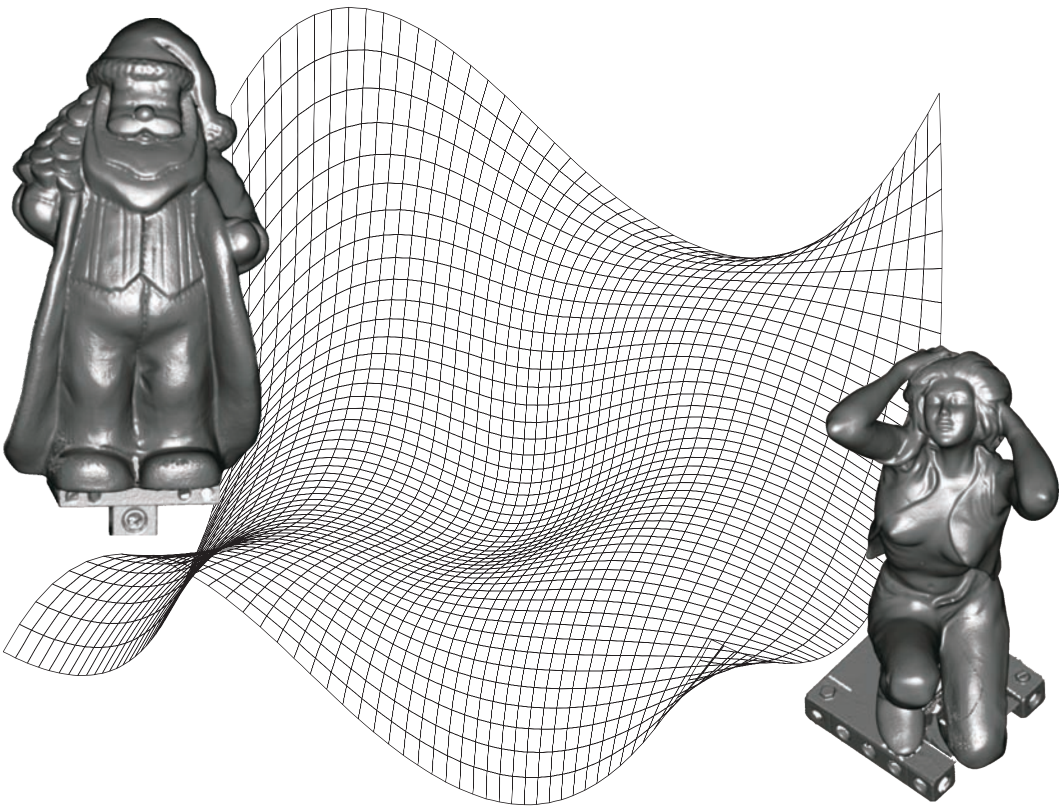
Magdeburger Schriften zur
Visualisierung

Christian Teutsch

Model-based Analysis and Evaluation of Point Sets from Optical 3D Laser Scanners



SHAKER
VERLAG

Abstract

The digitalization of real-world objects is of vital importance in various application domains. This method is especially applied in industrial quality assurance to measure the geometric dimension accuracy. Furthermore, geometric models are the very foundation of contemporary three-dimensional computer graphics. In addition to create new models by using a modeling suite, the use of 3D laser scanners has recently become more and more common. To reconstruct objects from laser scan data, usually very large data sets have to be processed. In addition, the generated point clouds usually contain a considerable amount of errors. Therefore, it is necessary to optimize the data for further processing.

Compared to algorithms that interactively manipulate point clouds through an approximation with polygonal meshes, we aim to automatically correct each measurement individually and directly integrate the methods into the measurement process. In addition to traditional methods which usually assume point clouds as unstructured, this work introduces techniques for the extraction of common data structures from optical 3D scanners. Based on this information, procedures are developed to enable automatable procedures of scan data optimization and evaluation. The feasibility of the proposed methods is shown at the example of different real-world objects and industrial applications.

Contents

Chapter 1

Introduction

Today, industrial processes such as assembly are complex, highly automated and typically based on CAD (Computer Aided Design) data. The problem is that the same degree of automation is also required for quality assurance. It is, for example, a very complex task to assemble a modern car, since it consists of many parts that must fit together at the very end of the production line. The optimal performance of this process is guaranteed by quality assurance systems. Especially the geometry of the metal parts must be checked in order to assure that they have the correct dimensions, fit together and finally work reliably. Therefore, CAD and CAM construction software supports the designer when developing those parts.

Within highly automated processes, the resulting geometric measures are transferred to machines that manufacture the desired objects. Due to mechanical uncertainties and abrasions, the result may differ from its digital nominal. In order to automatically capture and evaluate these deviations, the manufactured part must be digitized as well. For this purpose, 3D scanners are applied to generate point samples from the object's surface. Particularly tactile and optical scanners are used to obtain the 3D measures. The small size and simple construction of the wearless working optical scanners enable a flexible application, and thus replace the approved tactile systems more and more.

Besides industrial manufacturing, the digitization of real-world objects has many other application fields. For example, the conservation of cultural heritage and the worldwide exchange of archaeological findings becomes possible on the basis of digital models [LPC+00, BMM+02]. 3D scanners are even used for computer games and animated cartoons. In this case, the digitization of hand-modeled sculptures simplifies the production process.

In practice, the costs for a complete measuring system are often an important criterion. Therefore, in many cases the optimal technical solution is not used, knowing that the resulting problems must be compensated with software tools at many different places. This practical fact also requires the development of modular algorithmic tools applicable to a variety of measuring tasks. Thus, this work considers scan data processing from a practical view and discusses the entire process pipeline from the data acquisition to the data representation and interpretation.

1.1 Motivation and Goals

In contrast to digital geometric modeling, the data obtained from optical 3D scanners are often far from being perfect. The results are usually point clouds that contain several errors caused by system and measuring principle specific characteristics. Additionally, the data is affected by environmental influences such as unfavorable lighting conditions, dust or vibrations. Measurements on the generated data may lead to incorrect results. Therefore, procedures for the point cloud optimization, evaluation and inspection are needed that are robust against these influences. While a smooth and aesthetic visualization is desired for most computer graphics applications, industrial measurements primary require fast and automated procedures with a high reliability.

Classical approaches assume a point cloud as an unstructured 3D point set. In order to provide a structure, polygonal meshes are typically approximated. The large variety of existing methods for polygonal mesh processing produces very aesthetic 3D models, but often requires user interaction and is limited in processing speed and/or accuracy. Furthermore, operations on optimized meshes consider the entire model and pay only little attention to individual measurements. However, many measurements relate to parts or single scans with possibly strong differences between successive scans being lost during mesh construction.

In particular the large amount of measuring data, additionally disturbed by noise and errors, complicates the application of automated procedures. Thus, the goal of this work is the development of practical algorithms for the efficient processing, evaluation and management of 3D scan data obtained from optical measuring systems.

Since it can be assumed that most of the errors originate from the measuring principle, an optimized point cloud processing should start with the analysis of the data acquisition procedures. In the following steps, the obtained characteristics can be taken into account when performing data optimizations and evaluations. Furthermore, existing algorithms mainly consider unstructured point clouds although the scanned data is structured through device properties and measuring principles. By exploiting this underlying structure, intrinsic system parameters can be extracted and employed for a fast and problem-specific data processing. Therefore, this work devises new and adjusts approved algorithms while touching different research areas, including the technical foundation, data acquisition and visualization.

In summary, the following assumption motivates this work:
By analyzing and consequently exploiting the underlying measuring concepts, alternative methods for adaptable and more efficient 3D scan data representations and their evaluation can be found.

1.2 Results

Based on this motivation, the following brief overview of the concepts that significantly contribute to this work, is presented below:

- A review of common optical measuring principles and selected aspects of point cloud evaluation underlines the need for modular and adaptable scan data processing methods for practical applications. Therefore, different measuring principles are formalized, which enables the adaptability of the devised methods.

- In order to extract utilizable additional scan data information (*system information*), the optical 3D data acquisition and the following post-processing is analyzed using the example of two employed 3D scanners. Therefore, approaches for the analysis of 3D measuring systems with regard to relevant system parameters and data structures are presented.

- The precision and acquisition quality of the generated 3D point sets are formally determined to constitute the basis for the post-processing optimization and evaluation procedures.

- Based on the observed scan data systematics, NURBS curve and NURBS surface-based strategies for scan data representation, optimization and analysis are presented.

- Existing methods for point cloud optimization and evaluation are adapted and extended by employing the adaptively extracted additional system information.

- Potential applications in different domains are discussed and a variety of examples are given. These examples should give an orientation on how the methods developed can be applied for real application problems. Furthermore, a subset of the proposed scan data methods has been successfully integrated into industrial measuring systems.

1.3 Thesis Structure

This thesis gives an outlook on selected aspects of point cloud post-processing, for example 3D meshing and point cloud representation. In the second step, new approaches for point cloud optimization and adjusted preprocessing are developed with focus on the measuring principles and additional system information.

Chapter 2: Algorithmic and Technical Foundations

At the beginning of this thesis, we consider the state of the art of optical 3D measuring principles, such as light-section, fringe projection and photogrammetry and discuss their application fields. In the second part, typical methods for 3D point cloud processing, particularly 3D meshing of unstructured data, are discussed. This chapter constitutes the algorithmic and technical foundation and supports the reader to relate this work to the variety of different approaches.

Chapter 3: Scan Data Acquisition

Chapter three introduces the procedures for the acquisition of 3D point clouds from laser triangulation. Using two employed 3D scanners, calibration procedures are proposed, which yield the basis for the data acquisition from optical sensors. In further steps, algorithmic approaches to obtain spatial coordinates from digital images and laser light are proposed. Based on this discussion, the influences of errors, affecting the resulting 3D point set, are described. In the remainder, the contained noise is quantified for further evaluation. Finally, a scan data structure for optical 3D scanners is proposed, which is obtained from the measuring principle and employs the system configurations.

