left as is and the ambiguous components are treated further as follows.

A weighted bipartite graph is established by assigning weights w_{ijk} to the edges $e_{ij} \in \mathcal{E}$. The weight w_{ijk} is computed according to

$$w_{ijk} = w_{ij} \cdot w_{ik} \cdot w_{jk} \tag{4.9}$$

with normalized Gaussian weighted distances w_{ij}, w_{ik}, and w_{jk} representing the deviation of the features v_i and v_j from the epipolar geometry. The vertical distance of features v_i and v_j is reflected by

$$w_{ij} = e^{-\frac{\left(y_i^L - y_j^R\right)^2}{2\sigma_y^2}}$$

and the distance of v_i and v_j with respect to the calibration segments C_k^L and C_k^R by

$$w_{ik} = e^{-\frac{d_o\left(\mathbf{x}_i^L, C_k^L\right)^2}{2\sigma_l^2}} \quad \text{and} \quad w_{jk} = e^{-\frac{d_o\left(\mathbf{x}_j^R, C_k^R\right)^2}{2\sigma_l^2}}\,,$$

respectively. The Gaussian widths have been chosen from the results of Section 3.5.1 where σ_y is the standard deviation of the vertical distance between corresponding laser spots and σ_l is the standard deviation of the orthogonal distance between laser spots and the calibration segments.

Bipartite Matching

Finally, a subset $\mathcal{E}_M \subset \mathcal{E}$ is determined such that all vertices $v_i \in \mathcal{V}_L$ and $v_j \in \mathcal{V}_R$ are incident to at most one edge $e_{ij} \in \mathcal{E}_M$ and the sum of weights of the edges in $\mathcal{E}_M$ is maximal. Given the weighted bipartite graph $\mathcal{G}$ and the information about connected components, the solution is the maximum weighted bipartite matching in $\mathcal{G}$ (see Section 2.3.2). The matching is found by means of the Hungarian algorithm applied to each connected component in $\mathcal{G}$.

The vertices which are not covered by the matching have been rejected due to their stronger deviation from the epipolar geometry compared to other competing feasible correspondences. Either the correct correspondence is not available as the feature has not been detected or an erroneous correspondence wins if the competing correct correspondence has a stronger deviation from the epipolar geometry. Such erroneous correspondences lead to 3-D errors as shown in Section 4.3.2.

4.2.2 Calibration Segment Domain

The identification of corresponding features which have been detected in the domain of corresponding calibration segments is not as complex as the previous algorithm. A labeling is not necessary since all features are inherently labeled by their originating calibration segment. These unique labels allow for search of correspondences solely between features which have been detected in the domain of corresponding calibration segments. Let

$$\left\{\mathbf{x}_i^L\right\}_{i=1}^m \quad \text{and} \quad \left\{\mathbf{x}_j^R\right\}_{j=1}^n$$

be the image coordinates of the features detected in the domain of corresponding calibration segments. A pair (i, j) is a correspondence if their y-coordinates meet (4.7). Though each pair of corresponding calibration segments induces at most one correspondence, the true correspondence can not be identified if multiple pairs satisfy (4.7). Assuming that for the majority of calibration segments at most one correspondence is found, the erroneous correspondences are stored and identified in a later stage.

The calibration segment based correspondence search facilitates unscrambling of the laser spot clusters discussed in Section 4.1.1 and enables for more correspondences. A spot which appears as a unique feature in the image domain appears in all calibration segments that intersect the spot. Assuming the laser calibration segments to be error free, the error in the feature coordinates is limited to an error in the direction of the calibration segments. The scale space dimension reduction and the simplified search for correspondence reduces the complexity of feature detection. The drawback of this approach is a higher number of erroneous correspondences.

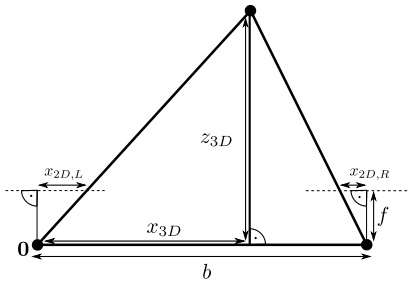


Figure 4.3: Triangulation in a rectified camera.

4.3 3-D Points from Correspondences

For all pairs of feature correspondences three-dimensional points are computed by means of triangulation using the epipolar geometry of the stereo camera. The point in question is the third point in the triangle which defines the epipolar plane as shown in Figure 2.1. The formulas used for computing the 3D coordinates are given in Section 4.3.1 and the propagation of errors in the feature coordinates to the error in the 3-D points is studied in 4.3.2.

4.3.1 3-D Triangulation

A point $\mathbf{x}_{3D}$ in 3-space is assumed to be imaged by the cameras to points $\mathbf{x}_{2D}^L$ and $\mathbf{x}_{2D}^R$ satisfying the equations

$$\mathbf{x}_{2D}^L = \mathbf{P}^L \mathbf{x}_{3D} \quad \text{and} \quad \mathbf{x}_{2D}^R = \mathbf{P}^R \mathbf{x}_{3D}, \tag{4.10}$$

where $\mathbf{P}^L$ and $\mathbf{P}^R$ are the camera matrices of the left and the right camera, respectively. In general, these equations can be combined into a form $\mathbf{A}\mathbf{x}_{3D} = 0$ and be solved by singular value decomposition of $\mathbf{A}$ [Hartley & Zisserman, 2004]. In the particular case of a rectified stereo camera with the stereo basis perpendicular to the principal axis of both cameras, $\mathbf{x}_{3D}$ can be computed using simple similarity relations in the triangles defined by the camera geometry. From the geometry depicted in Figure 2.1(b) and Figure 4.3 it follows that the coordinates are related by the equations

$$z_{3D} = f\frac{b}{d} \qquad \text{and} \qquad x_{3D} = x_{2D}^L \frac{b}{d} \qquad \text{and} \qquad y_{3D} = y_{2D}\frac{b}{d} \tag{4.11}$$

where $d = x_{2D}^R - x_{2D}^L$ is called the *disparity*.

4.3.2 3-D Errors

In practice, the feature coordinates are subject to errors which may be caused by inhomogeneous reflectivity of the observed object, noise in the camera electronics, and limitations of the correspondence algorithm. The error in the image coordinates introduce an error to the 3-D coordinate according to (4.11). Let a pair of corresponding features with true horizontal position x_{2D}^L and x_{2D}^R, true vertical position y_{2D}, and true disparity $d = x_{2D}^R - x_{2D}^L$ be given. The originating 3-D point is given by $\mathbf{x}_{3D} = (x_{3D}, y_{3D}, z_{3D})$. The measured coordinates may be subject to a disparity error Δd and an error Δy_{2D} orthogonal to the epipolar line. The disparity error is the sum of the errors Δx_{2D}^L and Δx_{2D}^R in the left and the right image, respectively. For $\Delta x_{2D}^L = -\Delta x_{2D}^R$ the disparity error vanishes even though the coordinates are erroneous.

Let the *depth error* be the deviation Δz_{3D} from the true depth z_{3D} and the *lateral error* be the pair of deviations $(\Delta x_{3D}, \Delta y_{3D})$. With (4.11) the depth error is related to the disparity and the disparity error by

$$\Delta z_{3D} = -f \frac{b}{d} \left(\frac{\Delta d}{d + \Delta d} \right). \tag{4.12}$$

The lateral error is given by

$$\Delta x_{3D} = -x_{2D}^L \frac{b}{d} \left(\frac{\Delta d}{d + \Delta d} \right) + \Delta x_{2D}^L \frac{b}{d + \Delta d} \tag{4.13}$$

and

$$\Delta y_{3D} = -y_{2D} \frac{b}{d} \left(\frac{\Delta d}{d + \Delta d} \right) + \Delta y_{2D} \frac{b}{d + \Delta d}. \tag{4.14}$$

For a stereo camera setup with fixed f and b both, the depth error and the lateral error depend on the true depth z_{3D} of the true 3-D point. The term

$$\frac{b}{d} \left(\frac{\Delta d}{d + \Delta d} \right)$$

is quadratic in the reciprocal of the disparity and thus quadratic in the depth according to (4.11). The lateral errors additionally depend on the error in the respective coordinate with gain $\Delta d / (d + \Delta d)$ which is linear in the depth.

4.4 Experimental Results

The performance of the algorithms developed in this section has been studied by means of computer simulations. The settings of the simulated measurement device are chosen to meet the requirements of the application discussed in Chapter 5. The point correspondence algorithm for features detected in the image domain is evaluated in Section 4.4.1 and the error in the three-dimensional coordinates due to errors in the coordinates of corresponding features with respect to the affecting parameters is visualized in Section 4.4.3.

4.4.1 Correspondences in the Image Domain

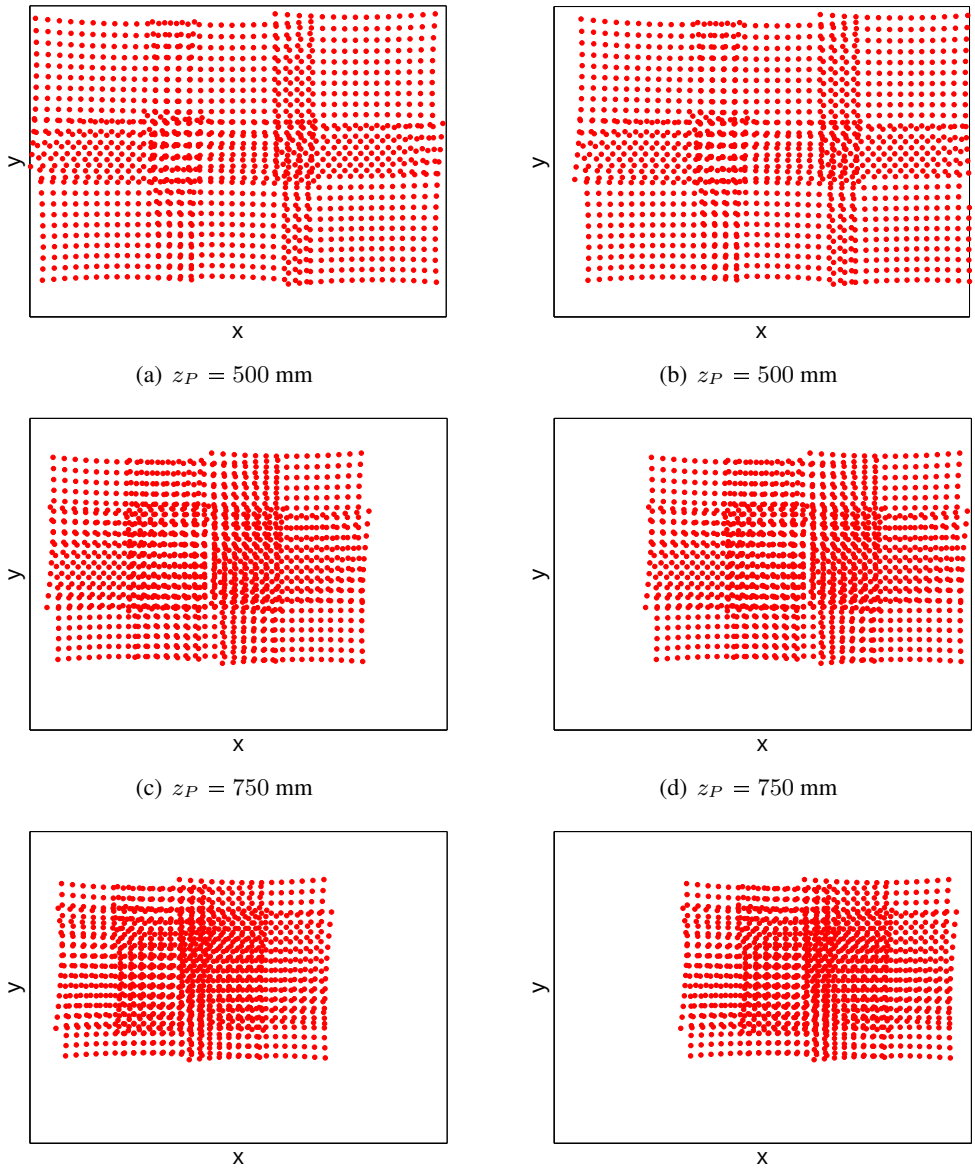
The point correspondence algorithm introduced in Section 4.2.1 does not consider the observed object's shape and the actual feasibility of a correspondence. This may lead to erroneous correspondences and hence loss of correct correspondences. The aim of this paragraph is to quantify the expected error and identify the optimal parametrization of the algorithm. Erroneous correspondences particularly resolve from laser spots closely located next to each other in the image domain for which the feature coordinates are subject to errors. Due to characteristics of the measurement device, the density of the laser spot pattern in the camera's images depends on the depth of the observed object. The pattern that would be imaged by the stereo camera if a planar surface illuminated by the laser projector is placed in depths $z_P = 500$ mm, $z_P = 750$ mm, and $z_P = 1000$ mm is shown in Figure 4.4.

The effect of the the labeling distance d_l and the density of the laser spot pattern with increasing depth to the performance of the point correspondence algorithm presented in Section 4.2.1 is studied by means of a Monte-Carlo simulation. A virtual plane is placed in equidistant depths

$$z_P = 700 \text{ mm}, 701 \text{ mm}, \ldots, 750 \text{ mm}$$

oriented parallel to the rectified camera's image planes. The range is chosen minimal with respect to the application in Chapter 5. The intersection of the laser rays with the virtual plane is computed and the 3-D points are imaged to the undistorted and rectified stereo camera. The 2-D points are subject to a random displacement according to the measurement errors estimated in Section 3.5.1. It is assumed that the effect of spot clusters does not occur and all individual laser spots generated by the laser projector are found. In each depth a number of $N = 100$ perturbed measurements are generated and the correspondence algorithm is carried out for the labeling distances

$$d_l = 0.5, 0.6, \ldots, 2.5.$$



(a) $z_P = 500$ mm (b) $z_P = 500$ mm

(c) $z_P = 750$ mm (d) $z_P = 750$ mm

(e) $z_P = 1000$ mm (f) $z_P = 1000$ mm

Figure 4.4: Pattern generated by a planar surface placed in various depths. Left column: images of the left camera. Right column: images of the right camera.

The results of the simulation are illustrated in Figure 4.5. The curves in the upper row show the relation of the labeling distance and the internal numbers of the correspondence algorithm. The values have been computed by joint evaluation of all N measurements in all depths z_P for fixed d_l. The solid curves are the median value of all results and the upper and lower dashed curves are the maximum and minimum value, respectively. In Figure 4.5(a) the number of correct correspondences is plotted as a function of the maximum labeling distance d_l. The number of connected components in $\mathcal{G}$ and the maximum size of any connected component in a measurement are depicted in Figure 4.5(b) and Figure 4.5(c), respectively.