Chapter 4: Curve-based Scan Data Processing

In the fourth chapter, we describe a subdivision of the entire point cloud by exploiting the scan data structure obtained from the data acquisition stage. Based on the measuring principle, a point cloud is represented as a set of scanned profiles, which are referred to as *scanlines*. In order to derive an analytical representation for further procedures, the scanlines are approximated by NURBS curves. The processing of those scanlines already allows a data evaluation during its acquisition, which significantly reduces the computation time for the entire evaluation process. The employed algorithms and features automatically obtained from the scanlines are finally discussed in case studies.

Chapter 5: Surface-based Scan Data Processing

The extracted scan data structure does not only provide point neighborhoods on scanlines derived from the measuring principle. Additionally, the scanning system provides information on neighboring scanlines obtained from defined sensor movements between them. Thus, this chapter introduces methods to represent a set of scanlines as a regular grid. The grids are used for a real-time, in-process visualization on the one hand, and as the basis for the approximation of NURBS patches on the other hand. Compared to the NURBS curves introduced in the chapter before, the NURBS patches are employed to obtain more global surface measures between single scanlines.

Chapter 6: Point Cloud Processing

In addition to the data structures, which represent measured point clouds as scanlines and grids of them, this chapter discusses methods for the global point cloud processing. Within this abstraction level, interrelationships between the grids and scanlines from different sensors and measurements are established. The data representation with tree and graph structures is discussed in particular. Furthermore, methods for point cloud correction and their visualization are proposed. The application of these methods and the results are discussed in several case studies.

Chapter 7: Practical Applications

The procedures proposed have proven their feasibility in a variety of different industrial applications. Therefore, two representative measuring systems and their measuring tasks are introduced in this chapter. Both systems employ an optical measuring concept that allows to apply the methods proposed in this work. Thus, the integration of the scan data algorithms for an efficient and robust data evaluation are discussed here.

Chapter 8: Concluding Considerations

Chapter 8 summarizes the dissertation by reviewing the achieved results. Furthermore, some points of criticism are given and additional future directions are outlined.

Chapter 2

Algorithmic and Technical Foundations

Processing point clouds of real-world objects, generated with 3D scanners, is affected by a variety of errors. This includes systematic measuring uncertainties, noise, outliers, gaps and holes. Particularly this imperfection is the main difference to artificially generated point clouds with certain guarantees, for example on point density or quality. Thus, the identification of optimization potential of point clouds obtained from 3D scanners requires a discussion of related algorithms for data processing and measuring systems with their underlying technical principles. Therefore, this chapter introduces the most common 3D scanning methods in the first part.

A set of single, independent 3D points can hardly be processed without additional information about the underlying shape or neighborhoods. Thus, many applications necessitate data structures providing connectivity information (e. g., for surface shading). Therefore, in the second part of this chapter, common algorithms for surface meshing and visualization are considered. Taking this as the basis, methods for point cloud manipulation and optimization are also discussed.

This chapter gives an overview on common measuring principles and point cloud processing. It introduces a variety of different approaches in both fields and serves as the algorithmic and technical foundation for this work. Due to the fact that there is a wide variety of different approaches, only the most common techniques are introduced.

2.1 Survey of Common 3D Measuring Principles

The majority of the employed scanners use optical methods for capturing and measuring three-dimensional surfaces. This section introduces common measuring principles for the generation of 3D point clouds and discusses their pros and cons and their fields of application.

In general, capturing and measuring spatial structures can be divided into three different basic approaches: triangulation, interferometry[1] and time-of-flight methods. Due

1 Interferometry is the observation of interferences. Interference is the overlap of coherent waves that meet at a common point in space. Coherent light is for example emitted by laser diodes.

to their flexible application, optical triangulation methods have a leading position. This includes light-sectioning methods and photogrammetrical approaches, which are introduced in the following sections.

2.1.1 Surface Measuring using the Triangulation Principle

The principle of triangulation has already been used in 1609 by Johannes Kepler[2] to explain the movements of planets in the solar system. In 1617, the Dutch mathematician Willebrord Snellius[3] published this principles in his work "Eratosthenes Batavus".

The principle depends on the fact that a triangle is explicitly defined by the length of one side and the adjacent angles. The remaining sides and the third angle can be derived from that [BSG+03]. Optical measuring systems use this principle to determine the spatial dimensions and the geometry of an item. Basically, the configuration consists of two sensors observing the item. One of the sensors is typically a digital camera device, and the other one can also be a camera or a light projector. The projection centers of the sensors and the considered point on the object's surface define a triangle (see Fig. 2.1). Within this triangle, the distance between the sensors is the base (b) and must be known. By determining the angles between the projection rays of the sensors and the basis (α, β), the intersection point is calculated by triangulation.

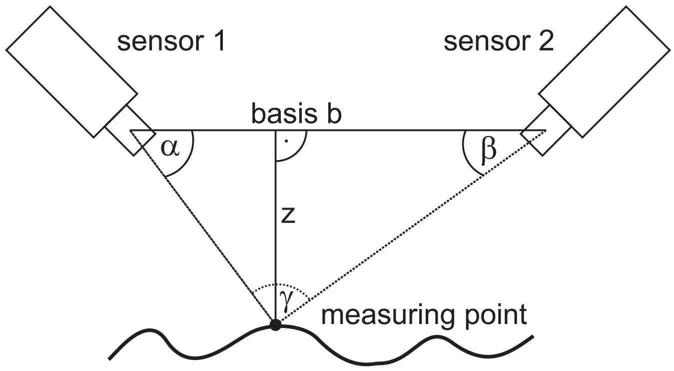


Fig. 2.1: Triangulation principle. If two sensors observe the same point in space, then the point's spatial coordinate can be computed with angular relations in a triangle. The necessary input parameters α, β and b are obtained from a prior calibration step.

The computation of the angles α and β as well as the length b is based upon a prior system calibration, where these sensor parameters are determined. The necessary triangle height z is finally calculated with the general triangulation equation:

$$z = \frac{b\ \sin\alpha\ \sin\beta}{\sin\gamma}, \qquad \gamma = \pi - \alpha - \beta\,. \tag{2.1}$$

2 Johannes Kepler: German astronomer and naturalist (⋆1571, †1630).
3 in fact Willebrord Snel van Rojen (⋆1580, †1626).

Typically, active and passive methods are distinguished. Passive methods get their information from camera devices and the evaluation of pixel positions in the respective images planes (*photogrammetry*). On the other side, active methods require a special object lighting with structured light (light pattern with known geometry, e. g., points, lines, random patterns). The observed surface point is lighted by the projector and captured by the camera, whereas the lighted positions enable an easy identification of the considered surface point in the camera images.

The majority of scan systems in industrial applications is based on active triangulation methods, which may differ in the configuration and the used sensors. The following section introduces common methods.

2.1.2 The Light Section Method

The most popular method for the optical capturing of shapes is light sectioning, which dates back to the work of Shirai and Suwa [SS71]. The identification of corresponding positions captured from two positions is realized by projecting structures with known geometry onto the surface, which are easy to identify. Light section means the projection of a point, which can be seen as spatial line intersecting the surface. Due to the reflection of the point on the surface, observed from another position, the corresponding 3D coordinates are computed. The procedure is also called structured lighting and can be extended to line projection, which enables the computation of multiple point coordinates in one step. Furthermore, the use of line patterns allows to compute multiple lines. The object is usually moved to capture the entire surface. There are different approaches to project lines and similar patterns that are explained in the following sections.

Light Sectioning with a Laser Line

For the measuring of complete surfaces, the projection of single points is very time-consuming. By projecting a line, significantly more points on a surface are represented (see Fig. 2.2). Therefore, a light beam is expanded using beam shaping optics (e. g., cylindrical lenses, refractive lens systems). The spatial position and orientation of the resulting light plane[4] is determined within a calibration procedure. The plane intersects the object surface following a profile line. Looking from the laser position, this line is straight. But looking from the viewing position of the camera, the line is curved due to the principle of stereoscopic viewing and depending on the surface topology. The amount of deformation in the camera image is a measure for the real object shape.

4 The light plane and the respective projection and viewing beams is a simple model of geometrical optics. This model solely considers the central beams of a beam bundle, which finally converges in the image and respective object coordinates.

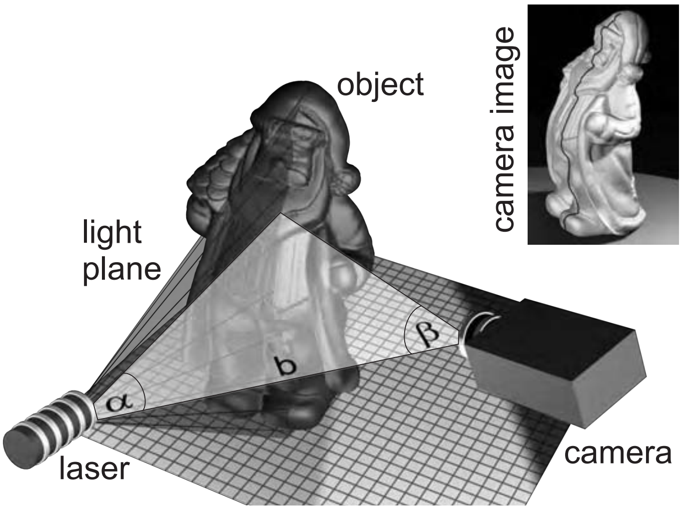


Fig. 2.2: Light section principle with a line laser. The resulting light plane “intersects” the object’s surface and the resulting deformed contour is observed by a camera.

The corresponding pixels of the deformed line are segmented and evaluated by image processing algorithms. Due to the strong contrast between color/intensity of the laser line and the background, a reliable segmentation is possible. Based on the known geometry of the sensors and the projection parameters, one 3D depth coordinate is calculated for each identified pixel position. After each measuring step, the object or the sensors are moved until the entire surface is captured.

Light Sectioning with Fringe Projection

Comparable to considering just one line, the projection of multiple lines can increase the measuring speed. This is achieved by using a multi-line projector (e. g., a diascope) or by using common light-beamers that project a set of alternating black and white stripe patterns (see Fig. 2.4(a)). Each black-white border exactly behaves like a single projected line as mentioned in the last section.

But in the following image processing step, the lines must be distinguished from each other to correctly map a detected line to the corresponding projection. This problem is solved by encoding the light pattern [Maa97]. In this process, a set of images with a successively increasing number/density of black and white stripes is projected. In the next step, the corresponding position for each detected light section to its projected black/white pattern is determined. This is equal to a symmetric binary code, (*Gray code*, see Fig. 2.4(b)).

In principle, the system configuration consists of a camera and a light beamer. Depending on the type and quality of the beamer, it can be time-consuming to calibrate the position of the beamer (light origin/orientation) to the camera. Therefore, normally a second camera is used, which also observes the pattern. The triangulation is then computed between the cameras instead of a projector and a camera. This approach is

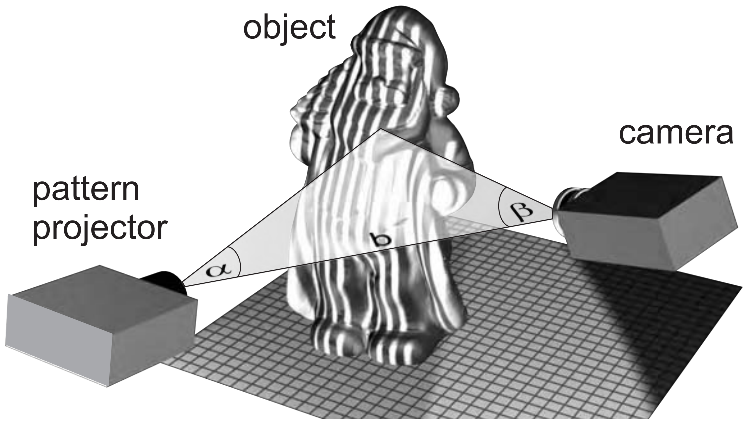


Fig. 2.3: Principle and system configuration for triangulation based on fringe projection.

also known as photogrammetry (see Sect. 2.1.3) [Bre93]. The determination of correlating stripes in both images is respectively carried out analogously to the Gray code approach.

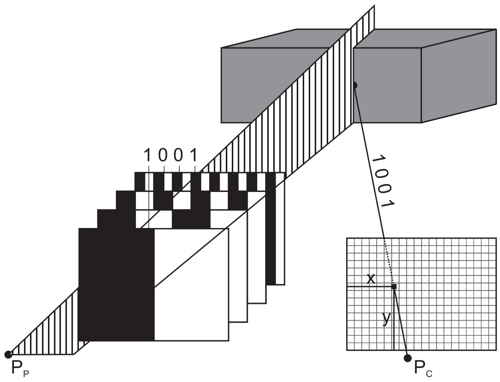
It is also possible to use point patterns instead of stripes. For the measuring of multicolored surfaces, a sequence of red, green and blue patterns is projected. Depending on the surface color, more or less light is emitted, which also allows to determine the real surface color. The performance and stripe density of the coded light section approach is limited by the resolution of the projector. A further increase of the resolution is possible by using a phase-shift technique, explained in the next paragraph.

Phase-Shifting

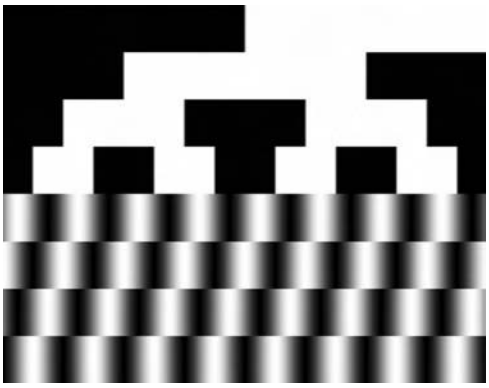
The *phase-shift* method is a modification of the coded light-sectioning [SWPG05]. The principle configuration consists of a video projector and a digital camera (see Fig. 2.4).

In contrast to coded lighting-sectioning, where a contour can only be measured at sharp black/white borders, phase-shifting achieves a higher resolution. Therefore, the video beamer projects a sequence of sinusoidal stripes. Within each step, the sine pattern is horizontally shifted by known angles (phase-shifting), which are parts of a sine phase (see Fig. 2.4(b)). Due to the phase-shifting of the sine function, the gray values in the captured images change cosinusoidally, and depending on the surface topology the projected pattern is additionally deformed. The deformation is a measure for the topology change and is determined by the quotient of the sinusoidally and cosinusoidally shifted signals, which is the tangent. Finally, the arc tangent yields the phase angle (see Fig. 2.5(b)), which describes the phase-shift in portions of the sinusoidal stripe period ([Str93]). This principle enables the signal subdivision into intensity, amplitude and phase, and thus, its reconstruction.

In contrast to coded light-sectioning the resolution increases, because a phase value exists for every image pixel and it exactly represents one position in the (originally)

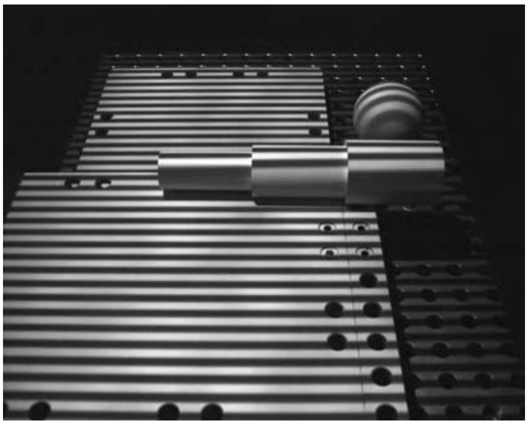


(a) fringe projection principle.

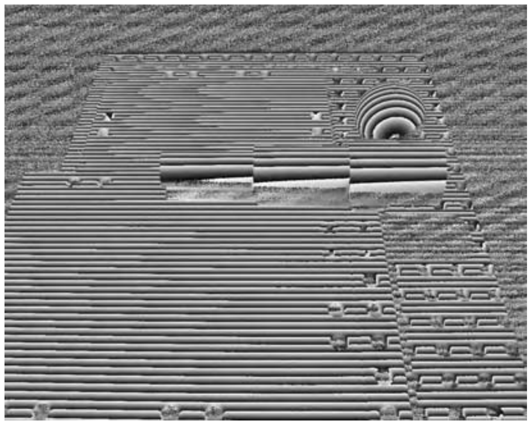


(b) gray code sequence.

Fig. 2.4: Fringe projection principle: P_P and P_C indicate the centers of the projector and the camera (a). The combination of a phase-shifted sinusoidal pattern and a gray code is shown in (b).



(a) fringe projection.



(b) phase angles.

Fig. 2.5: Illustration of the projected fringe pattern (a) and a higher resolved representation after phase shifting and the obtained phasing (b).

projected pattern. The assignment of pixel positions and lighting directions enables a higher precision, but is unique only within one period. This problem is solved again with gray codes (see Fig. 2.3).

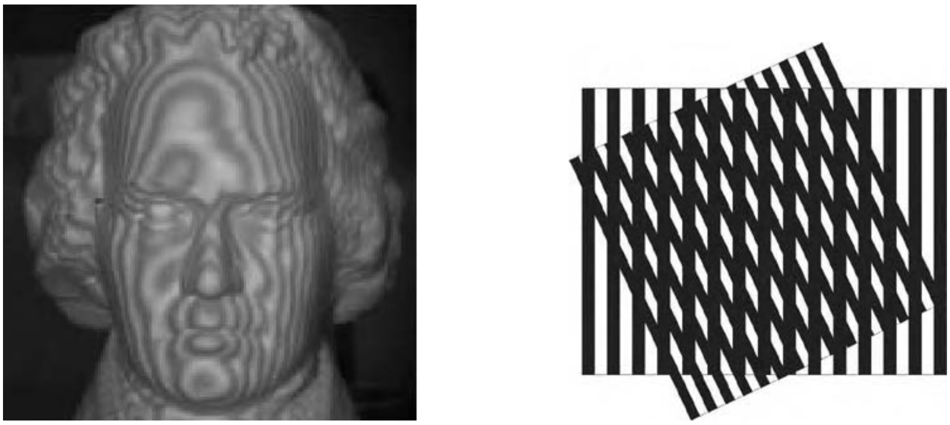
There is still a variety of other approaches, which differ in the way the light pattern is generated or shifted [Bre93]. But the basic principle and the effectiveness is similar to the described ones.

The Moiré Approach

Instead of projecting light planes, it is also possible to project a grid structure onto the object surface and map it to a reference grid, which results in the Moiré-effect (see

Fig. 2.6(a)). Moiré is a uniform pattern that arises from the overlap of two marginally differing structures, which finally leads to geometric interference (see Fig. 2.6(b)).

This method is also called mechanical interferometry. The first scientific application of the Moiré effect dates back to Lord Rayleigh[5] and his work [Ray74] in 1874. He predicated that this interference can be used as the test of irregularity ruling of grating or deflection. In practice, the Moiré effect arises, for example, by looking through consecutively standing fences or folded curtains.



(a) (b)

Fig. 2.6: The Moiré effect (a) (from [Maa97]) and an exemplary Moiré pattern (b).

The captured camera image already contains depth information, which enables at least a visual interpretation. In order to derive depth values, a stripe evaluation method is needed, which can be phase-shifting. Due to blurring effects, the possible precision is relatively small. This method is hardly used for industrial geometry measurements, but the resulting Moiré stripes are used to observe motions, rotations, curvatures (e. g., deformations, bumps), and derived measures (e. g., expansions).