The graphs show that a small value of d_l leads to loss of correspondences whilst a growing d_l gives rise to more ambiguities. The reason for loss of correspondences is the labeling process. Features are not assigned their appropriate calibration segment if the error is larger than the labeling distance. In turn, a high value of the labeling distance causes more ambiguities because more calibration segments are in the labeling range of a feature. With growing d_l the number of correct correspondences increases up to saturation for $d_l \approx 1.5$. The optimal parametrization of d_l is the minimum value such that a further increment would not lead to significantly more correct correspondences. A further increment would lead to more ambiguities and hence requires more runtime.

The optimal value of $d_l = 1.5$ is derived from the detailed view on the results for

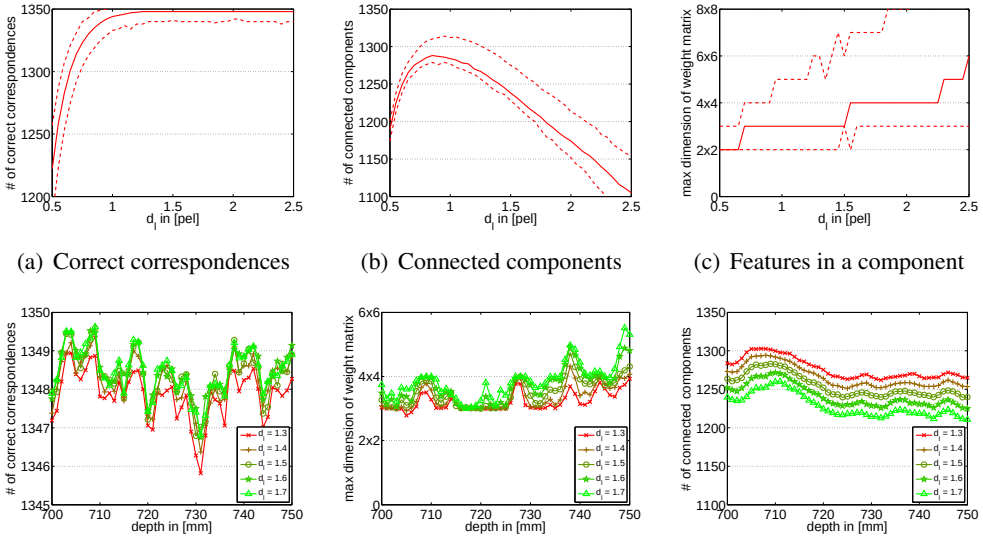
$$d_l = 1.3, 1.4, 1.5, 1.6, 1.7$$

in the bottom row of Figure 4.5. Each graph depicts the number of correct correspondences (Figure 4.5(d)), the number of connected components (Figure 4.5(e)) and the maximum number of elements in a connected component for a fixed d_l in all depths z_P. The oscillation of the curves is caused by the intersection of the calibration segments in specific depths.

The maximum number of correct correspondences is $15 \cdot 15 \cdot 6 = 1350$. The expected number of correct correspondences with $d_l = 1.5$ is 1348. In 39% of all measurements the maximum number of 1350 correspondences has been found. Even in the worst case (2 measurements) only 9 correspondences – less than 1% – are lost. Let N be the number of features, M the minimum number of connected components, and $n \times n$ the maximum size of a connected component. Since any feature is present in exactly one component, the maximum expected runtime with the Hungarian algorithm applied to resolve ambiguous correspondences is

$$\left\lceil \frac{N-M}{n-1} \right\rceil n^3 \tag{4.15}$$

and is thus quadratic in the maximum component size $O\left(n^2\right)$. For the depth range $[700, 750]$ mm with $N = 1350$, $M = 1228$, and $n = 6$ the expected runtime is 5400.



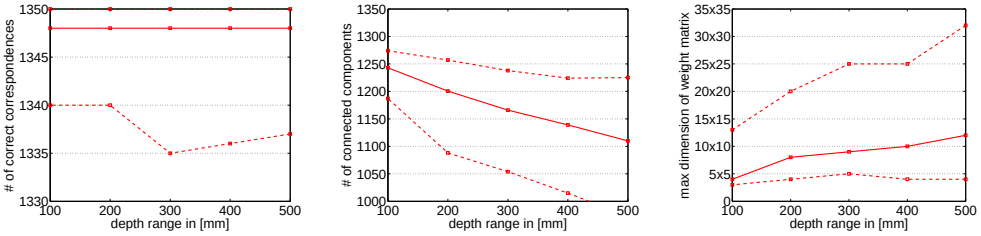
(a) Correct correspondences (b) Connected components (c) Features in a component

(d) (e) (f)

Figure 4.5: Upper row: minimal (dashed), median (solid), and maximal (dashed) values for varying labeling distance d_l. Bottom row: median values for varying depth z_P and varying labeling distance d_l depicted by multiple graphs.

The simulations carried out so far have focused on the depth range $[700, 750]$ mm. The simulation have been carried out again for wider depth ranges and same optimal labeling distance $d_l = 1.5$. The graphs in Figure 4.6 show the joint results for 100 measurements in various depths within the range with the same y-axis as in the upper row of Figure 4.5. Even though the number of connected components decreases and their size increases, the expected number of correct correspondences remains almost the same. With a wider range the calibration segments expand and the ambiguities increase due to labeling of more features by the same calibration segment. A wider depth range thus leads to a growth of the algorithm's complexity.

Because of their extension of several pixels in the image domain, a laser spot will cluster with its neighbors and form a single spot if they are located close enough to each other. For such clusters the algorithm of Section 4.1.1 will not be able to detect the individual spots and hence correspondences will be lost. The effect of spot clustering has been analyzed for several depths by means of data clustering of the true imaged 2-D coordinates. The graphs in Figure 4.7 show the lower and the upper limit of the number of clusters containing more than a single spot for depths between 500 mm and 1000 mm. The lower limit assumes a cluster distance of 1.5 pel and the upper limit of 3.0 pel.



(a) Correct correspondences (b) Connected components (c) Features in a component

Figure 4.6: Performance for various ranges of depth and labeling distance $d_l = 1.5$.

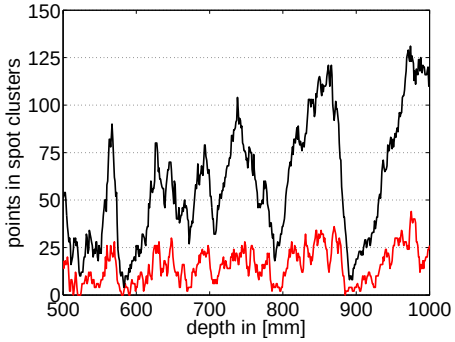
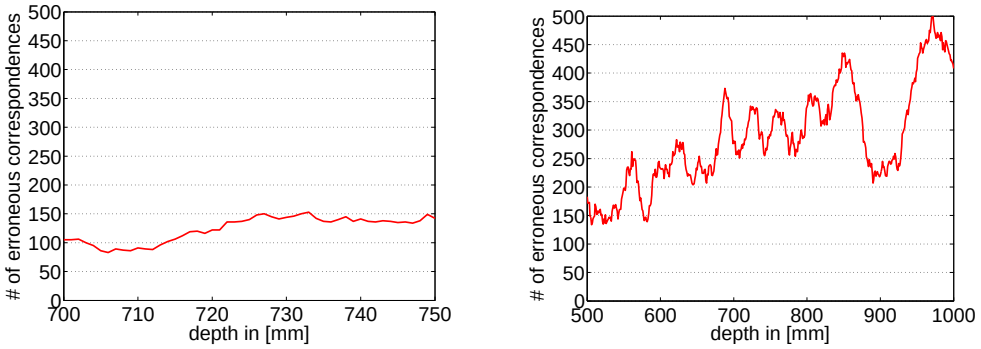


Figure 4.7: Number of spot clusters for maximum inter-point-distance 1.5 pel (lower graph) and 3.0 pel (upper graph).



(a) 50 mm depth range (b) 500 mm depth range

Figure 4.8: Erroneous correspondences for the calibration segment based algorithm.

4.4.2 Correspondences in the Calibration Segment Domain

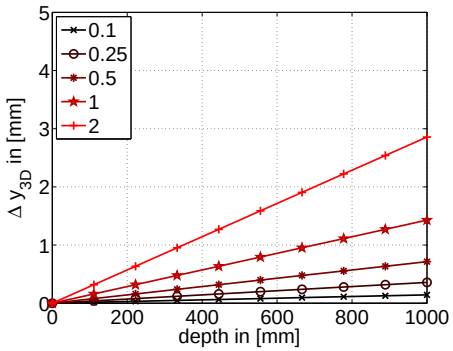
The performance of the point correspondence algorithm for features detected in the domain of the calibration segments is evaluated by analyzing the number of erroneous correspondences leading to outliers in the 3-D points. The 2-D coordinates of the laser spots are generated using the same simulation environment as in the previous paragraph. The laser spots are assumed to have an extension of approximately 3 pel and thus the labeling part of Algorithm 5 can be used to determine those features that would be intersected by an calibration segment. The number of erroneous correspondences can be easily determined by the difference of the number of all detected correspondences and the actual number of correspondences. The results are illustrated for the depth ranges $[700, 750]$ and $[500, 1000]$ in Figure 4.8. The graphs show that a wider depth range leads to more erroneous correspondences. For the 50 mm wide depth range up to 153 erroneous correspondences are introduced (approximately 10%) and for the 500 mm wide range up to 499 (approximately 27%). The erroneous correspondences lead to outliers in the 3-D data and may thus have a negative impact to the following model fit procedure.

4.4.3 Feature Error Propagation

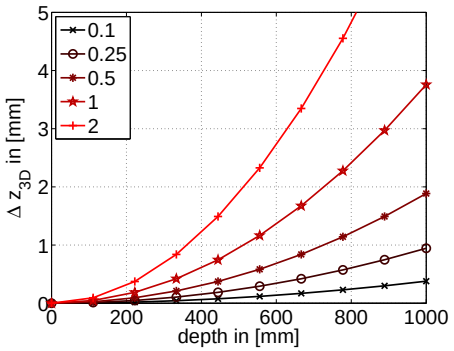
The propagation of the error in the feature coordinates discussed in Section 4.3.2 has been studied for the measurement device presented in Chapter 3. The stereo basis of the stereo camera is $b = 377.31$ mm and the focal length of both rectified cameras is $f = 700$ pel. The principal points of the left and the right camera are

$$\mathbf{p}^L = (400, 500) \quad \text{and} \quad \mathbf{p}^R = (1000, 500)\,,$$

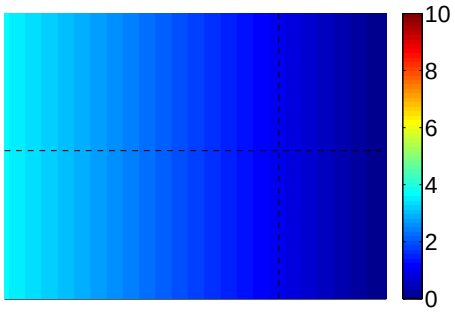
respectively. The expected error in the feature coordinates has been estimated in Section 3.5.1 and is limited to maximal $\Delta = 2$ pel. The influence of the depth to the error components independent of the feature's position within the image is depicted in Figure 4.9. Both plots consist of five graphs, each illustrating the influence of depth to the 3-D error for a fixed feature coordinate error. Figure 4.9(c) and Figure 4.9(d) show the error in the x- and y-components of the 3-D coordinates for an error of 1 pel in both, the x- and y-components of the image coordinates. The overall 3-D error for a disparity error of 1 pel is shown in Figure 4.9(e) and with an additional error of 1 pel in the features y-coordinate in Figure 4.9(f). A disparity error of 2 pel leads to a depth error of up to 5 mm in a depth between 700 mm and 750 mm. The overall 3-D error even surpasses this value in the image corners. The accuracy in the feature's coordinates are thus of crucial importance for the reconstruction of 3-D point cloud.



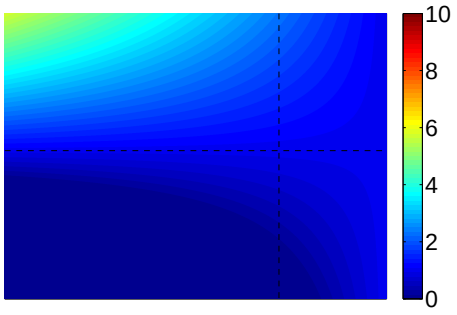
(a) Error in y_{3D} for varying Δy_{2D} with $\Delta d = 0$



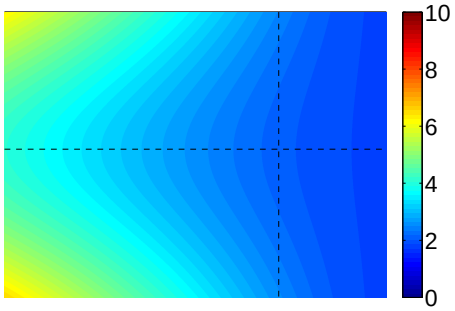
(b) Error in z_{3D} for varying Δd



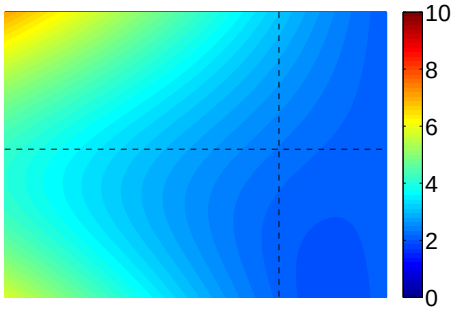
(c) Error in x_{3D} in mm



(d) Error in y_{3D} in mm



(e) Overall 3-D error for $\Delta y_{2D} = 0$ pel



(f) Overall 3-D error for $\Delta y_{2D} = 1.0$ pel

Figure 4.9: Propagation of errors in the feature's coordinates to 3-D errors. (a), (b): 3-D errors independent of the position within the image. (c), (d): 3-D errors for $\Delta x^L_{2D} = \Delta d = 1.0$ pel and $\Delta y_{2D} = 1.0$ pel. (e), (f): Overall 3-D errors in mm for $\Delta x^L_{2D} = \Delta d = 1.0$ pel.

NUR NICHT MATT WERDEN,
SONST KOMMT MAN UNTERS RAD.

(Hermann Hesse)

5 Surface Reconstruction of a Wheel

The focus of this chapter is on the reconstruction of the surface and the trajectory of a moving wheel from a sequence of sparse point clouds. The point clouds are acquired by the measurement device presented in Chapter 3 using the methodology developed in Chapter 4. The depth range of the measurement device is limited to the extension of the object's surface plus a certain tolerance. The wheel is represented by a surface of revolution utilizing the curve representation and approximation techniques of Chapter 2. A global surface model and the trajectory is recovered in a global optimization framework incorporating all point clouds.

The entire reconstruction process consists of the four stages illustrated in Figure 5.1. In the first stage covered in Section 5.1 initial model parameters for each point cloud of the sequence are estimated. The robustness of the proposed algorithm in presence of noise and outliers is studied using simulated data. The second stage defines an order of the scattered data points based on an initial global surface model that is established by smoothing of all points. The third and the fourth stage are carried out iteratively, alternating between global surface approximation and trajectory recovery. The entire algorithm involving stage two, three, and four is presented in Section 5.2.

In Section 5.3 the uncertainty of the surface model and the trajectory reconstructed from real measurements is studied by means of statistical model assessment methods (see Section 2.8). The improvement that is achieved by the global surface model is quantified by comparison of the uncertainties of the worst and the best initial model in a sequence with the uncertainties of the global model.

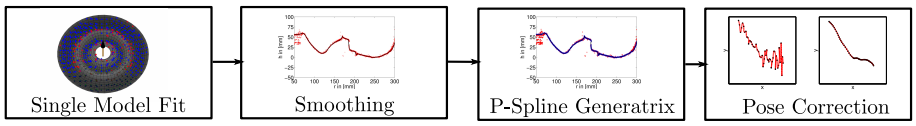


Figure 5.1: Flow Diagram

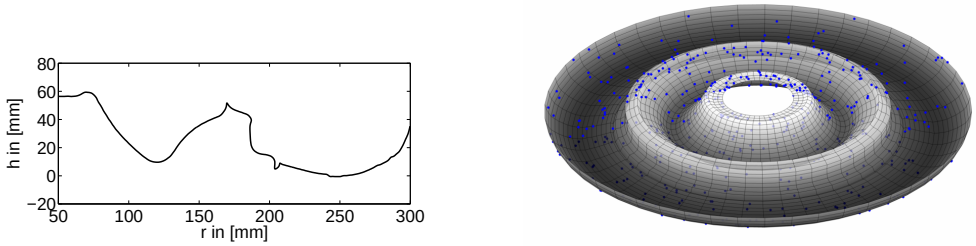
Algorithm 6: Reconstruction from single point cloud

```
Data: sparse point cloud {x_k}_{k=1}^m
Result: initial model M = (S, P)
begin
    m_r ⟵ ⌈0.8 · m⌉
    /* LMedS loop */
    for i ← 1 to 100 do
        randomly select m_r points from the input
        compute surface normals
        compute initial pose parameters
        compute initial surface parameters
        /* iteratively reweighted least squares */
        while model parameters change do
            compute residuals
            compute weights using M-estimator
            weighted non-linear optimization of all model parameters
        compute error of fit F_i
        save model M_i
    choose best model M = argmin_{M_i} median F_i
end
```

5.1 Reconstruction from Single Point Clouds

In this section the algorithm for estimation of an initial model from a sparse point cloud is presented. The observed object is a wheel represented by surface of revolution. The input to the algorithm is a sparse three-dimensional point cloud $\{\mathbf{x}_k\}_{k=0}^{m}$ and a neighborhood relation $\mathcal{N}$ of the points. The point cloud has been acquired by means of the measurement device presented in Chapter 3 and the methodology of Chapter 4. The output is an initial model in terms of Section 2.5.1 which consists of a curve representing the generatrix of the surface of revolution, the model center and an axis of symmetry.

The algorithm runs through four major steps. In step one, two, and three an initial model $\mathcal{M}_0 = \{\mathcal{S}_0, \mathcal{P}_0\}$ is estimated incrementally using direct methods. In the fourth step the model parameters are fitted simultaneously using the initial values of $\mathcal{M}_0$. The model fit is embedded in a robust estimation framework that facilitates rejection of outliers. The least median of squares approach is used in an outer loop and step four is carried out in an iteratively reweighted least squares loop using the Huber M-estimator. Each step of the algorithm is treated in one of the paragraphs of this section.



(a) Generatrix

(b) Surface with generated point cloud

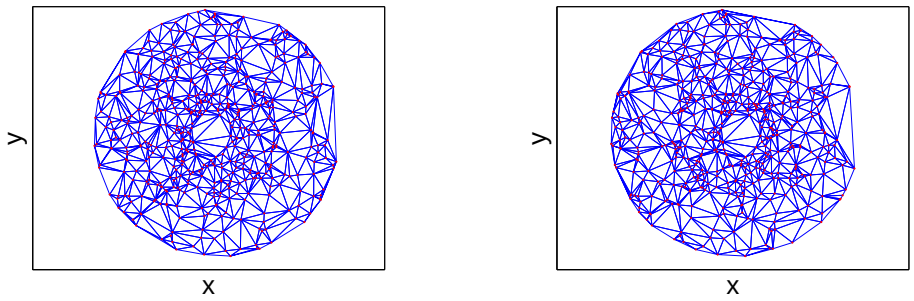
Figure 5.2: The surface of revolution studied by means of a Monte-Carlo simulation.

5.1.1 Surface Normal Estimation

It was shown in Section 2.5.5 that the symmetric axis of a surface of revolution can be computed from a sampling of surface normals. If the surface normals can not be measured but a point cloud $\{\mathbf{x}_k\}_{k=1}^{m}$ of the surface is available, the surface normals may be estimated from these points.

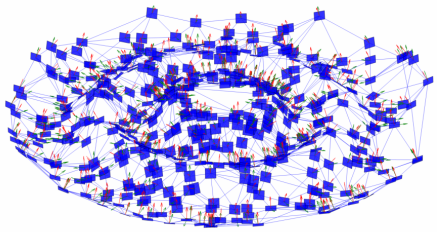
The normal $\mathbf{n}_k$ at a point $\mathbf{x}_k$ on the surface may be approximated by the normal of the regression plane through all points in the local neighborhood $\mathcal{N}(\mathbf{x}_k)$ of $\mathbf{x}_k$ [Hoppe et al., 1992]. The method introduced in Section 2.5.4 is feasible for this approach. If the neighborhood relation is not given it needs to be reconstructed from the points. The complexity of neighborhood reconstruction for a set of unordered points grows with the dimension of the points. A popular method for grid generation of scattered data in two-dimensional space is the Delaunay triangulation (see Section 2.2.3). The *quickhull algorithm* is the n-dimensional generalization of the Delaunay triangulation[Barber et al., 1996]. As the surface sampling is generated by the method of Chapter 4, the neighborhood is defined using the Delaunay triangulation of the feature coordinates in the images of the stereo camera.

The accuracy of a normal estimated by the plane regression method depends on (1) the errors in the point samples, (2) the curvature of the underlying manifold, (3) the density and the distribution of the points, and (4) the local neighborhood size [Mitra et al., 2004]. A fifth quantity having an effect on the accuracy is the position of the object with respect to the measurement device as occlusions may cause distant points in 3D space to be neighbors in the camera image with respect to the Delaunay triangulation. The sparsity of the point cloud implicate a great distance of adjacent points. In order to minimize the influence of distant points to the estimated surface normals, the size of the neighborhood is chosen minimal. This minimum size is established using only direct neighbors.



(a) Left camera (b) Right camera

Figure 5.3: Projection of a generated point cloud to the image planes of the stereo camera with neighborhood defined by the Delaunay triangulation.



(a)

Figure 5.4: True surface normals (green) and surface normals approximated by local regression planes (red).

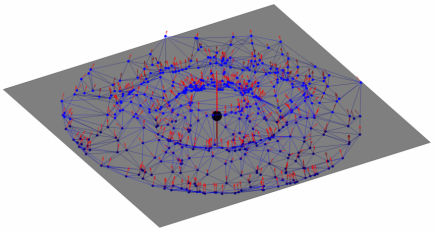


Figure 5.5: Regression plane, symmetric axis estimated from estimated surface normals, and and model center by intersection of the plane and the symmetric axis.

5.1.2 Pose Initialization

The pose initialization procedure intends to find a good approximation $\mathcal{P}_0$ of the true pose parameters for a point cloud $\{\mathbf{x}_k\}_{k=1}^m$ representing a surface of revolution. In the previous section it was shown that the surface normals can be computed from a point sampling with a certain accuracy. Given the surface normals, the symmetric axis $\mathcal{A} = (\mathbf{a}, \bar{\mathbf{a}})$ of the surface can be estimated applying the method presented in Section 2.5.5. Finally, the model center $\mathbf{x}_C$ is the point of intersection between the symmetric axis and the regression plane through all data points.

Errors in the observational data, such as noise and outliers may affect the precision of this procedure. The error arising from the pose initialization procedure has been estimated by a computer simulation using typical conditions of a real measurement. The sampled data points have been corrupted using combinations of

- Gaussian noise with $\sigma = \{0, 1, 2, 3, 4, 5\}$ mm, and
- outliers (0%, 10%, 20%) equally distributed in $h \in [-25, 75]$ mm.

For each combination of the above listed values $1,000$ measurements have been generated with each 500 points randomly sampled from the surface shown in Figure 5.2. Each measurement is placed at a random position within the field of view of the stereo camera such that at least 50% of the surface is within the field of view. The image coordinates have been computed by projection of the three-dimensional points into the image plane using the cameras epipolar geometry. The true symmetric axis of the object is constant and parallel to the principal axis of the cameras.

The pose initialization procedure has been applied to each measurement and the errors of the symmetric axis and the model center have been evaluated by comparison of the input and the computed pose. The pose error is quantified by a radial error and an axial error. The *radial error* Δr is the offset of the computed model center from the true model center in orthogonal direction to the true symmetric axis and the *axial error* α is the deviation of the direction of the computed symmetric axis from the correct axis. The median error curves are shown Figure 5.6. All plots consist of four graphs, each showing the relation of pose error and noise for a fixed ratio of outliers. Apart from the overall error over all measurements the error has also been computed for the subset of measurements with 100% surface coverage ($\approx 50\%$ of all measurements) and for the complementary set of measurements with partial surface coverage.

Both, the axial error and radial error increase with stronger noise and higher ratio of outliers. The error of measurements with 100% surface coverage is significantly below that of measurements with partial coverage. The measurement error due to outliers is obviously higher than the error caused by noise in the data points. The reason for this

effect is the error introduced by the normal vector estimation. An outlier is involved in the surface normal estimation of all data points in its neighborhood with respect to the Delaunay triangulation. Hence, even a small number of outliers may corrupt a large number of the surface normals. For this reason, some effort needs to be spent in order to reduce the impact of outliers to the surface normals. Even uncorrupted data may cause errors in the surface normals as a local regression plane computed in the neighborhood of a sparse point cloud is does not capture strong curvature in the true shape. The robust estimation methods introduced in Section 2.6.1 are feasible for a robust pose initialization. The least median of squares method with 100 random samplings from the original point cloud is carried out for data with outliers.

Even though the pose initialization procedure may introduce a large error to the parameters of the initial model, the algorithm introduced in Section 5.1.4 may be able to find the correct model. It is shown that a large initial pose error can be corrected with a considerable likelihood.

5.1.3 Surface Initialization

In this paragraph the surface is initialized by approximation of a generatrix to the points in *surface coordinates*. Using the initial pose parameters $\mathcal{P}_0$, the point cloud $\{\mathbf{x}_k\}_{k=1}^m$ is transformed into model coordinates according to the equations (2.49) and (2.51). A further transform into two-dimensional surface coordinates is established by

$$r = \sqrt{x_{\mathcal{M}}^2 + y_{\mathcal{M}}^2} \tag{5.1}$$

$$h = z_{\mathcal{M}} \tag{5.2}$$

where r is the radial distance from the axis of symmetry and h is the height with respect to the model center. The initial surface $\mathcal{S}_0$ is a curve approximated to the points (r_k, h_k) in surface coordinates.

Various errors have an effect to the distribution of the points in surface coordinates. Due to the potentially large errors in the initial pose parameters, the surface coordinates may be subject to a systematic displacement. Furthermore, the point cloud may be perturbed by noise and contaminated by outliers due to erroneous correspondences. The curve approximation algorithm must be able to deal with such errors. The initial surface should establish a sufficient smoothing and provide a thorough representation of the true underlying shape. At the same time overfitting caused by too many parameters must be avoided and the number of parameters should be kept small in order to facilitate robustness and fast convergence in the following non-linear model fit procedure.

The B-spline curve approximation using only few intervals provides an effective and numerically well-conditioned way to find such a curve with implicit smoothing

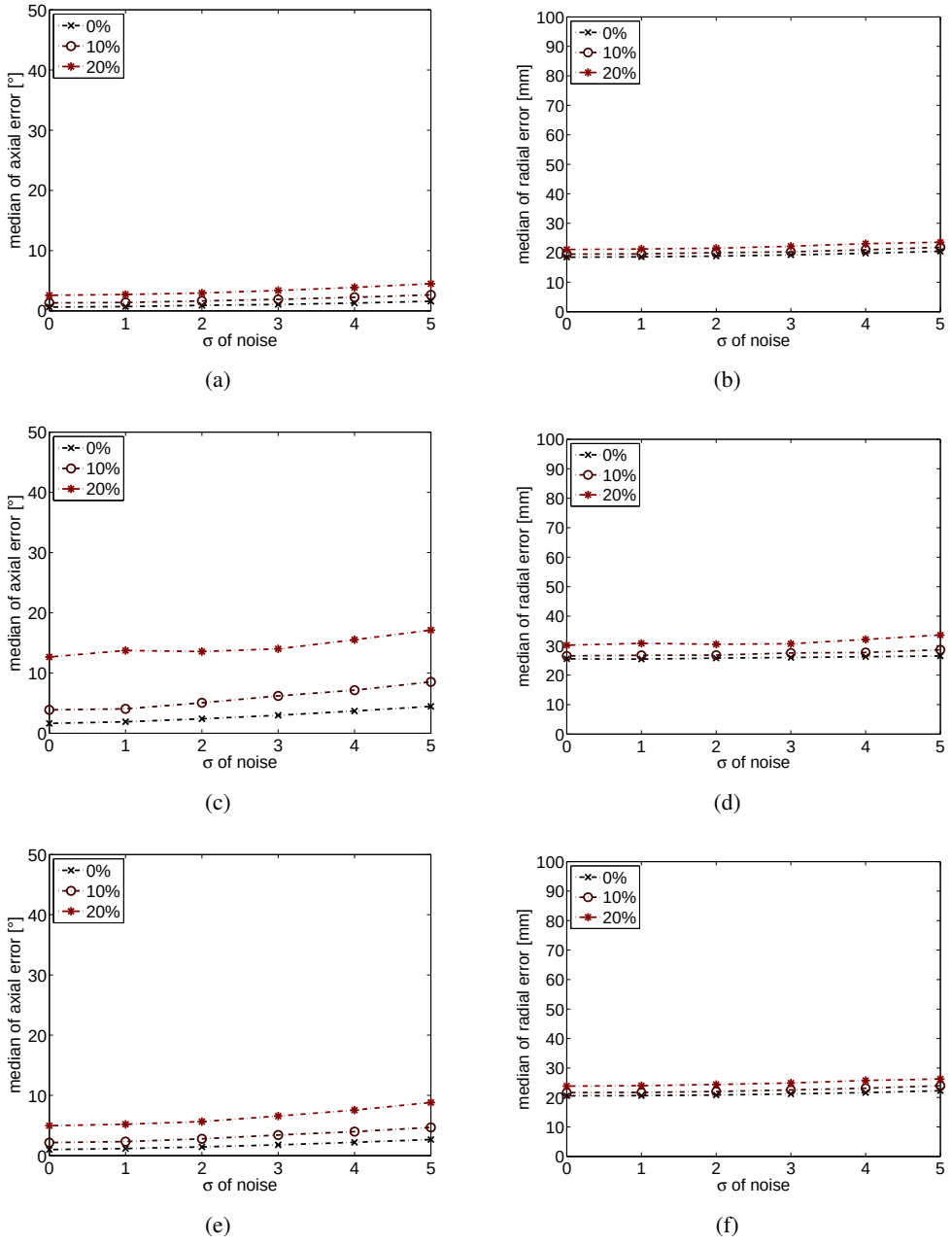


Figure 5.6: Median of axial and radial error after model initialization for various degrees of noise and outliers. The diagrams show the errors for (a),(b) measurements with 100% surface coverage; (c),(d) measurements with partial surface coverage; (e),(f) all measurements.

of the data in surface coordinates. The optimal number of intervals with respect to the convergence characteristics of the subsequent model fit procedure has been identified by means of a Monte-Carlo simulation and is discussed in the following section.