2.1.3 Photogrammetric Approaches

In this scope, *photogrammetry* means the usage of at least two camera devices. Thus, these methods are also known as *Stereo-/Multi-View-Vision* [Bre93]. The triangulation is computed between the cameras and the observed surface point. Again, there are active and passive methods. Active approaches project an additional light pattern onto the surface to resolve possible ambiguities. For the reconstruction of the spatial location and shape from photographs the geometric projection laws of a photograph must be known. The used cameras produce images that represent a central projection with sufficient accuracy [Dol97].

5 John William Strutt Lord Rayleigh (⋆1842, †1919).

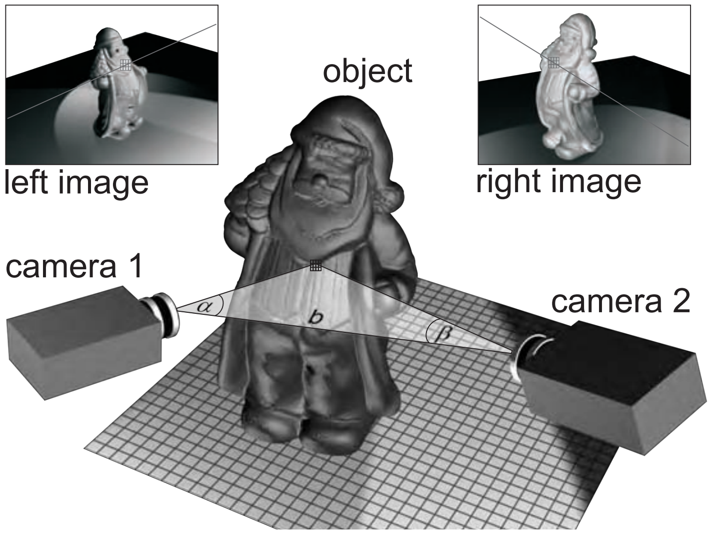


Fig. 2.7: Principle of photogrammetric methods. A point in space is observed by at least two sensors (e. g., cameras). Based on a prior sensor calibration, the 3D coordinate of the point, relatively to the sensors, is computed with triangulation.

In contrast, passive methods do not use additional information. The object to be measured is only observed by cameras. For triangulation, a considered surface point must be identified in both images, which is not trivial. Usually, statistical methods (e. g., correlation) are used to detect corresponding pixel positions. Typically, block-matching approaches are applied to detect and compare image regions. The comparison is often based on cross-correlation. When two corresponding blocks (e. g., 15×15 pixel) have been found, there is a statistical probability that the pixel positions in the middle of the blocks are corresponding [EEAHM05]. In the next step, the intersection of the beams from the projection centers of the cameras "through" the image planes at the corresponding pixel positions is computed and results in a triangulation (see Fig. 2.7). An efficient block-matching approach is introduced by SCHROEDER [Sch00]. The image is evaluated in different resolutions, whereas the algorithm starts at a low resolution and corresponding image regions are identified in larger areas. In the following steps, this process is repeated with higher resolutions in the previously detected corresponding areas. This hierarchical approach leads to higher processing speed and robustness against false correlations.

Furthermore, it is useful to involve the camera positions and orientations. Based on the epipolar geometry between two cameras, the search area is drastically reduced to only one line, the so called epipolar line. Therefore, a ray from the projection center to the object of the first camera is mapped to the image plane of the second camera. This line is the epipolar line and it represents all possible positions for the corresponding pixel in the first camera image.

2.1.4 Tactile Coordinate Metrology

The 3D measuring method established in the industry is the mechanical pointwise surface sampling with high-precision probes. These are mounted on flexible joints with known relative and absolute positions. The probe is guided along the surface, and single 3D positions are generated. In the last years, coordinate measuring machines have evolved from origins founded in simple layout machines and manually operated systems to highly accurate, automated inspection centers. Major factors in this evolution have been the touch trigger and other forms of inspection probes, and subsequent innovations such as the motorized probe head and automatic probe exchange system for unmanned flexible inspection. The technology is sophisticated and achieves a high precision ($< 5\,\mu m$). But nevertheless it is very expensive, slow, and the mechanic is complex. Furthermore, it cannot be used for pliant surfaces and can not be integrated into online production processes. Therefore, optical metrology is becoming an important alternative. There are combinations that use an optical scanner head or heads with small electric charges (e. g., electron microscope) instead of a probe.

2.1.5 Further 3D Measuring Methods

Besides the methods that are based on optical triangulation or mechanical sampling, there is a variety of other methods for displacement measurements. This includes time-of-flight methods, confocal microscopy, and shape-from-shading.

Time-of-Flight Methods

To capture large areas and buildings, time-of-flight approaches are used very often. In these cases, a distance is measured based on the runtime of an emitted signal. Typical systems employ ultrasonic [WS05], positron emission [SKP+06] infrared or coherent laser light [SGS05, MAB06]. In this respect, a light impulse is emitted at a certain time. A part of the reflected light is received from the object and detected by a light-sensitive unit. Based on the time delay and the known speed of light, the distance is computed. Recent developments allow the real-time 3D data acquisition in limited resolution also in the close-up range [BOW+04].

Measuring devices that use infrared light work independent from the surface color. This is due to the fact that their wavelength is larger than that of the visible light. These approaches require an acutely sensitive hardware to achieve an accuracy in the cm range, but they have a large measuring range (>80 m). Ultrasonic devices are applied in parking assistants and for underwater measurings.

Another interesting approach for time-of-flight measuring is used in the LIDAR project (Light Detection And Ranging). Based on emitting and receiving light impulses, the pollutant and gas concentration in the earth atmosphere is captured together with its spatial expansion [FS07].

Confocal Microscopy

The principles of confocal microscopy were invented and also patented by Marvin Minsky in 1956 [Min61]. Recent developments in the computer and laser technologies allow the conception and application of confocal microscopy for the 3D object inspection, especially of microscopically small structures [SN06].

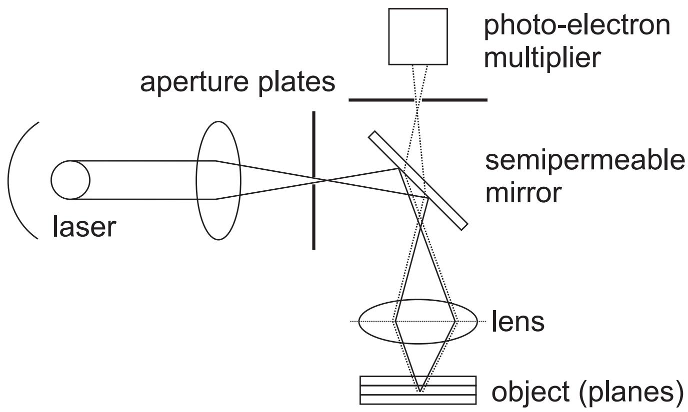


Fig. 2.8: Scheme for the confocal laser-fluorescence microscopy.

In contrast to the traditional microscopy, light that is not originating from the focal plane, is suppressed. Based on a pinhole, the illumination of the observed object is limited. A second pinhole finally reduces the viewing field to a single point. Due to the system configuration, both apertures and one point of the object are confocal in the focal plane. The diameter of the aperture is chosen very small so that light from outside the focal plane can be suppressed (see Fig. 2.8).

With this principle, only one single (image) point (originating exactly from the focal plane of the lenses) can be measured. For a complete survey, the object must be sampled by laser point by point. By changing the focus, the next depth plane can be measured. The test item is additionally coated with a fluorescent substance that is activated by the laser light. The emitted photons are detected by a photo-electron-multiplier.

Shape-from-Shading

3D surface shading is widely used in computer graphics. Based on the surface normal and the light direction and intensity, a color value is assigned to each point on the surface in order to produce a spatial impression.

The inversion of this principle is called *Shape-from-Shading* and dates back to the work in [Rin66]. Shape-from-Shading techniques try to determine the local surface orientation from color gradient information. Only based on color information, this problem cannot be solved. Therefore, additional estimates must be defined. This includes the estimation of an illumination and reflection model, estimates for the local surface continuity and even for the basic shape of the entire surface. Due to these limitations, shape-from-shading methods cannot be used as a general tool for 3D surface reconstruction [Maa97].

2.2 Measuring Systems Summary

The introduced 3D measuring methods are applied in different areas. On the one hand, they are applied for measurements in industrial quality assurance, and on the other hand, for 3D surveying and visualization in external systems.

Precise geometry and surface measurement is typically based on tactile methods and optical triangulation. Especially optical 3D metrology methods are becoming more and more important due to the fact that they are significantly faster and less expensive. Besides several types of fringe projection, laser triangulation is widely-used. Optical methods work contactless (e. g., for flexible surfaces) and wearless, but are not suitable for translucent surfaces.

Due to its simple configuration, the laser triangulation method is a robust and cost-efficient method. With the advancement of digital camera hardware and laser equipment, compact measuring systems can be designed. Therefore, these methods are predestined for applications in online production processes. The tactile probes of coordinate measuring machines can also be replaced by laser sensor heads.

The phase-shift method is a very fast and precise technique. It is mostly applied to homogeneous, smooth surfaces (e. g., car body panels). Sharp or prominent surface discontinuities can lead to a misinterpretation between fringe projection and phase assignment. Besides a video projector, the hardware requirements are low.

Photogrammetric approaches, which employ additional lighting, are also robust and precise. The use of multiple camera devices causes a more complex system configuration and a time-consuming calibration procedure. Naive approaches without additional projected geometry information strongly depend on the texture and color information contained in the captured images. Typically, photogrammetry is used for area measurements, where a very short measuring time is needed. Principally, the accuracy of the measuring devices strongly depends on the system calibration. Furthermore, the size of the measuring area/volume and the resolution of the hardware devices (e. g., cameras, linear motorized axes) affect the overall precision. An additional limitation is given by the environmental and lighting conditions and the reflection properties of the objects. In practice, not the measuring accuracy but the measuring uncertainty is specified, which is more significant. Empirical studies show typical uncertainties up to $5\,\mu m$.