For the error in the initial model center the radial and the axial component are distinguished. The axial component is the error parallel to the true axis of symmetry and the radial component is the error perpendicular to the true symmetric axis. The axial component of the center error causes a shift of the generatrix in the direction of the axis and thus does not affect the error of the generatrix with the data points. Special attention has to be payed only to *axial error* and the *radial error*. The axial error is the angle between the correct and the estimated axis of symmetry, and the radial error is the distance between the estimated and the true model center in radial direction, with respect to the true axis of symmetry.

In Figures 5.9 and 5.10 at the end of this section, the effect of the radial and the axial error to the distribution of the data points in surface coordinates is illustrated for various error values. A shift of the model center causes a shift of the data points along the r-axis in surface coordinates, which leads to a increased variance of the data in the y-component. A tilt of the model causes the points to drift apart in h-direction with increasing distance from the model center. The effect of the axial error also depends on the angle β between shift direction and direction of the axial error as shown in the right column of Figure 5.10. The surface initialization may output a infeasible generatrix in case the error in the initial pose parameters is too strong. The details of this effect and the limits are discussed in the next section.

5.1.4 Modelfit for Sparse Pointclouds

The model fit algorithm simultaneously finds the optimal parameters of the model $\mathcal{M}$ for a given point cloud $\{\mathbf{x}_k\}_{k=1}^m$ and initial model parameters $\mathcal{M}_0 = \{\mathcal{S}_0, \mathcal{P}_0\}$ by means of non-linear optimization. The error of fit measure is the sum of squared distances between the data points and the model $\mathcal{M}$

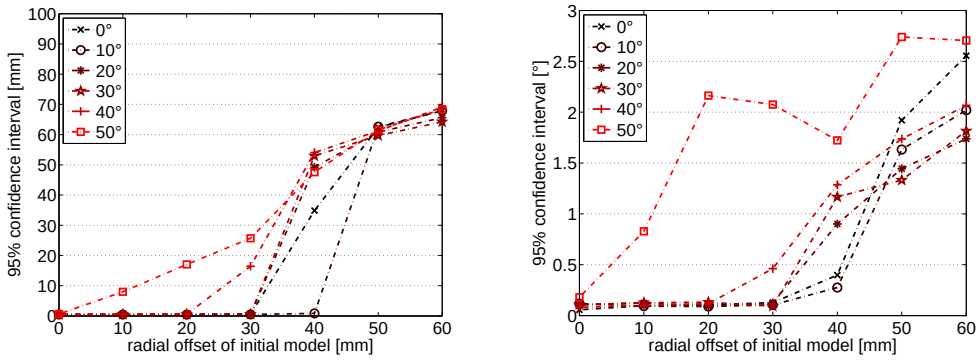
$$F_{\mathcal{M}} = \sum_{k=1}^{m} d^2\left(\mathbf{x}_k, \mathcal{S}\right) \tag{5.3}$$

where the distance measure

$$d\left(\mathbf{x}, \mathcal{M}\right) = h - \mathcal{S}\left(r\right) \tag{5.4}$$

is the vertical distance between the data points in surface coordinates and the generatrix. The minimization problem

$$\min_{\mathcal{M}} F_{\mathcal{M}} \tag{5.5}$$



(a) Radial error (b) Axial error

Figure 5.7: Limits of the model initialization with initial pose offset. The graphs are the maximum errors of a measurement in the 95% confidence interval.

is solved by means of non-linear least squares using the Levenberg-Marquardt algorithm. The parameters of the algorithm are two components which encode the direction of the axis of symmetry and the B-spline parameters. The algorithm has been embedded in an iteratively reweighted least squares framework using the Huber M-estimator. Finally, the model center is placed such, that the generatrix intersects the x-axis at $x = 250$.

Depending on the shape of the underlying surface and the accuracy of the model initialization the model fit algorithm may converge to a local minimum and thus fail to find the correct pose and surface parameters. The robustness of the model fit algorithm has been evaluated for the surface shown in Figure 5.2 by a computer simulation taking into account of the pose and axis errors of the model initialization procedure estimated previously in Section 5.1.2. Error free point clouds with 500 points have been sampled from a from the surface $\mathcal{S}$ with a given pose $\mathcal{P}$. The initial pose $\mathcal{P}_0$ that is input to the model fit algorithm is the original pose $\mathcal{P}$ perturbed by any possible combination of

- radial shift of the model center $\Delta r = (0, 10, 20, 30, 40, 50, 60)$ mm, and
- tilt of the axis of symmetry $\alpha = (0°, 10°, 20°, 30°, 40°, 50°)$.

For each combination of the above pose errors the model fit procedure has been applied to 1000 point samples. The effect of a perturbed initial pose to the model fit result is shown in Figure 5.7. The graphs are the upper bound of the 95% confidence interval of the radial error in Figure 5.7(a) and the axial error in Figure 5.7(b). In general, the axial error in the resulting model remains small but increases slightly with higher initial radial error. The effect of the initial axial error to the final axial error is

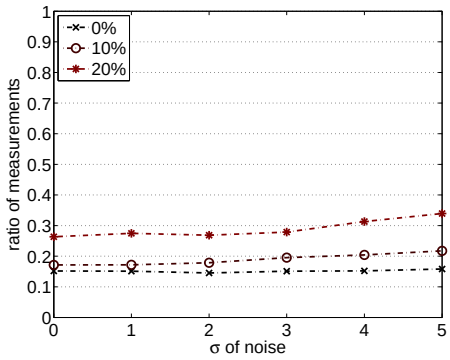
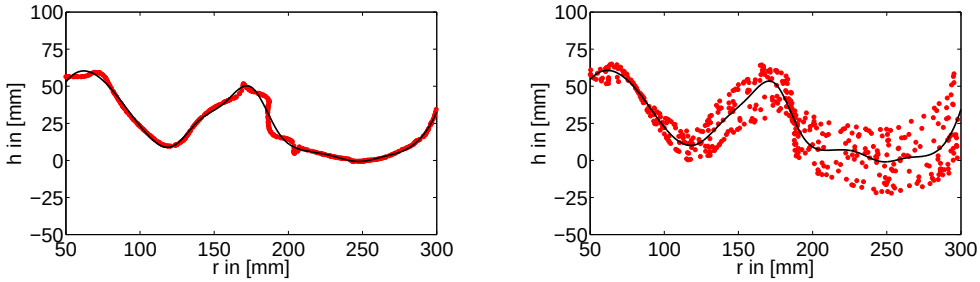


Figure 5.8: Expected error free measurements with initial radial error $\Delta r > 30$ mm or axial error $\alpha > 30^\circ$

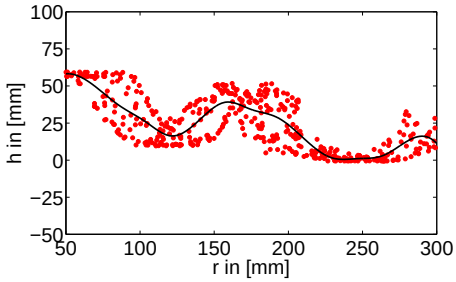
relatively small. For $\alpha = 50^\circ$ an abrupt change is observed. A strong sensitivity of the model fit algorithm to the error in the initial pose is observed in the final radial error. The convergence of the model fit algorithm is very unlilkely with an initial axial error equal or greater than 50°. However, a good initial model has been found for almost all measurements (95%) with an initial axial error of up to $\alpha = 30^\circ$, if the initial radial error did not exceed $\Delta r = 30$ mm.

For an error free point sampling of the given surface, the algorithm is able to correct an initial axial error of up to $\alpha = 30^\circ$ with simultaneous radial error of up to $\Delta r = 30$ mm. The likelihood to exceed these limits for a point cloud subject to noise and outliers is shown in Figure 5.8. The likelihood for a measurement free of outliers to obtain an initial pose error exceeding the limits is about 0.15. Assuming the noise to play a minor role in the model fit, the likelihood of the proposed algorithm to converge for measurements without outliers is about 0.85.

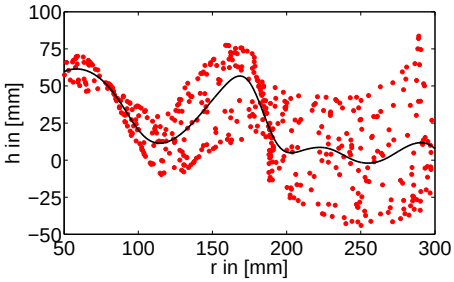


(a) $\Delta r = 0$ mm, $\alpha = 0^\circ$

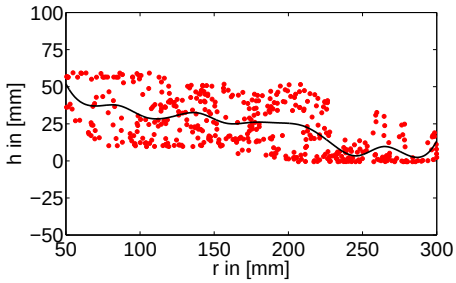
(b) $\Delta r = 0$ mm, $\alpha = 5^\circ$



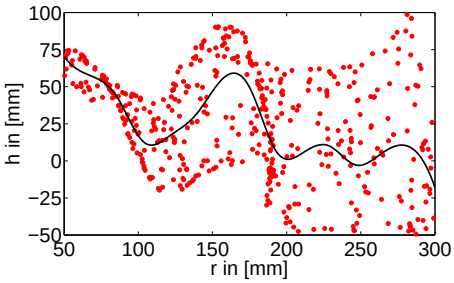
(c) $\Delta r = 20$ mm, $\alpha = 0^\circ$



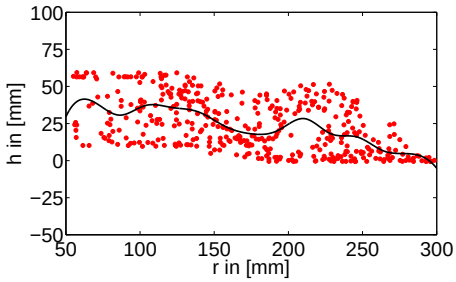
(d) $\Delta r = 0$ mm, $\alpha = 10^\circ$



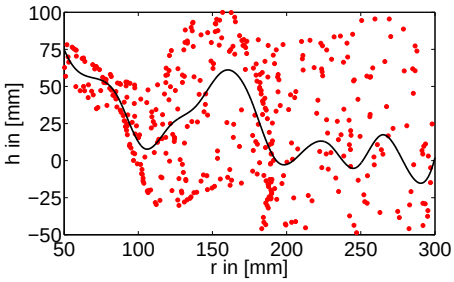
(e) $\Delta r = 40$ mm, $\alpha = 0^\circ$



(f) $\Delta r = 0$ mm, $\alpha = 15^\circ$

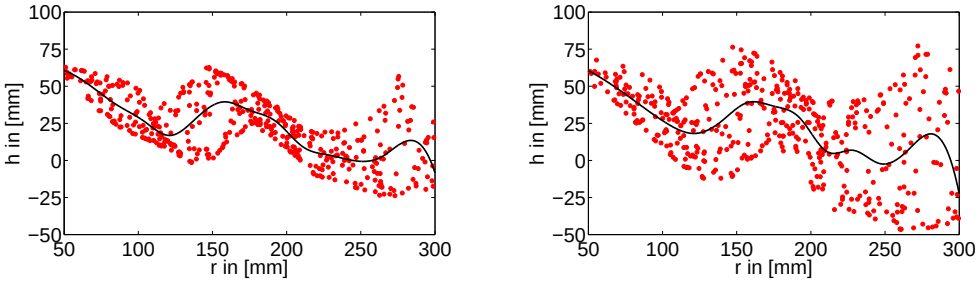


(g) $\Delta r = 60$ mm, $\alpha = 0^\circ$



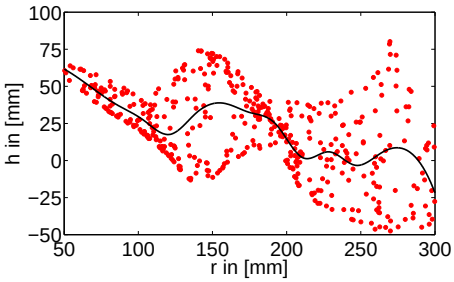
(h) $\Delta r = 0$ mm, $\alpha = 20^\circ$

Figure 5.9: Effect of radial and axial errors in the initial pose parameters. Points in surface coordinates and 3rd degree initial B-spline curve with 15 segments and uniform clamped knot vector.

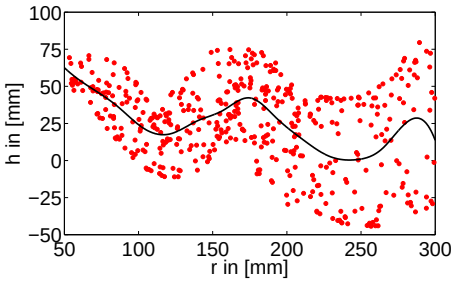


(a) $\Delta r = 20$ mm, $\alpha = 5^\circ$

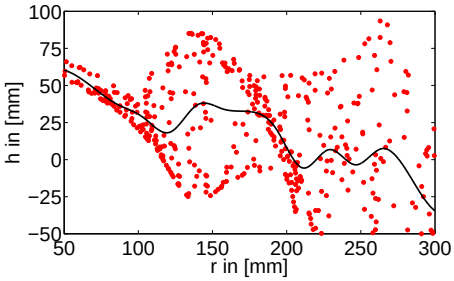
(b) $\Delta r = 20$ mm, $\alpha = 10^\circ$, $\beta = 45^\circ$



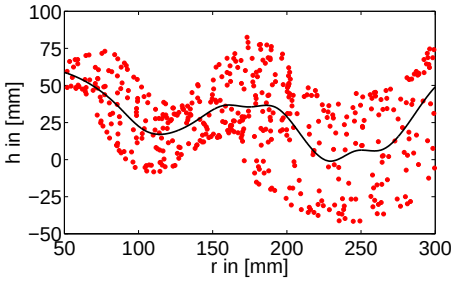
(c) $\Delta r = 20$ mm, $\alpha = 10^\circ$



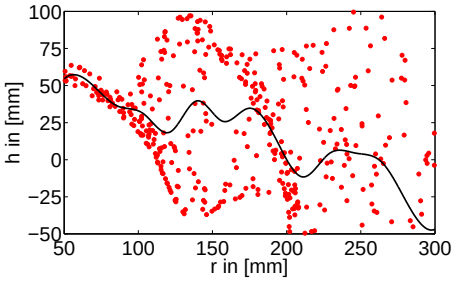
(d) $\Delta r = 20$ mm, $\alpha = 10^\circ$, $\beta = 90^\circ$



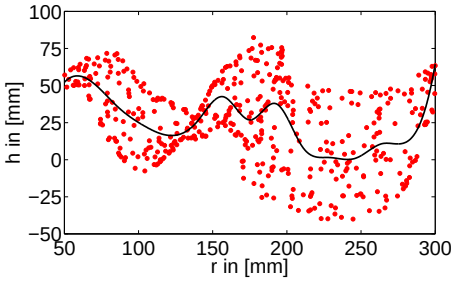
(e) $\Delta r = 20$ mm, $\alpha = 15^\circ$



(f) $\Delta r = 20$ mm, $\alpha = 10^\circ$, $\beta = 135^\circ$



(g) $\Delta r = 20$ mm, $\alpha = 20^\circ$



(h) $\Delta r = 20$ mm, $\alpha = 10^\circ$, $\beta = 180^\circ$

Figure 5.10: Effect of combined radial and axial errors in the initial pose parameters. Points in surface coordinates and 3rd degree initial B-spline curve with 15 segments and uniform clamped knot vector.

Algorithm 7: Global model fit

Data: sequence of point clouds $\{\mathbf{x}_{i,k}\}_{k=1}^{m_i}$ and initial models $\mathcal{M}_{i,0}$, $i = 1, ..., N$
Result: global surface $\mathcal{S}$ and corrected pose $\mathcal{P}_i$
begin
  `/* initial global surface model */`
  projection of all point clouds into common surface coordinate system
  compute precise generatrix $\mathcal{S}_0$ using kernel smoothing
  orthogonal projection of points on $\mathcal{S}_0$
  compute curve parametrization from $\mathcal{S}_0$ and orthogonal projection
  `/* surface computation and trajectory recovery */`
  while *model parameters change* **do**
    `/* global surface model */`
    compute $\mathcal{S}$ as weighted two-dimensional P-spline
    compute residuals
    compute weights using M-estimator
    `/* trajectory recovery */`
    compute all $\mathcal{P}_i$ by regularized weighted non-linear least squares
    compute residuals
    compute weights using M-estimator
  $\mathcal{M}_i = (\mathcal{S}, \mathcal{P}_i)$
end

5.2 Global Surface and Trajectory Recovery

In the previous section it has been shown that the degree of detail of the model extracted from single measurements is limited due to the sparsely sampled data points, the object's shape characteristics and the errors in the data points. In order to maximize the chance for the algorithm to converge for all measurements, the optimal number of intervals of the B-spline model is 15. However, the desired degree of detail of a precise surface model can not be established.

The aim of the algorithm presented in this section is to establish a framework for the extraction of a precise global surface model and a feasible motion trajectory incorporating the information available in the whole sequence. An outline of the proposed method is given in Algorithm 7. The input to the algorithm are the point clouds $\{\mathbf{x}_{i,k}\}_{k=1}^{m_i}$ and the initial models $\mathcal{M}_i = (\mathcal{S}_{i,0}, \mathcal{P}_{i,0})$, $i = 1, ..., N$ computed for each measurement using the methodology presented in Section 5.1. The output is a unique precise surface model $\mathcal{S}$ and a smooth trajectory represented by the discrete sampling of the sequence $\{\mathcal{P}_i\}_{i=1}^{N}$.

The algorithm runs through two major stages. In the first stage an initial global generatrix is extracted by means of kernel smoothing and a feasible parametrization of the data with respect to this initial shape is identified. The second stage is an iteration which alternates between generatrix computation and trajectory recovery. The iteration of the second stage simultaneously establishes an iteratively reweighted least squares minimization. The weights are updated after each iteration. The iteration finishes when the model parameters do not change anymore. The details of the first stage and the generatrix computation are treated in Section 5.2.1 and the pose correction part of the iteration is covered in Section 5.2.2.

5.2.1 Global Surface Model

Preparing for the estimation of a global model, in each step of the iteration the data points $\{\mathbf{x}_{i,k}\}_{k=1}^{m_i}$ of each individual measurement i are transformed into surface coordinates according to (5.1) and (5.2) using the initial model $\mathcal{M}_i$. The high number of data points allows for the extraction of a precise generatrix though some initialization issues have to be considered.

The error in the pose that has been estimated for each measurement individually is rather high and as shown in Section 5.1.3, a tilt and shift of the pose results in an increased variance in the distribution of the data points in model coordinates. Thus, the estimator of a global generatrix has to deal with (1) noise due to errors in feature extraction, (2) variance due to pose errors of the initial models, and (3) outliers due to erroneous point correspondences. All these errors are of unknown scale and degree.

Given the data points in surface coordinates an initial one-dimensional shape $\mathcal{S}_0$ is extracted using a kernel smoothing approach as introduced in Section 2.7.1. A smooth one-dimensional curve in h is extracted at a pre-defined number of sampling points on the r-axis using local linear regression according to (2.123). In order to deal with outliers, the regression is performed iteratively with an M-estimator. Here, the number of outliers is assumed to be small enough to allow for application of robust linear regression without a preceding outlier rejection method like LMedS. A discussion of these robust estimation approaches has been given in Section 2.6.2. The optimal smoothing parameter is obtained by repetition of the kernel smoothing with multiple values for λ and analysis of the residuals using generalized cross validation.

Let $\mathbf{x}_k = (r_k, h_k)$ be a data point in surface coordinates and let $\mathbf{s} = (r_k, \mathcal{S}_0(r_k))$ be a point on the initial global generatrix. The *foot-point* of $\mathbf{x}_k$ on $\mathcal{S}_0$ is that point $\mathbf{s}$ such that $\mathbf{x}_k \cdot \mathbf{s} = 0$ and $\|\mathbf{s} - \mathbf{x}_k\|$ is minimal. The foot-points are approximated by points of a discrete sampling of the generatrix. From this sampling also the length of $\mathcal{S}_0$ is estimated. The length is used to compute a parametrization t_k for each data point using its foot-point $\mathbf{s}$.

Given the parametrization a parametric curve can be fitted to the data points. Because of its numerical robustness and its intrinsic smoothing properties a two-dimensional P-spline curve is used. The model is fitted using an iterative approach alternating between coefficient computation and re-parametrization. The point distance error term (2.97) is used in the fit and the Hoschek correction term (2.105) is used for re-parametrization.

5.2.2 Motion Model

During the acquisition of the sequence of measurements the object is allowed to move along a trajectory which can be represented analytically and thus introduce further constraints to the pose parameters in the pose correction stage of Algorithm 7. The simplest model one could think of is that of zero motion which constrains all measurements to a unique set of pose parameters $\mathcal{P}$. In case the trajectory describes a geometric object the parameters of that object are estimated and the pose of all measurements is restricted to the subspace of the geometric object. If no parametric motion model can be assumed, but it is known that the motion is smooth, a constraint using a penalty on the acceleration may be used. With $\mathcal{P}(t)$ is the pose of the object at a point in time t, the penalty on the acceleration is defined as

$$F_{\mathcal{P}} = \int \left|\mathcal{P}''(t)\right|^2 dt. \tag{5.6}$$

The minimization in the pose correction stage uses the error of fit measure

$$F = \sum_i F_{\mathcal{M}_i} + \lambda_{\mathcal{P}} \cdot F_{\mathcal{P}} \tag{5.7}$$

which establishes the penalty (5.6) by means of regularization. The parameter $\lambda_{\mathcal{P}}$ determines the degree of regularization. The error term $F_{\mathcal{M}_i}$ is the distance between the data points of measurement i in surface coordinates and the surface generatrix $\mathcal{S}$. The distance is measured between the data points and its foot-point with respect to the current data parametrization.

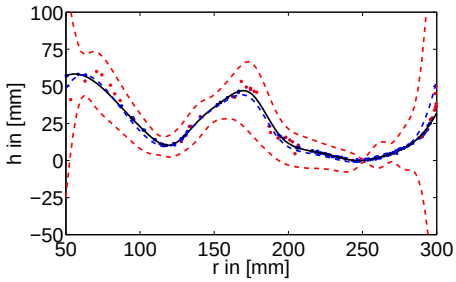
5.3 Experimental Results

The performance of the algorithms presented in Section 5.1 and Section 5.2 are evaluated using real measurement data. The application motivated in Section 1.1 has been simulated by means of a wheel installed on top of a mobile skid. The experimental setup has been arranged such that the motion direction of the skid is approximately parallel to the basis of the stereo camera and the principal axes of cameras are approximately parallel to the wheel's axis of symmetry. A total of five sequences with each 50 measurements have been acquired while the skid was moving through the field of view of the stereo camera. In each sequence the skid enters the field of view with the first measurement and leaves in the last measurement at the opposite side. The uncertainties of the models extracted from single measurements and those of the global models are analyzed by various error measures and the improvement achieved by the global approach is shown.

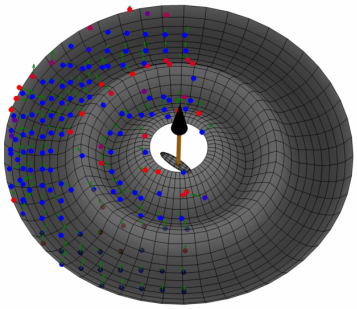
5.3.1 Model Initialization

For each measurement of a sequence the pose and a coarse surface model have been extracted according to Algorithm 6. The model uncertainty is estimated by a non-parametric bootstrap with 100 samples. The error bands of the generatrix and the error ellipsoids of the model center are shown in 5.11 for three measurements of Sequence #3 with different degree of surface coverage. The center of the error ellipsoid does not correspond with the position of the model extracted from the best bootstrap sample. The uncertainty at $r = 250$ mm is zero, because the ordinate of the generatrix is forced to zero at this point. In measurement 1, only half of the surface is covered by point samples. A well initial model has been found though both, the model center and the surface are subject to a high uncertainty. In agreement with the results of Section 5.1 the model errors decrease with higher surface coverage. The strongest variation of the generatrix is observed in the borders due to missing points and outliers.

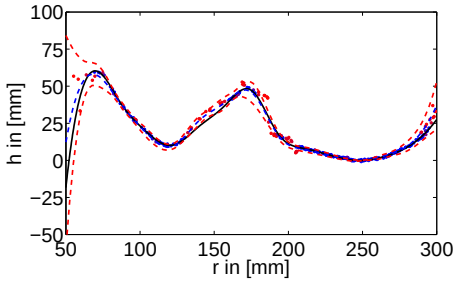
The results of the initial model fit procedure are shown in Figure 5.12. The x-axis is the number of the measurement in the sequence and the graphs illustrate the errors of the initial models (red), the fitted models (black), and the best bootstrap sample (blue) for each measurement. The error measure is the robust standard deviation estimate of the residuals according to (2.115). The two graphs are the results independently computed for the point clouds gained by both variants of the laser spot detection and their respective point correspondence algorithms. The left and the right plot show the results for laser spot detection in the image domain and in the calibration segment domain (see Section 4.1), respectively. The number of point correspondences of each measurement is also visualized in the same plot (green).



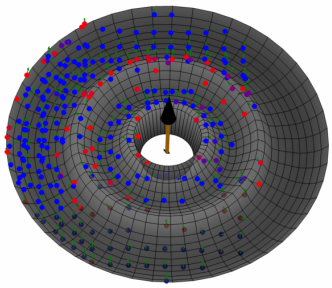
(a) Sequence #3, measurement 1



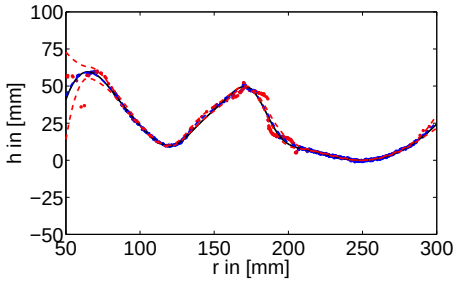
(b) Sequence #3, measurement 1



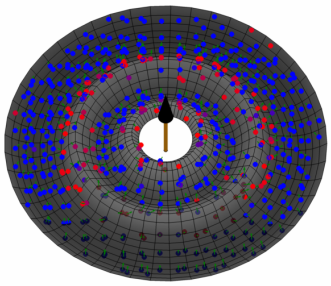
(c) Sequence #3, measurement 13



(d) Sequence #3, measurement 13

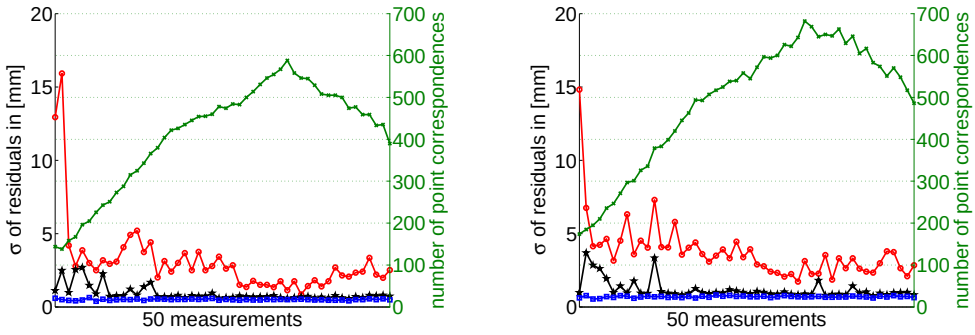


(e) Sequence #3, measurement 27



(f) Sequence #3, measurement 27

Figure 5.11: Model uncertainty of measurements in sequence #3 estimated from 100 bootstrap samples. (a),(c),(e) Data points in surface coordinates with best model (black), mean (blue), and 95% standard error bands (red) of generatrix. (b),(d),(f) Data points in 3-D space with estimated normals and 95% error ellipsoid of model center.