The Moiré technique is not suitable for precise measurements, because of a very time-consuming interference evaluation, which always needs an adjusted calibration procedure. Nevertheless, this method is used to detect and observe dynamic scenes [LCK06] and surface deformations.

Tactile coordinate measuring machines have been established for 3D inspection tasks since many years. They reach uncertainties lower than optical metrology (< 1–$5\,\mu m$) and can measure translucent surfaces. On the other side, they are relatively inflexible due to the static configuration, and also very slow. Furthermore, touching the surface is not possible for many objects. Recent systems have combined measuring heads with probe tips and optical sensors.

Confocal microscopy is used to capture microscopically small objects. This includes cells, coatings, and micro-structured surfaces. Compared to light-microscopy, the depth resolution is drastically increased, but strongly depends on the precision of the employed optics and lenses (1–100 nm).

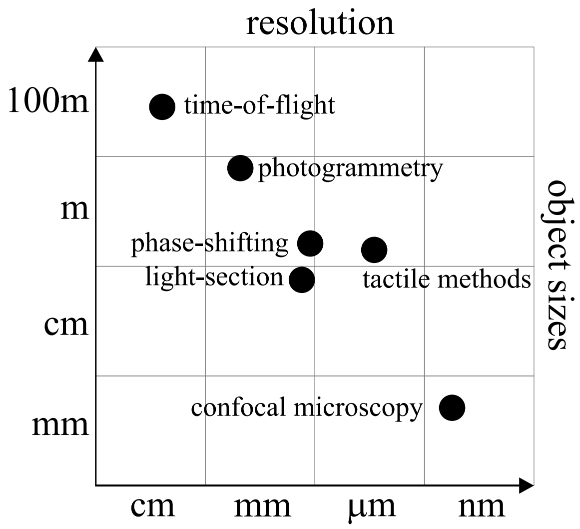


Fig. 2.9: Measuring systems classification. The marked positions in the diagram represent the typical application fields.

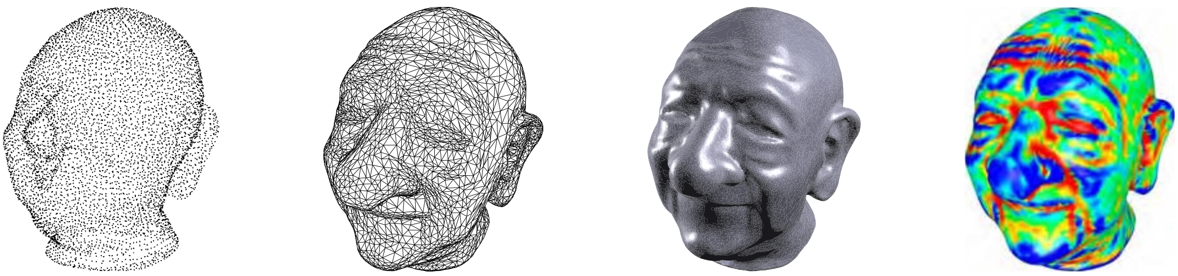
Besides the mentioned methods for 3D metrology, there are further existing approaches, which are partly based on the described ones or are used for special measuring tasks.

The generated point sets, so-called *point clouds*, have to be processed and evaluated. This includes correction, measurements (e. g., dimensions), and visualization. Typically, there is no analytical description for the underlying object geometry, which finally means that it must be approximated. The most common tools for this approximation are triangulation or tessellation.

2.3 Surface Reconstruction

Surface reconstruction is increasingly important in geometric modeling to generate surfaces from data points captured from real objects, often by optical 3D scanners and other technologies as discussed in the previous section. Industrial applications include reverse engineering, quality assurance, metrology and product design.

There are different basic principles to generate a surface model from a set of 3D points (see Fig. 2.10 (a,c)). An intuitive approach is a spatial subdivision of the space into uniform cells (voxels). A volumetric model is generated by only selecting occupied cells. The resulting model is relatively coarse but it can be computed very fast with either coordinate maps or octrees [TO04].



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

Fig. 2.10: Processing point clouds: captured measuring points from the surface (a), polygonal representation (b), shaded model (c) and detecting features like Gaussian surface curvature (d).

Polygonal meshes allow a significantly more precise description. Usually, triangle meshes are used due to their unique and advantageous properties. They represent the simplest polygons, consisting of only three coordinates and they are always planar and convex. Planarity allows linear relations and interpolations between the vertices. Furthermore, their convexity enables their connection to more complex structures. These properties enable the combination of triangles to a polygonal mesh that describes more complex models. This is called triangulation (see Fig. 2.10(b)), and the generation process is called meshing or tessellation. On the basis of a triangulated surface, an optimized visualization as well as the correction (e. g., noise smoothing) and optimization (e. g., thinning) of the approximated polygonal point cloud model are possible.

There are three general surface meshing concepts: *Marching-Cubes*, *Delaunay triangulation/tetrahedrization*, and the fitting of *parametric surfaces* (e. g., B-Splines or Bézier representations), whereas combinations are also common. The following section briefly describes these basic techniques and associated algorithms.

2.3.1 The Marching Cubes Technique

Marching Cubes is one of the best known algorithms of surface construction used to display 3D data. This algorithm produces a triangle mesh by computing iso-surfaces[6] from discrete point cloud data. By connecting the patches from all cubes on the iso-surface boundary, a surface representation is generated. In 1987 Lorensen and Cline introduced this technique [LC87].

The algorithm is based on a space subdivision into cells, whereas the cells can be assumed as small cubes (voxels). The center points of the voxels constitute the corners of the cells. For every cube corner the location to the object surface is determined. Therefore, the iso-value at the corner position is computed by solving the iso-function for this position. If the resulting value is greater than the assumed iso-value, the cube corner is outside the surface. A special case is equality, where the position is defined inside. These values are computed for all cube corners to determine at which positions the surface is intersected. The intersection point is computed by interpolation along the edges between the iso-values at the corners.

In the following steps, each cube is replaced by a set of polygons, depending on which cube edges have intersected the assumed iso-surface. For each cube $2^8 = 256$ possible intersection variants exist that can be reduced to 15 by eliminating symmetric variants. The authors propose a certain set of polygon approximations (see Fig. 2.11(a)). By adding or omitting cubes with all edges/corners inside and outside respectively, the literature proposes 14 or 16 variants [SLS07]. Due to the fact that neighboring cubes share the same edges the mesh connectivity among them is consistent. But the algorithm also produces gaps, because not all polygon variants are consistent to each other. This method is applied to all cubes intersecting the surface. As a result, a triangular polygon mesh is produced, which is typically used to visualize the set of 3D coordinates (see Fig. 2.11).

Precise Marching Cubes [AC98] extended the original algorithm by trilinear interpolation and adaptive error-controlled refinement of surface patches inside of surface-containing cells. As a result, the precision as well as the smoothness of the extracted iso-surfaces could be improved. Unfortunately, this method creates a lot more triangles, requiring a much longer calculation time. Furthermore, Nielson's Dual Marching Cubes approach also connects adjacent surface cells to quadrilateral surface patches and then iteratively relaxes the extracted surface constrained to a binary volume [Nie04a].

In practice, the analytical description of the topology or an iso-surface of the scanned objects is unknown. Therefore, an approximation is needed to successfully apply the Marching Cubes. For homogeneous and smooth surfaces without sharp edges and corners (*soft objects*), Wyvill et al. propose the approximation of a distance function comparable to that of iso-surfaces [WMW86]. For each point, the local neighborhood

6 An iso-surface is a surface that represents points of a constant value. An iso-surface of a 3D distributed quantity f implicitly follows from the constraint $f(p_{x_{iso}}, p_{y_{iso}}, p_{z_{iso}}) = f_{iso}$ and defines the spatial coordinates p_{iso}, where the function f has the iso-value f_{iso}. For example, the surface of a sphere is an iso-surface of the sphere function $f(x, y, z) = x^2 + y^2 + z^2$.

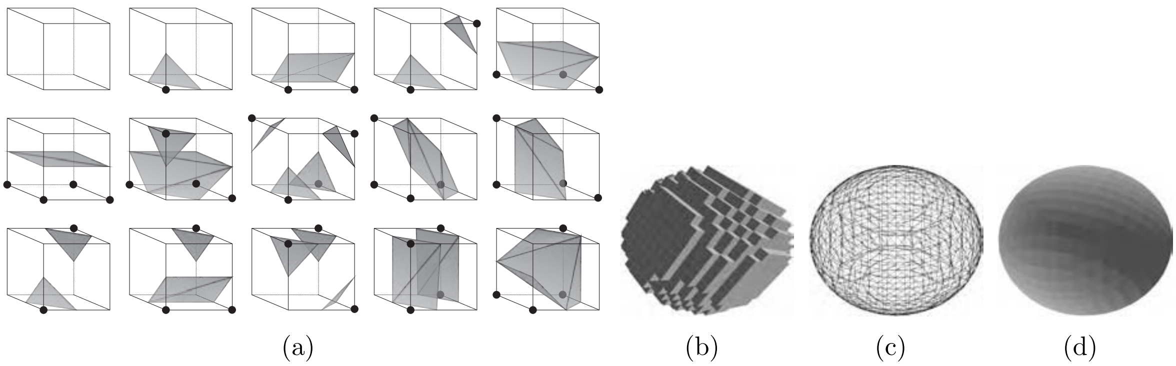


Fig. 2.11: Triangulation of the sphere surface with the marching cubes approach: 15 possible polygon variations that can approximate the surface cell (marked vertices are inside the surface) (a), voxel intersecting the surface (b), approximated triangle mesh (c), final shaded model (d).