(a) Image domain (b) Calibration segment domain

Figure 5.12: Number of data points (green), standard deviation of residuals in initial (red), fitted models (black), and best bootstrap samples (blue).

5.3.2 Global Surface and Trajectory Recovery

With the results of the model initialization as input the reconstruction of a global surface model and a smooth trajectory is carried out according to Algorithm 7. The data points of all measurements in a sequence are transformed into common surface coordinates and a initial global generatrix is computed by kernel smoothing using local linear regression. The number of data points in a single measurement varies between 200 and 600 points. The overall number of data points of the 50 measurements is about $20,000$. A discrete number of smooth points is computed and the initial model is the B-spline curve interpolating these points. The results of the local linear regression with and without M-estimator are shown for three values of the smoothing parameter in Figure 5.16 for Sequence #1. The optimal smoothing parameter has been determined by generalized cross validation. The normalized curves of the cross validation results are shown in Figure 5.13 and the values of the curve at a selection of sample points are given in Table 5.1.

The smooth data points have been used to extract an initial parametrization of the input data points. The chord length parametrization (2.85) has been used where the curve length has been approximated by the sum of distances between consecutive smooth data points. For this reason a slightly too high smoothing parameter is preferred in order to suppress all uncertain variations.

The parametrization is input to the alternating stage of P-spline regression and pose correction. A P-spline model is used to enable for the use of a high number of intervals and simultaneously avoid singular matrices and overfitting. The number of 250 intervals has been chosen in order to provide one interval in 1 mm radial extension of the object. The size of 1 mm per interval corresponds with the accuracy of the mea-

λ	0.001	0.005	0.010	0.050
standard regression	4.022	3.932	3.978	6.431
robust regression	4.920	4.84	4.85	6.712

Table 5.1: Generalized cross validation for kernel smoothing results

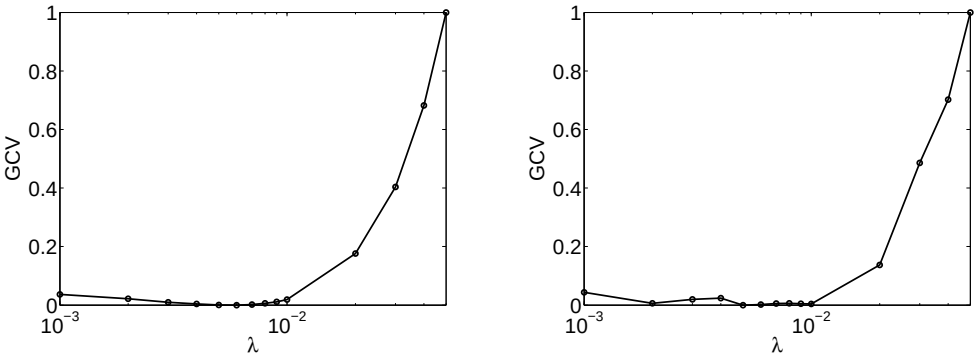
surement device. The internal parameters of the P-spline model are listed in Table 5.2. The final surface model and the smooth trajectory are computed in five iterations of the loop in Algorithm 7. In each iteration an inner loop alternates between P-spline regression and data parametrization. The number of iterations in the inner loop is limited to 15.

The results of both, the single point cloud fit and the global approach for the measurements of Sequence #1 are shown in Figure 5.17. The plots in the left column show the results of the laser spot detection in the image domain and right column shows the results for the detection of laser spots in the calibration segment domain. The plots in the upper row show the results of the model fit of Algorithm 6. The middle row and the bottom show the results of the global fit in surface coordinates and model coordinates, respectively. The colors of the data points encode their final weight used by the weighted least squares algorithm in model fit. The red colored points have a weight near zero and blue colored points have a weight close to one. The laser spot detection in the calibration segment domain leads to a considerable amount of outliers but does not achieve a better sampling of the surface.

The projection of the normalized trajectory of Sequence #1 to the coordinate planes is shown in Figure 5.18. The first row shows the trajectory of the single point cloud fit and the middle and bottom row show the global fit result for two different regularization parameters. The smoothness of the trajectory increases with a higher regularization parameter. The optimal value of the regularization parameter has been estimated using 10-fold cross validation. The result is shown in Figure 5.14. The error decreases until saturation for values around $\lambda_{\mathcal{P}} = 100$. Similar results have been found for the other sequences.

The global surface model (solid) and the 95% standard error bands (dashed) are shown in Figure 5.15. The detail views illustrate the increased surface diameter in some areas of the generatrix. The uncertainties of the global model have been estimated from the results of the 10-fold cross validation carried out for the estimation of the regularization parameter.

A comparison of the uncertainties of the model initialization and the global fit are given in Table 5.3. The pose uncertainties are the lengths of the error ellipsoid's principal axes and the surface uncertainty is the diameter of the surface (the distance



(a) using linear regression (b) robust linear regression

Figure 5.13: Generalized cross validation (normalized) of the initial smoothing for different values of the smoothing parameter λ.

Parameter	Value
dimension	2 (r, h)
degree	3
order of finite differences	2
intervals/control points	250
endpoints	fixed (smoothing result)
knot vector	clamped and equally spaced
parametrization	chord length
error of fit	PDM

Table 5.2: Internal parameters of the global P-spline model.

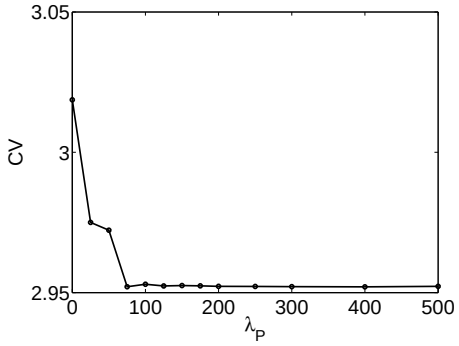


Figure 5.14: Result of the 10-fold cross validation for Sequence #1 with a selection of values for the regularization parameter $\lambda_{\mathcal{P}}$.

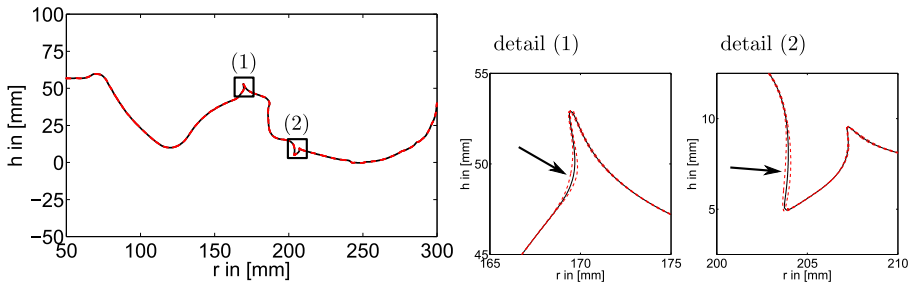
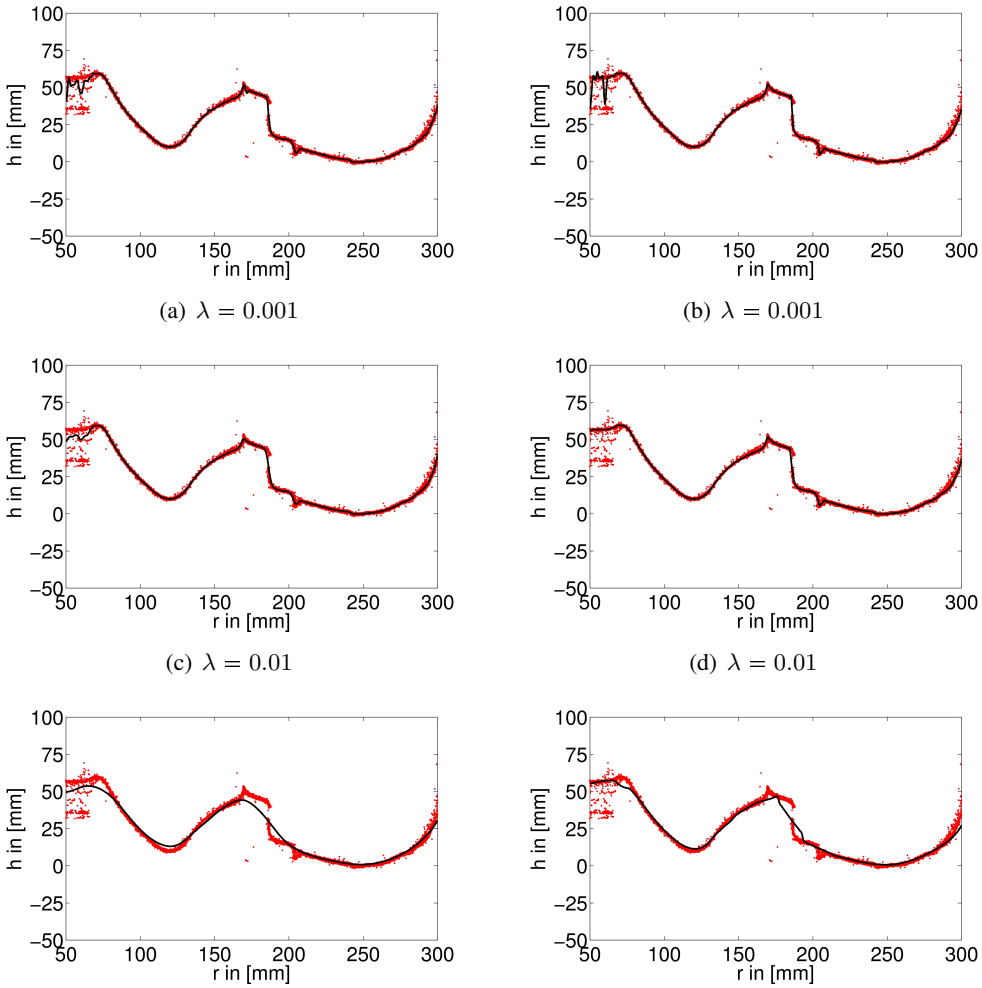


Figure 5.15: Uncertainty of surface in global model.

between upper and lower 95% standard error band). The surface diameter of the generatrix has been estimated from a discrete sampling of the 10 curves computed in the cross validation. For each of the five sequences a worst case and a best case example with respect to the uncertainty of the model initialization process is chosen and the improvement achieved by the global model is shown. The first measurement of all sequences is the measurement with highest degree of uncertainty. The reason is the surface coverage of about 50% at begin of each sequence. The best measurement is one of those where the object is located near the center of the field of view and a maximum number of points is available to the fit. All uncertainties of the globally estimated model are significantly below the values of the models reconstructed from single measurements. The position uncertainty of all measurements in the globally estimated trajectory is less than 0.1 mm. The heavy variations of the initial surface in the boundaries is eliminated by use of fixed endpoints. The generatrix of the global surface model consists of 250 intervals whereas the models are limited to 15 intervals. This fact and the dense sampling of the generatrix yield a maximal surface diameter in the global model of about 0.5 mm. The standard deviation of the surface diameter over the whole curve is only 0.05 mm.

Sequence	#1	#2	#3	#4	#5
best measurement	27	30	15	32	27
worst measurement	1	1	1	1	1
pose uncertainties (max. radius of the error ellipsoid)					
single fit (best measurement)					
σ_{max}	0.20	0.20	0.23	0.30	0.26
global fit					
σ_{max}	0.01	0.01	0.01	0.01	0.01
single fit (worst measurement)					
σ_{max}	22.42	23.68	10.46	36.86	31.93
global fit					
σ_{max}	0.05	0.07	0.04	0.08	0.09
surface uncertainties (surface diameter)					
single fit (best measurement)					
σ	3.83	2.66	4.42	4.34	3.94
median	0.78	0.93	0.95	0.74	1.38
max	28.08	20.01	33.28	31.60	29.30
single fit (worst measurement)					
σ	10^6	170.88	23.49	10^4	$2 \cdot 10^5$
median	20.92	19.11	11.73	37.44	35.28
max	$7 \cdot 10^6$	10^4	171.98	$6 \cdot 10^5$	10^6
global fit					
σ	0.05	0.07	0.04	0.05	0.03
median	0.05	0.05	0.05	0.05	0.05
max	0.47	0.61	0.41	0.50	0.28

Table 5.3: Uncertainties in initial fit and global fit estimated by bootstrapping (initialization) and 10-fold cross validation (global fit). The unit of all error values is [mm].

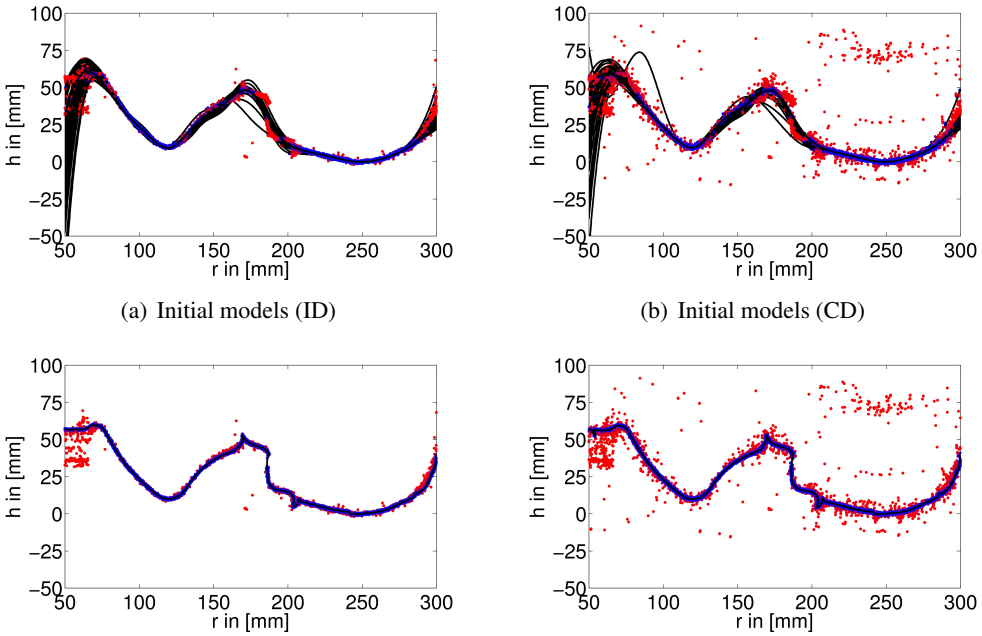


(a) $\lambda = 0.001$ (b) $\lambda = 0.001$

(c) $\lambda = 0.01$ (d) $\lambda = 0.01$

(e) $\lambda = 0.1$ (f) $\lambda = 0.1$

Figure 5.16: Epanechnikov kernel smoothing with local linear regression (left column) and robust linear regression using the Huber M-estimator (right column) for different values of the smoothing parameter λ.

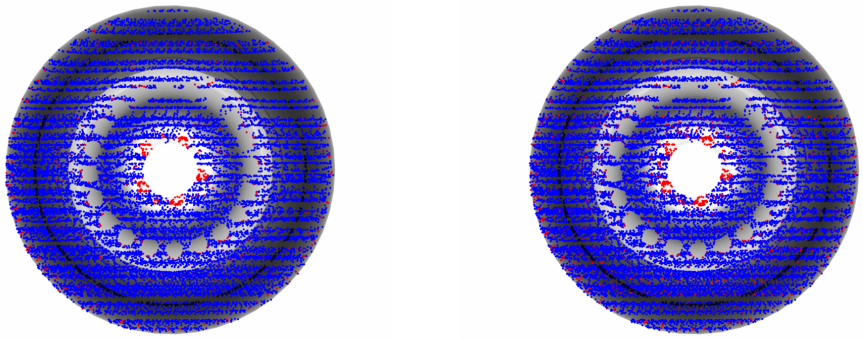


(a) Initial models (ID)

(b) Initial models (CD)

(c) Global model (ID)

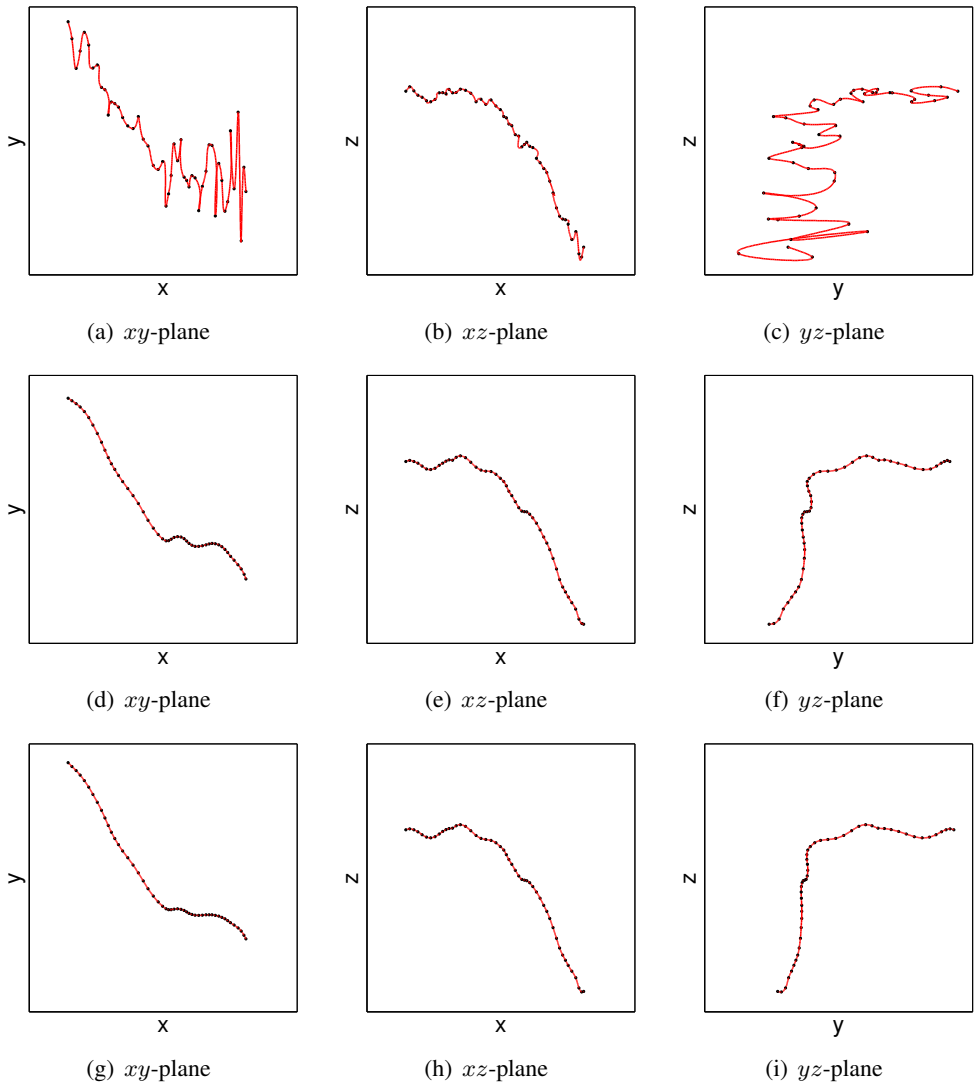
(d) Global model (CD)



(e) Surface coverage of sequence (ID)

(f) Surface coverage of sequence (CD)

Figure 5.17: Point cloud of sequence in surface coordinates for laser spot detection in image domain (left column) and calibration segment domain (right column). The color encodes the weight of a data point (0 = red, 1 = blue).



(a) xy-plane (b) xz-plane (c) yz-plane

(d) xy-plane (e) xz-plane (f) yz-plane

(g) xy-plane (h) xz-plane (i) yz-plane

Figure 5.18: Projection of the normalized trajectory to the coordinate planes. Upper row: result of fitting each measurement individually. Center row: Global model fit with trajectory smoothing penalty $\lambda_{\mathcal{P}} = 25$. Bottom row: Global model fit with trajectory smoothing penalty $\lambda_{\mathcal{P}} = 100$.

6 Summary

The goal of this thesis was the reconstruction of a moving surface of revolution from a sequence of three-dimensional measurements. The practical importance of such objects have motivated earlier work for the reconstruction of surfaces of revolution from single point clouds. The existing work has been extended by combining surface reconstruction and trajectory recovery for a sequence of point clouds acquired from a moving object.

The 3-D data is acquired by means of active triangulation using a stereo camera and a laser projector. The measurement device provides sparse point clouds of the object which is located in the field of view at acquisition time. The shape of the surface is represented by a B-spline curve and the optimal number of parameters for the initial model reconstructed from single point clouds has been determined by means of a computer simulation. As the variation of the B-spline curve with that number of parameters is insufficient to reflect the true shape of the observed object, a dense sampling of the surface which enables for the extraction of a precise model is desired.

The major contribution of the thesis is the reconstruction of a precise surface model and the simultaneous recovery of the motion trajectory of a surface of revolution which is moving through the field of view of the measurement device. The high uncertainty of the model parameters in the single measurements due to the sparse point sampling is overcome in the global approach. By transformation of all point clouds into a common coordinate system a dense surface sampling is established. The parameters of the global model are recovered iteratively by alternation between fit of a surface model and trajectory estimation.

A contribution to active triangulation is the novel calibration method for a laser projector and the use of trifocal epipolar constraints for resolving ambiguous point correspondences. The measurement device consists of a stereo camera and multiple laser projectors. The beam geometry of each projector is reconstructed by an elaborate calibration procedure. The image of the beam geometry in the stereo cameras is utilized by the point correspondence algorithm to resolve ambiguities. The number of outliers in the three-dimensional data due to erroneous correspondences is limited by employing the trifocal epipolar constraints defined through the stereo camera and the image of the laser beam geometry.

List of Figures

List of Tables

Index

Bibliography

Abraham, S. (1999). *Kamera-Kalibrierung und metrische Auswertung monokularer Bildfolgen*. Dissertation, Friedrich-Wilhelms-Univeristät, Bonn.

Abraham, S. & Hau, T. (1997). Towards autonomous high-precision calibration of digital cameras. In El-Hakim, S. F. (Ed.), *Proceedings of SPIE (Videometrics V)*, volume 3174 of *SPIE*, (pp. 82–93)., San Diego.

Abramowitz, M. & Stegun, I. A. (1965). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Table*. Dover Publications, Inc.

Ahn, S. J. (2004). *Least squares orthogonal distance fitting of curves and surfaces in space*, volume 3151 of *Lecture notes in computer science*. Springer.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723.

Aoki, H., Ichimura, S., Kiyooka, S., & Koshiji, K. (2007). Non-contact measurement method of respiratory movement under pedal stroke motion. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2007)*, (pp. 374–377).

Arbter, K., Snyder, W. E., Burkhardt, H., & Hirzinger, G. (1990). Application of affine-invariant fourier descriptors to recognitionof 3-d objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(7), 640–647.

Bajcsy, R. & Solina, F. (1987). Three dimensional object representation revisited. In *Proceedings of the First IEEE International Conference on Computer Vision (ICCV '87)*, (pp. 231–240)., London, UK.

Barber, C. B., Dobkin, D. P., & Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, *22*(4), 469–483.

Barr, A. H. (1981). Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, *1*(1), 11–23.

Besl, P. J. (1990). The free-form surface matching problem. In H. Freeman (Ed.), *Machine Vision for Three-Dimensional Scenes* (pp. 25–71). Academic Press.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning* (1 ed.). Springer.

Blake, A. & Isard, M. (2000). *Active Contours*. Springer.

Bolle, R. & Vemuri, B. (1991). On three-dimensional surface reconstruction methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(1), 1–13.

Bronstein, I. N., Semendjajew, K. A., Musiol, G., & Mühlig, H. (1999). *Taschenbuch der Mathematik* (4 ed.). Verlag Harri Deutsch.

Brown, C. M. (1981). Some mathematical and representational aspects of solid modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *3*(4), 444–453.

Büker, U., Drüe, S., Götze, N., Hartmann, G., Kalkreuter, B., Stemmer, R., & Trapp, R. (2001). Vision-based control of an autonomous disassembly station. *Robotics and Autonomous Systems*, *35*(3–4), 179–189.

Campbell, R. J. & Flynn, P. J. (2001). A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, *81*(2), 166–210.

Carcassoni, M. & Hancock, E. R. (2003). Spectral correspondence for point pattern matching. *Pattern Recognition*, *36*(1), 193–204.

Chang, S.-H., Cheng, F.-H., Hsu, W.-H., & Wu, G.-Z. (1997). Fast algorithm for point pattern matching: Invariant to translations, rotations and scale changes. *Pattern recognition*, *30*(2), 311–320.

Chen, F., Brown, G. M., & Song, M. (2000). Overview of three-dimensional shape measurement using optical methods. *Opt. Eng.*, *39*(1), 10–22.

Christmas, W. J., Kittler, J., & Petrou, M. (1995). Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(8), 749–764.

Chui, H. & Rangarajan, A. (2000). A new algorithm for non-rigid point matching. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '00)*, volume 2, (pp. 44–51).

Clabian, M., Rötzer, H., & Bischof, H. (2001). Tracking structured light pattern. In Casasent, D. P. & Hall, E. L. (Eds.), *Proceedings of SPIE (Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision)*, volume 4572.

Conte, D., Foggia, P., Sansone, C., & Vento, M. (2007). How and why pattern recognition and computer vision applications use graphs. In *Applied Graph Theory in Computer Vision and Pattern Recognition*, volume 52 of *Studies in Computational Intelligence* (pp. 85–135). Springer.

Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(6), 681–685. Active Appearance Models.

Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. (2004). A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(10), 1367–1372.

Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1990). *Introduction to Algorithms*. The MIT Press.

de Berg, M., van Kreveld, M., Overmars, M., & Schwarzkopf, O. (2000). *Computational Geometry: Algorithms and Applications* (2 ed.). Springer-Verlag, Heidelberg.

de Boor, C. (1972). On calculating with B-splines. *Journal of Approximation Theory*, *6*(1), 50–62.

de Boor, C. (2001). *A Practical Guide to Splines*. New York: Springer.

Dickinson, S. J., Bergevin, R., Biederman, I., Eklundh, J.-O., Munck-Fairwood, R., Jainf, A. K., & Pentland, A. (1997). Panel report: the potential of geons for generic 3-d object recognition. *Image and Vision Computing*, *15*(4), 277–292.

Dipanda, A. & Woo, S. (2005). Efficient correspondence problem-solving in 3-D shape reconstruction using a structured light system. *Optical Engineering*, *44*(9), 093602.

Dipanda, A., Woo, S., Marzani, F., & Bilbault, J. M. (2003). 3d shape reconstruction in an active stereo vision system using genetic algorithms. *Pattern Recognition*, *36*(9), 2143–2159.

Eilers, P. H. C. & Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, *11*(2), 89–121.

Elsässer, B. & Hoschek, J. (1996). Approximation of digitized points by surfaces of revolution. *Computers & Graphics*, *20*(1), 85–94.

Farin, G. E. (2001). *Curves and Surfaces for CAGD. A Practical Guide*. Morgan Kaufmann.

Faugeras, O. (2001). *Three-dimensional Computer Vision* (4 ed.). MIT Press.

Fielding, G. & Kam, M. (1997). Applying the Hungarian method to stereo matching. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 2, (pp. 1928–1933)., San Diego, CA, USA.

Fischler, M. A. & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applicatlons to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395.

Foley, T. A. & Nielson, G. M. (1989). Knot selection for parametric spline interpolation. In T. Lyche & L. L. Schumaker (Eds.), *Mathematical methods in computer aided geometric design* (pp. 261–272). Boston: Academic Press.

Folkers, A. & Samet, H. (2002). Content-based image retrieval using fourier descriptors on a logo database. In *Proc of the 16th Int. Conf. on Pattern Recognition (ICPR '02)*, volume 3, (pp. 521–524)., Quebec City, Canada.

Fusiello, A., Trucco, E., & Verri, A. (2000). A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, *12*(1), 16–22.

Golub, G. H. & Loan, C. F. V. (1996). *Matrix Computations* (3 ed.). Baltimore: Johns Hopkins.

Goshtasby, A. & Stockman, G. (1985). Point pattern matching using convex hull edges. *IEEE Transactions on Systems, Man and Cybernetics*, *15*(5), 631–637.

Goshtasby, A. A. (2000). Grouping and parameterizing irregularly spaced points for curve fitting. *ACM Transactions on Graphics*, *19*(3), 185–203.

Granlund, G. H. (1972). Fourier preprocessing for hand print character recognition. *IEEE Trans. Computers*, *21*, 195–201.

Griffin, P. M. & Alexopoulos, C. (1989). Point pattern matching using centroid bounding. *IEEE Transactions on Systems, Man and Cybernetics*, *19*(5), 1274–1276.

Gross, A. D. & Boult, T. E. (1988). Error of fit measures for recovering parametric solids. In *Proceedings of the 2nd International Conference on Computer Vision (ICCV'88)*, (pp. 690–694).