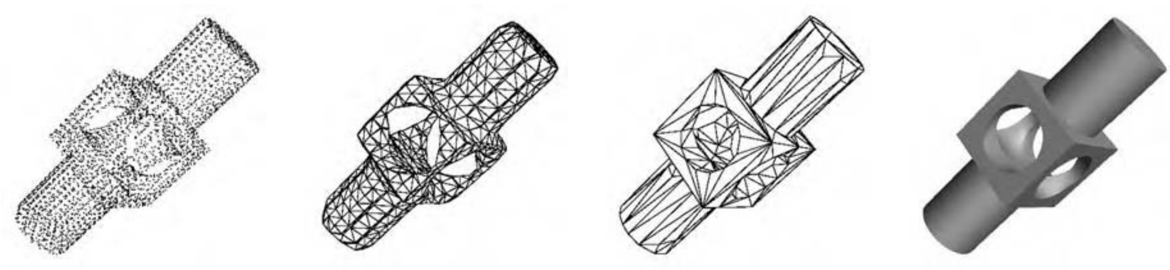
is computed and approximated by a cubic function C (Eq. 2.2). The influence of neighbors is limited by their distance r that has to be smaller than a predefined radius R. The considered point itself has the maximum weight of 1 which is non-linearly decreasing with increasing distance to produce smooth transitions. After computing the iso-function, for each point (or for regions, for more efficiency) all cells that potentially lie in the influence region are detected.

$$C(r) = 2\,\frac{r^3}{R^3} - 3\,\frac{r^2}{R^2} + 1 \tag{2.2}$$

For every cell corner, the function is evaluated analogously to the standard Marching Cube approach. But in contrast, another meshing scheme is proposed that uses only seven variating polygonal approximations. The generated polygons are simply divided into triangles by connecting the corner points with their mean. This basic procedure cannot handle all kinds of polygons, but is sufficient to create smooth surfaces.

A more complex algorithm to mesh unorganized points, which also applies Marching Cubes, was presented by Hoppe [Hop94]. This approach consists of three phases and enables the additional modeling of surface edges. Therefore, for every point P_i in the point set P, a tangential plane is approximated for a small neighborhood (k-nearest points with k=15). The distance to the plane is again used as iso-value, comparable to the approach of Wyvill. But it is important to take care of the consistent orientation of the planes in order to ensure the right orientation of the produced polygons. The second phase optimizes the initial triangular mesh by deleting and inserting triangles and points. The goal is to reduce redundant and dispensable triangles, whereas the edges are kept. Therefore, the energy function in Eq. 2.3 is minimized. The term E_{dist} represents the squared distance between mesh and point cloud, and E_{rep} describes the ratio between the number of triangles and points respectively. The function E_{spring} assigns a spring force to each vertex and ensures a certain degree of smoothness between neighboring triangles [HDD+93].

$$E = E_{dist} + E_{rep} + E_{spring} \tag{2.3}$$



(a) initial unorganized point cloud. (b) phase 1: initial mesh. (c) phase 2: optimized mesh. (d) phase 3: smooth surface.

Fig. 2.12: Triangulation using the marching cubes technique proposed by Hoppe [Hop94].

Finally, in the third phase, the optimized mesh is subdivided to enhance local features. Therefore, the algorithm of Doo for the generation of *subdivision surfaces* is applied [Doo78]. The three phases are additionally illustrated in Figure 2.12.

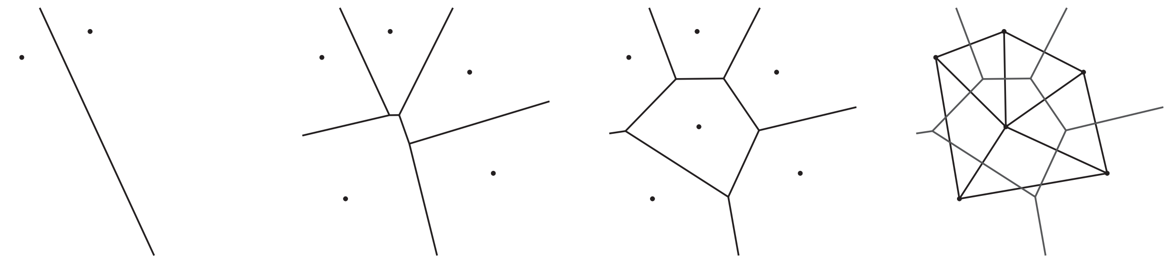
2.3.2 The Delaunay Triangulation

The *Delaunay triangulation* is an old mathematical principle introduced by Delaunay in 1924 [Del24]. He describes an incremental method to generate closed triangle meshes. The algorithm starts with an initial triangle. By iteratively inserting points, new edges and triangles are produced until no points are left. The Delaunay triangulation can be generated in two different ways: with the *Voronoi diagram*[7], and based on a circumcircle criterion.

For the construction of Voronoi diagrams a line between neighboring points is determined in a first step. In the second step, the perpendicular bisector of the side is computed. When adding further points to the system, new bisectors are computed and non-relevant existing parts are removed [Aur91]. Thus, for each point, a convex area arises. Regions without border edges were connected and represent the convex hull of the point set. Finally, points within neighboring regions can be connected without overlapping (see Fig. 2.13). This method requires a theoretical runtime of $O(n^2)$. The required time can be reduced to $O(\log(n))$ by using divide-and-conquer approaches that initially generated independent meshes that are merged in a second step. This approach is also suitable for parallel data processing [KKŽ05].

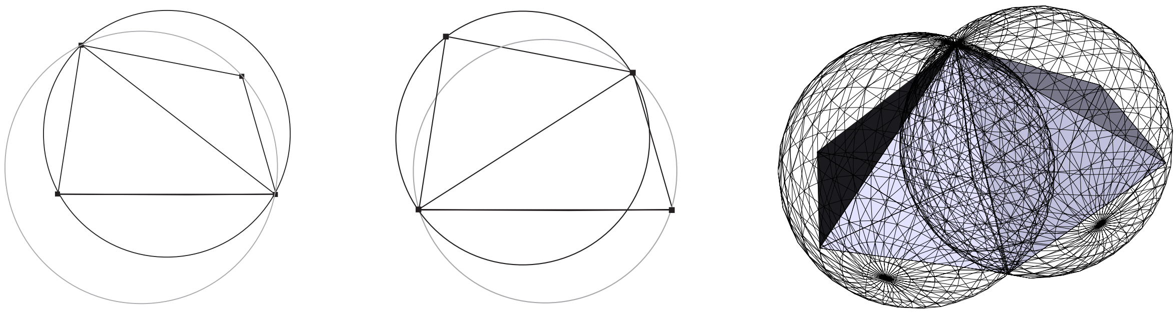
The second method is based on the circumcircle. Three points are initially connected to a triangle. A new inserted point is only connected with an edge if it lies outside the circumcircle of the triangle(s) (see Fig. 2.14). Otherwise, the edge shared by two triangles is swapped and the circumcircle criterion must be reviewed recursively. The Delaunay triangulation maximizes the minimum angle and is unique while no three points lie on a line.

7 Voronoi diagram: named after Georgy Fedoseevich Voronoi (04/28/1868 – 04/20/1908).



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

Fig. 2.13: Generation of the Voronoi regions (a-c) and the resulting Delaunay triangulation(d).



(a) Two triangles that do not satisfy the circumcircle criterion.

(b) Two triangles satisfying the circumcircle criterion.

(c) Two Tetrahedra that satisfy the circumsphere criterion.

Fig. 2.14: The circumcircle and circumsphere criteria for $\mathbb{R}^2$ and $\mathbb{R}^3$ respectively.

For the triangulation of 2.5D point clouds it is common to initially ignore the depth coordinate and compute a 2D mesh. By applying the depth coordinate to the 2D polygons, a spatial mesh is produced. It is obvious that this method is unsuitable for the generation of real 3D models, but it is often used for the visualization of terrains and (height) maps.

Considering the Delaunay criteria in $\mathbb{R}^3$, the circumcircle is replaced by a circumsphere and the points are not connected to triangles but to tetrahedrons [Aur91]. The circumsphere criterion defines that no further point besides the vertices of a tetrahedron is allowed to lie within the circumsphere (see Fig. 2.14(c)). As long as these four points do not lie in a plane, the 3D Delaunay triangulation (tetrahedrization) is unique. This technique produces the smallest convex hull of the 3D point set. However, a lot of intersection tests must be performed, which requires efficient data structures to manage the point cloud and the tetrahedrons. In the following steps, the triangle vertices are moved to achieve an optimal surface approximation. By removing triangle unions, originally existing holes can be restored.

The Power Crust

A triangulation method adapting the Delaunay technique is the *power crust* algorithm developed by Amenta et al. [ACK01]. The power crust is a construction process which takes a sample of points from the surface of a three-dimensional object and produces a surface mesh and an approximate medial axis. First, the approach approximates the medial axis transform (MAT) of the object. Then, an inverse transform produces the surface representation from the MAT. It comes with a guarantee that does not depend in any way on the quality of the input point sample. Any input gives an output surface which is the "watertight" (no holes) boundary of a three-dimensional polyhedral solid: the solid described by the approximate MAT. This unconditional guarantee makes the algorithm robust against noise and eliminates the polygonalization and hole-filling steps required in the previous surface reconstruction algorithms.

(a) (b) (c)

Fig. 2.15: Generation of polygon meshes with the power crust algorithm: laser range data, the reconstructed watertight polygonal model, and its simplification represented by the medial axis (from [ACK01]).

Besides the application of triangulations to construct polygonal meshes, the underlying geometric principles can be exploited for alternative visual descriptions of three-dimensional point clouds, e. g., for point set surfaces.

Point Set Surfaces

In contrast to triangulation and meshing approaches, Alexa et al. introduced *point set surfaces* [ABCO+03]. In their work, a set of 3D points in the proximity of the shape is mapped onto local polynomial surface approximations. Due to the local decomposition of Voronoi regions (see Sect. 2.3.2), the projection operator enables the computation of displacements from smoother to more detailed levels (see Fig. 2.16). The visualization is only based on the points themselves and their density.

The particular appeal of point sets is their generality: every shape can be represented by a set of points on its boundary, whereas the degree of accuracy typically only depends on the number of points. Point sets do not have a fixed continuity class or are limited to certain topologies as in most other surface representations. Polygonal meshes, in particular, have a piecewise linear C^0 geometry, often resulting in an unnatural appearance. Furthermore, this technique allows the selective refinement at significant and interesting surface positions.

This algorithm is particularly useful for very large polygonal models (e. g., 40 mio triangles), where single triangles would appear smaller than pixels on the screen.

Fig. 2.16: A point cloud representing an angel. The point density and precision changes in vertical direction [ABCO+01].

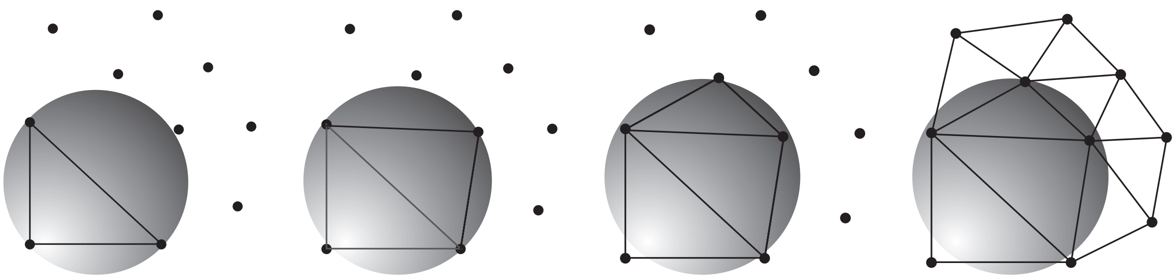
2.3.3 Advancing Front Meshing

Another family of polygonal mesh generation algorithms is the advancing front or moving front method. In these methods, the triangulation process starts with an initial front (e. g., one triangle), and elements are created on each triangle. This is achieved by creating new points in the interior of the front's domain. The current front always consists of the exposed faces in the domain. The front is advanced ("grows") either by establishing new points or by using existing points to create new elements (points/triangles). In this way, the complete domain is successively filled with triangles. Intersection checks are also required to ensure that triangles do not overlap as opposing fronts advance towards each other. A sizing function (e. g., a predefined distance value) can also be defined to control the size of newly added triangles. Lohner proposed to use a coarse Delaunay mesh of selected boundary nodes by which the sizing function could be quickly interpolated [LC00]. Furthermore, Schreiner et al. use a guidance field to determine the triangle size. This field contains information on edge lengths which helps to prevent large triangles from being created near small ones [SSFS06].

Scharf et al. present an approach that employs a deformable model to reconstruct a surface from a point cloud [SLS+06]. The model is based on an explicit mesh representation composed of multiple competing evolving fronts. These fronts adapt to the local feature size of the target shape in a coarse-to-fine manner. In principle, a flexible shape (e. g., a ball) grows and deforms in the "inside" of the point sets until all points have been added to the mesh. As a result, a "watertight" model (no holes) is produced.

Although the advancing front method is often used to create triangle meshes, it is a general meshing approach.

The Ball Pivoting Algorithm. Advancing front methods are especially suitable for large point sets. Since they only consider local neighborhoods, they are very fast. This principle is also used by the Ball-Pivoting Algorithm (BPA), which computes a triangle mesh interpolating a given 3D point cloud [BMR+99]. Typically, the points are surface samples acquired with multiple range scans of an object. Three points form a triangle, if a ball of a user-specified radius ρ touches them without containing any other point. Starting with a seed triangle, the ball pivots around an edge (i. e., it revolves around the edge while keeping in contact with the edge's endpoints) until it touches another point, forming another triangle. The process continues until all reachable edges have been included, and then starts from another seed triangle, until all points have been considered (see Fig. 2.17). The relatively small amount of memory required by the BPA, its time efficiency, and the quality of the results obtained compare favorably with the previously discussed techniques. The simplicity of this method also exhibits



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

Fig. 2.17: A sequence of ball-pivoting operations. From left to right: A seed triangle is found; pivoting around an edge of the current front adds a new triangle to the mesh; after a number of pivoting operations, the active front closes on itself; a final ball-pivoting completes the mesh.

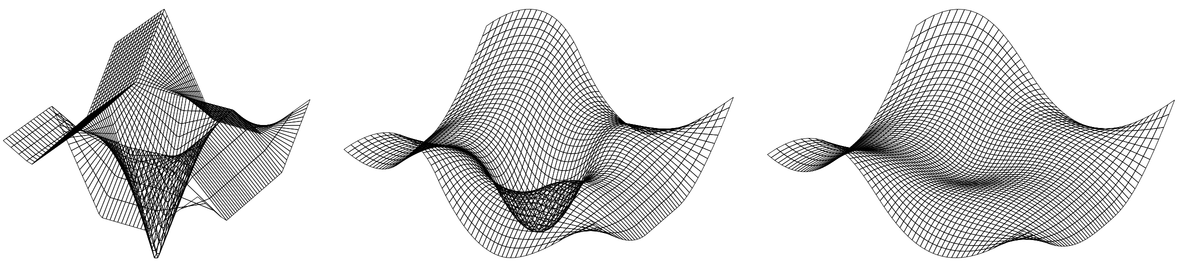
disadvantages. When the sampling density is too low, some of the edges will not be created, leaving holes. Thus, multiple runs may be required. When the curvature of the manifold is larger than $\frac{1}{\rho}$, some of the sample points will not be reached by the pivoting ball, and features will be missed.

2.3.4 Parametric and Analytical Surface Representations

B-Spline and Bézier Representations

Both naive variants of the Marching Cubes algorithm and the Delaunay approaches produce piecewise linear connected surface models, since they simply merge single triangles. Depending on the resolution and the number of triangles, the generated surfaces may look square-edged. Particularly on scanned surfaces, which contain noise and erroneous points, the triangulation can fail or represent the assumed topology only insufficiently.

A better surface approximation that guarantees smoother transitions between neighboring surface parts, are provided by *Bézier* and *B-Spline* surfaces. These are defined by piecewise polynomial functions and thus have an analytical description that enables the generation of additional points on the surface. The surface continuity is (user) defined by the degree of the control polygons (see Fig. 2.18). Furthermore, rational B-Splines enable the definition of weights for single vertices. Thus, erroneous and noisy vertices can be weighted differently from others. Due to the guaranteed continuity, the approximation of gaps is more sophisticated.



(a) (b) (c)

Fig. 2.18: Surface representation through B-Spline surfaces of different order: meshed linear (a), quadratic (b) and cubic surface (c).

Usually, existing polygonal meshes are used to approximate B-spline or Bézier surfaces. For example, Bajaj et al. replace triangle faces by weighted Bézier patches [BBX97]. Surface blending techniques and free-form surface fitting is also discussed by Xu et al. [XPB06]. Eck and Hoppe discuss the construction of a mesh of B-spline patches to represent the original surface [EH96]. Since not every shape may be representable by a single patch (e. g., self-intersections, multi-part objects), Park et al. present an enhancement [PYL99]. They introduce a technique that enables the generation of a network of rational B-spline patches.

In this thesis, B-spline curve and surface approximation are also used. Thus, more detailed discussions on the associated algorithms are given in Chapter 4 and 5.

2.3.5 Comparison of Meshing Approaches

There are many techniques to process point clouds in 3D. They basically differ in the method how the point set has been initially described and organized. Traditional methods are based on a cell space subdivision (octrees, Marching Cubes) or use Voronoi diagrams with the Delaunay triangulation as its dual. Further approaches focus on parametric descriptions (e. g., B-spline surfaces).

The Marching Cubes. This approach uses a set of polygon variants to approximate the entire surface based on a cell subdivision. The technique is particularly suited for the triangulation of iso-surfaces. By approximating the (unknown) surface through local functions, the algorithm may process point clouds as well. But this also requires the point cloud to represent a continuously smooth surface. The 3-phase approach of Hoppe processes arbitrary unstructured point clouds, but needs a lot of computation time [Hop94].

The Delaunay Triangulation. This method is mostly used, because it has mathematical guarantees. It computes the convex hull of an arbitrary point cloud, whereas the minimum angle of the polygons are maximized. Furthermore, its dual Voronoi diagram represents neighborhoods. Meshing algorithms use this technique to reconstruct non-convex surfaces piecewise. In many cases, the resulting triangle meshes are refined, interpolated or replaced by Bézier and B-spline patches.

Advancing Front methods. Instead of using a global parameterization, this class of algorithms creates triangle meshes iteratively. Therefore, an initial seed edge or triangle is advanced to more complex structures by adding vertices from the surrounding area. Thus, the mesh grows until all vertices have been reached. This approach is particularly suited for large point sets (i. e., $>$ 5 mio points). The literature discusses different methods which differ in the criteria that a new triangle is added. For example, the BPA ([BMR+99]) evolves a ball around an edge until a new vertex is touched and Schreiner et al. use a guidance field to locally control triangle sizes ([SSFS06]).

Bézier and B-spline surfaces. These representations approximate local polynomial surface functions with a certain mathematical degree and thus guarantee continuity and smoothness. By additionally applying weights, edges can be modeled as well. The analytical descriptiveness enables the computation of polygonal (quad) meshes with variable density. Considering the large point sets from 3D scanners, methods that produce networks of B-spline patches are dominating.