Gupta, A. & Bajcsy, R. (1993). Volumetric segmentation of range images of 3D objects using superquadric models. *CVGIP: Image Understanding*, *58*(3), 302–326.

Gupta, A., Bogoni, L., & Bajcsy, R. (1989). Quantitative and qualitative measures for the evaluation of the superquadric models. In *Proceedings of the Workshop on Interpretation of 3D Scenes*, (pp. 162–169).

Hartley, R. & Zisserman, A. (2004). *Multiple view geometry in computer vision* (2 ed.). Cambridge University Press.

Hastie, T. & Tibshirani, R. (1990). *Generalized Additive Models*. Chapman & Hall.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). *The Elements of Statistical Learning*. Springer.

Heikkilä, J. & Silven, O. (1997). A four-step camera calibration procedure with implicit image correction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*, (pp. 1106–1112)., San Juan.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., & Stuetzle, W. (1992). Surface reconstruction from unorganized points. *Computer Graphics*, *26*(2), 71–78.

Hoschek, J. (1988). Intrinsic parametrization for approximation. *Computer Aided Geometric Design*, *5*(1), 27–31.

Hoschek, J. (1992). Circular splines. *Computer-Aided Design*, *24*(11), 611–618.

Hoschek, J., Lasser, D., & Schumaker, L. L. (1993). *Fundamentals of Computer Aided Geometric Design*. A. K. Peters.

Hoschek, J. & Schneider, F.-J. (1990). Spline conversion for trimmed rational Bézier- and B-spline surfaces. *Computer-Aided Design*, *22*(9), 580–590.

Huber, P. J. (1981). *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley.

Inokuchi, S., Sato, K., & Matsuda, F. (1984). Range imaging system for 3-d object recognition. In *Proceedings of the International Conference on Pattern Recognition (ICPR '84)*, (pp. 806–808).

Jähne, B. (2002). *Digitale Bildverarbeitung* (5 ed.). Berlin: Springer Verlag.

Jaklic, A., Leonardis, A., & Solina, F. (2000). *Segmentation and Recovery of Superquadrics*, volume 20 of *Computational imaging and vision*. Kluwer.

Kehl, R. & Gool, L. V. (2006). Markerless tracking of complex human motions from multiple views. *Computer Vision and Image Understanding*, *104*(2–3), 190–209.

Kittler, J. & Hancock, E. (1989). Combining evidence in probabilistic relaxation. *International Journal of Pattern Recognition and Artificial Intelligence*, *3*(1), 29–51.

Klar, T. A., Engel, E., & Hell, S. W. (2001). Breaking Abbe's diffraction resolution limit in fluorescence microscopy with stimulated emission depletion beams of various shapes. *Physical Review E*, *64*(6), 066613.

Kowarschik, R., Kühmstedt, P., Gerber, J., Schreiber, W., & Notni, G. (2000). Adaptive optical three-dimensional measurement with structured light. *Optical Engineering*, *39*(1), 150–158.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, *2*, 83–97.

Lee, E. T. Y. (1989). Choosing nodes in parametric curve interpolation. *Computer Aided Design*, *21*(6), 363–370.

Lee, I.-K. (2000). Curve reconstruction from unorganized points. *Computer Aided Geometric Design*, *17*(2), 161–177.

Lequellec, J.-M. & Lerasle, F. (2000). Car cockpit 3D reconstruction by a structured light sensor. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, (pp. 87–92). Siemens.

Levin, D. (1998). The approximation power of moving least-squares. *Mathematics of Computation*, *67*(224), 1517–1531.

Li, B., Meng, Q., & Holstein, H. (2003). Point pattern matching and applications - a review. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 1, (pp. 729–736).

Li, W., Xu, S., Zhao, G., & Goh, L. P. (2005). Adaptive knot placement in B-spline curve approximation. *Computer-Aided Design*, *37*(8), 791–797.

Lin, C.-S. & Jungthirapanich, C. (1990). Invariants of three-dimensional contours. *Pattern Recognition*, *23*(8), 833–842.

Lindeberg, T. (1994). Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, *21*(2), 225–270.

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, *30*(2), 1573–1405.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Lubeley, D. (2007). Unambiguous dynamic diffraction patterns for 3D depth profile measurement. In Hamprecht, F. A., Schnörr, C., & Jähne, B. (Eds.), *Proceedings of the 29th DAGM Symposium on Pattern Recognition*, number 4713 in Lecture Notes in Computer Science, (pp. 42–51).

Luhmann, T. (2003). *Nahbereichsphotogrammetrie* (2 ed.). Wichmann Verlag.

Malz, R. W. (1999). Three-dimensional sensors for high-performance surface measurement in reverse engineering. In B. Jähne, H. Haußecker, & P. Geißler (Eds.), *Handbook of Computer Vision and Applications*, volume 1 chapter 20, (pp. 507–539). San Diego: Academic Press.

Marr, D. & Hildreth, E. (1980). Theory of edge detection. *Proc. of the Royal Society of London B*, *207*, 187–217.

Marzani, F. S., Voisin, Y., Voon, L. F. C. L. Y., & Diou, A. (2002). Calibration of a three-dimensional reconstruction system using a structured light source. *Optical Engineering*, *41*(2), 484–492.

Mitra, N. J., Nguyen, A., & Guibas, L. (2004). Estimating surface normals in noisy point cloud data. *International Journal of Computational Geometry and Applications*, *14*(4–5), 261–276.

Nister, D. (2001). *Automatic dense reconstruction from uncalibrated video sequences*. Phd thesis, Kungl Tekniska Högskolan, Stockholm.

Nocedal, J., Wright, S., & Glynn, P. (2000). *Numerical Optimization* (1 ed.). Springer.

Notni, G., Riehemann, S., Kühmstedt, P., Heidler, L., & Wolf, N. (2004). OLED microdisplays - a new key element for fringe projection setups. *Proceedings of SPIE*, *5532*, 170–177.

Ogawa, H. (1984). Labeled point pattern matching by fuzzy relaxation. *Pattern Recognition*, *17*(5), 569–573.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66.

Papadimitriou, C. H. & Steiglitz, K. (1998). *Combinatorial Optimization* (Dover ed.). Mineola, New York: Dover Publications, Inc.

Park, H. & Lee, J.-H. (2007). B-spline curve fitting based on adaptive curve refinement using dominant points. *Computer-Aided Design*, *39*(6), 439–451.

Pentland, A. P. (1986). Perceptual organization and the representation of natural form. *Artificial Intelligence*, *28*(3), 293–331.

Piegl, L. (1991). On NURBS: a survey. *IEEE Computer Graphics and Applications*, *11*(1), 55–71.

Piegl, L. & Tiller, W. (1997). *The NURBS Book* (2 ed.). Berlin: Springer.

Plass, M. & Stone, M. (1983). Curve-fitting with piecewise parametric cubics. *Computer Graphics*, *17*(3), 229–239.

Popescu, V., Bahmutov, G., Sacks, E., & Mudure, M. (2006). The model camera. *Graphical Models*, *68*(5-6), 385–401.

Posdamer, J. L. & Altschuler, M. D. (1982). Surface measurement by spaceencoded projected beam systems. *Computer Graphics and Image Processing*, *18*(1), 1–17.

Pottmann, H. & Randrup, T. (1998). Rotational and helical surface approximation for reverse engineering. *Computing*, *60*(4), 307–322.

Pottmann, H. & Wallner, J. (2001). *Computational Line Geometry*. Berlin: Springer.

Prautzsch, H., , Boehm, W., & Paluszny, M. (2002). *Bezier and B-Spline Techniques*. Springer.

Qian, X. & Huang, X. (2004). Reconstruction of surfaces of revolution with partial sampling. *Journal of Computational and Applied Mathematics*, *163*(1), 211–217.

Raja, N. S. & Jain, A. K. (1992). Recognizing geons from superquadrics fitted to range data. *Image and Vision Computing*, *10*(3), 179–190.

Ranade, S. & Rosenfeld, A. (1980). Point pattern matching by relaxation. *Pattern Recognition*, *12*(4), 269–275.

Rangarajan, A., Chui, H., & Mjolsness, E. (2001). A relationship between spline-based deformable models and weighted graphs in non-rigid matching. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, volume 1, (pp. I–897–I–904).

Rangarajan, A., Chui, H., Mjolsness, E., Pappu, S., Davachi, L., Goldman-Rakic, P. S., & Duncan, J. S. (1997). A robust point matching algorithm for autoradiograph alignment. *Medical Image Analysis*, *1*(4), 379–398.

Razdan, A. (1999). Knot placement for B-spline curve approximation. Technical report, Arizona State University.

Reinsch, C. H. (1967). Smoothing by spline functions. *Numerische Mathematik*, *10*, 177–183.

Rogers, D. F. & Fog, N. R. (1989). Constrained B-spline curve and surface fitting. *Computer-Aided Design*, *21*(10), 641–648.

Rousseeuw, P. J. & Leroy, A. M. (2003). *Robust Regression and Outlier Detection*. Wiley Series in Probability and Statistics. Wiley & Sons.

Salvi, J., Pagès, J., & Batlle, J. (2004). Pattern codification strategies in structured light systems. *Pattern Recognition*, *37*(4), 827–849.

Sampson, P. D. (1982). Fitting conic sections to 'very scattered' data: An iterarive refinement of the Bookstein algorithm. *Computer Graphics and Image Processing*, *18*(1), 97–108.

Saux, E. & Daniel, M. (2003). An improved Hoschek intrinsic parametrization. *Computer Aided Geometric Design*, *20*(8–9), 513–521.

Scharstein, D. & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, *47*(1–3), 7–42.

Schmähling, J. (2006). *Statistical characterization of technical surface microstructure*. Phd thesis, Ruprecht-Karls-Universität, Heidelberg.

Schmid, C., Mohr, R., & Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision*, *37*(2), 151–172.

Schoenberg, I. J. (1967). On spline functions. In O. Sischa (Ed.), *Inequalities* (pp. 255–291). New York: Academic Press.

Scholz, O., Kostka, G., Jobst, A., & Schmitt, P. (2007). Cost-effective tire geometry using a fixed sheet-of-light measuring assembly. In *Tire technology international. Annual review of tire materials and tire manufacturing technology* (pp. 130–132). UKIP Media & Events Ltd.

Schwarte, R., Heinol, H., Buxbaum, B., Ringbeck, T., Xu, Z., & Hartmann, K. (1999). Principles of three-dimensional imaging techniques. In B. Jähne, H. Haußecker, & P. Geißler (Eds.), *Handbook of Computer Vision and Applications*, volume 1 chapter 18, (pp. 463–484). San Diego: Academic Press.

Schwetlick, H. & Schütze, T. (1995). Least squares approximation by splines with free knots. *BIT Numerical Mathematics*, *35*(3), 361–384.

Sclaroff, S. & Pentland, A. P. (1995). Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(6), 545–561. Finite Elemente ASM.

Scott, G. L. & Longuet-Higgins, H. C. (1991). An algorithm for associating the features of two images. *Proceedings of the Royal Society B: Biological Sciences*, *244*(1309), 21–26.

Shapiro, L. S. & Brady, J. M. (1992). Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, *10*(5), 283–288.

Solina, F. & Bajcsy, R. (1990). Recovery of parametric models from range images: the case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(2), 131–147.

Speer, T., Kuppe, M., & Hoschek, J. (1998). Global reparametrization for curve approximation. *Computer Aided Geometric Design*, *15*(9), 869–877.

Staib, L. H. & Duncan, J. S. (1992). Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(11), 1061–1075. Fourier Descriptor ASM.

Steihaug, T. (1983). The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, *20*(3), 626–637.

Stewart, C. V. (1999). Robust parameter estimation in computer vision. *SIAM Review*, *41*(3), 513–537.

Stockman, G., Kopstein, S., & Bennet, S. (1982). Matching images to models for registration and object detection via clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *4*(3), 229–241.

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*(1), 44–47.

Sullivan, S., Sandford, L., & Ponce, J. (1994). Using geometric distance fits for 3-d object modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*(12), 1183–1196.

Taubin, G. (1991). Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(11), 1115–1138.

Taubin, G., Cukierman, F., Sullivan, S., Ponce, J., & Kriegman, D. J. (1992). Parametrizing and fitting bounded algebraic curves and surfaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '92)*, (pp. 103–108).

Terzopoulos, D. & Metaxas, D. (1991). Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(7), 703–714.

Ton, J. & Jain, A. K. (1989). Registering landsat images by point matching. *IEEE Transactions on Geoscience and Remote Sensing*, *27*(5), 642–651.

Ullmann, J. R. (1976). An algorithm for subgraph isomorphism. *Journal of the ACM*, *23*(1), 31–42.

Umeyama, S. (1988). An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *10*(5), 695–703.

Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(4), 376–380.

Velho, L., Gomes, J., & de Figueiredo, L. H. (2002). *Implicit objects in computer graphics*. Springer.

Wang, W., Pottmann, H., & Liu, Y. (2006). Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM Transactions on Graphics*, *25*(2), 214–238.

Wiora, G. (2001). *Präzise Gestaltvermessung mit einem erweiterten Streifenprojektionsverfahren*. Dissertation, Ruprecht-Karls-Universität, Heidelberg.

Woo, S. & Dipanda, A. (2000). Matching lines and points in an active stereo vision system using genetic algorithms. In *Proceedings of the International Conference on Image Processing (ICIP 2000)*, volume 3, (pp. 332–335)., Vancouver, BC, Canada.

Wu, M.-F. & Sheu, H.-T. (1998). Representation of 3D surfaces by two-variable fourier descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 858–863.

Wu, M.-S. & Leou, J.-J. (1995). A bipartite matching approach to feature correspondence in stereo vision. *Pattern Recognition Letters*, *16*(1), 23–31.

Yin, P.-Y. (2006). Particle swarm optimization for point pattern matching. *Journal of Visual Communication and Image Representation*, *17*(1), 143–162.

Yokoya, N., Kaneta, M., & Yamamoto, K. (1992). Recovery of superquadric primitives from a range image using simulated annealing. In *Proceedings of the 11th IAPR International Conference on Pattern Recognition, Vol. I. Conference A: Computer Vision and Applications*, (pp. 168–172)., The Hague, Netherlands.

Yuen, P. (1993). Dominant point matching algorithm. *Electronics Letters*, *29*(23), 2023–2024.

Zhang, C., Cheng, F. F., & Miura, K. T. (1998). A method for determining knots in parametric curve interpolation. *Computer Aided Geometric Design*, *15*(4), 399–416.

Zhang, D. & Lu, G. (2002). A comparative study on shape retrieval using fourier descriptors with different shape signatures. In *Proc. of the Fifth Asian Conference on Computer Vision (ACCV'02)*, (pp. 646–651)., Melbourne, Australia.

Zhang, Z. (1997). Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, *15*(1), 59–76.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(11), 1330–1334.