In fact, visualization and geometry evaluation (e. g., feature detection) are the main goals which are common for all approaches. Special emphasis is placed on a precise approximation (edge modeling) and an aesthetic geometry representation of the underlying point clouds.

2.4 Concluding Considerations

The surveyed triangulation and meshing techniques principally exhibit difficulties with modeling unevenly sampled and noisy data or sharp edges. This is due to the requirements of underlying smooth surfaces or special topologies (homeomorphic to spheres, discs, etc.). Therefore, approximations must be usually employed. Remedies may be found in new approaches that attempt to detect special geometry features, which are then passed to an optimized surface modeling. A varying point density is problematic, since most algorithms implicitly interpret this as a feature or expect a uniform density. Furthermore, holes and multi-part objects are also problematic and may lead to erroneous meshes, which must be repaired in additional processing steps [MK05]. The biggest problem are single significant erroneous points (outliers) which lead to significant distortions or defects.

The analysis of the measuring principles showed that there is already ancillary information about the generation of the point clouds. This includes the geometry of projected patterns or the coordinates of the tactile probes, which can be employed to efficiently optimize and correct the data in a preliminary step. Since these methods sample a surface based on defined movements and geometric principles, they are able to sort the data and store the additional information about the scanning and reconstruction process, the system configuration as well as the sensor types and the measuring principle. As a result, the point clouds are no longer unstructured and the post-processing becomes significantly more efficient.

Thus, the analysis of the acquisition procedure yields additional information about the scan process independently from point clouds or polygon meshes. The following algorithms (e. g., triangulation, geometry evaluation, segmentation) benefit from this through a prior optimization of the scan data based on the scan system specific characteristics or the measuring principle.

Chapter 3

Scan Data Acquisition

Many complex real-world objects cannot be easily reconstructed as a combination of primitive 3D objects or parametrized surfaces. Point clouds created from a variety of different 3D scanners are typically used for that purpose. Techniques for automatic data processing play an important role in a variety of applications. Unfortunately, the result is often disturbed by noise and artifacts, which complicate the data processing. Thus, the basis for the analysis, evaluation and correction of point clouds from 3D laser scanners is the data acquisition step. Besides the spatial coordinates, each internal procedure provides additional information about quality and structural relations for each measured point within a scan.

At the example of two 3D scanners, this chapter discusses the basic system setups and the point cloud generation procedures. Additionally, the sources and influences of errors are analyzed. In order to derive useful features and parameters from the acquisition stage, common internal data structures and relations used for further considerations are also introduced.

3.1 Employed Laser Scan Systems

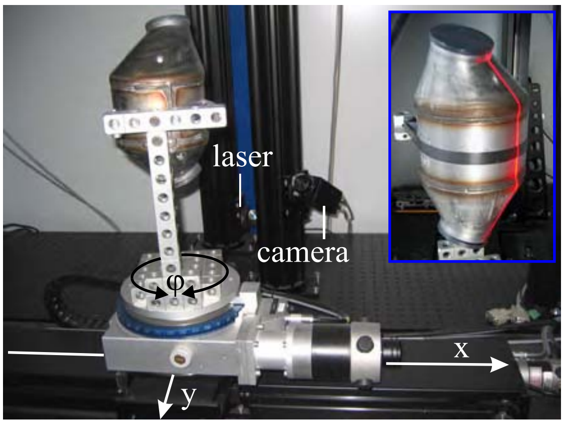
Point clouds generated by optical 3D measuring systems are generated and affected by many parameters in the acquisition process. On the basis of these parameters, analyses and evaluation procedures in the following processing stages become more robust. Particularly, correction and optimization algorithms can draw on reliable information derived from the measuring principle and system specific characteristics. Thus, the number of unknown parameters reduces, at the same time the adaptivity increases and finally, the degree of automation can be elevated.

This section introduces two 3D scanners utilizing the light section and triangulation principle as introduced in Section 2.1.2: a measuring machine and a flexible scanner. Both were used to generate the point clouds for which new algorithms are presented in this work. The identification of system-specific features for an optimized data processing is discussed at the example of these scanners.

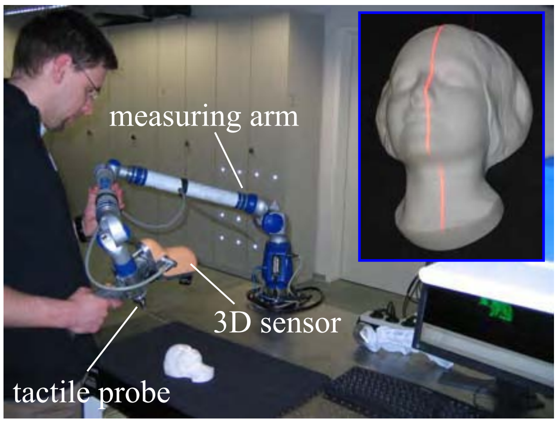
3.1.1 The 3D Measuring Machine

The first scanner is a complex, multi-axis system consisting of two light-section sensors, where each sensor contains a line laser and a digital gray level camera. The measuring machine allows three independent motions by two linear drives and one rotation stage. The machine was originally designed to capture the complex geometry of catalytic converters for industrial measuring tasks (see Fig. 3.1(a)). For this application, the resulting 3D point cloud is processed to detect and quantify geometrical measures, such as dimensions, radii and deviations from its nominal geometry. Due to the variability of the system it can also be used to scan a variety of other objects.

The measuring principle is optical triangulation based on light section (see Sect. 2.1.2). The system covers a measuring volume of approx. $400 \times 400 \times 400\,mm^3$. One of the sensors captures the upper surface parts and the other one captures the lower parts. Based on the high-precision locomotor system, the object is moved in front of the sensors until the entire surface has been captured. For a robust data acquisition process, all components are mounted on a granite plate to compensate vibrations from external sources in industrial environments. To ensure a unique projection of a laser to its connected camera, a trigger unit alternately generates impulses for each sensor. Within each point of time, the actual position of the motion axes is obtained and used to successively construct a 3D point cloud from single line scans of each sensor. The typical surface depending measuring uncertainty of this system is $80\,\mu$m.



(a)



(b)

Fig. 3.1: The employed 3D laser scanners. The measuring machine with its high-precision locomotor system (a) and the flexible scanner, consisting of a measuring arm and an optical 3D sensor (b).

3.1.2 The Flexible Laser Scanner

The second scanner is a flexible device combining optical metrology and approved tactile methods. While tactile methods are very precise, the pointwise sampling of large surfaces is extremely time-consuming. The combination of a conventional tactile device with a 3D laser scanner provides more flexibility. Thus, this scanner consists of a high-precision measuring arm with six degrees of freedom (7 axes) and an optical 3D sensor head.[1]

The optical sensor consists of a high-speed camera and a line projecting laser (see Fig. 3.1(b)). The contours of the laser lines, projected onto the object's surface, are captured at a preset clock speed. By using a CMOS sensor and integrated hardware-based image processing, up to 120 contour lines are digitized within one second, each with a maximum of 1.536 2D coordinates. The pixel positions of the profile line in the image are already calculated by the camera and subsequently transmitted to the connected computer in form of a list of 2D data. This reduces the quantity of data to be transmitted by over 95% compared to the standard procedure, where the whole image is transmitted. The measuring range is dimensioned to scan a line length of 70 mm in a maximum depth range of 40 mm. The position and orientation is given by the flexible kinematic system of the measuring arm. The sensor itself is integrated at the probe tip of the arm and can be moved within a hemisphere with a radius of 1.20 m, which finally results in an overall measuring uncertainty of approx. 100 µm.

Both introduced measuring systems and their data acquisition principles are representative for the ordered and unordered generation of 3D point clouds applying the laser light section method.

3.2 Sensor Calibration

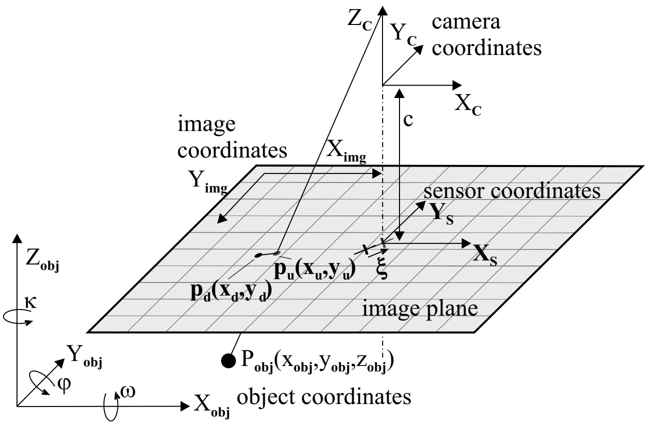
Before 3D coordinates can be generated by triangulation, the optical sensor components must be calibrated. This includes the determination of the mapping properties for the camera devices as well as the spatial position and orientation of cameras and laser planes to each other.

3.2.1 Camera Calibration

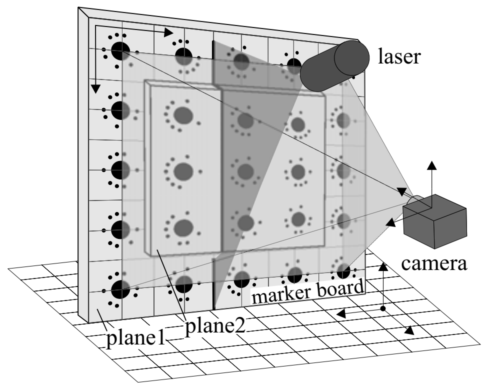
An exact 3D measurement depends on an exact camera calibration in order to extract metric information from 2D images. Therefore, the intrinsic and extrinsic parameters must be determined. The calibration problem is formulated as a functional Φ for the mapping from object coordinates into camera coordinates (see Fig. 3.2(a)). The functional Φ depends on the intrinsic parameters: camera constant (focal length) c, parameters of a nonlinear function for correcting the lens aberration a_i, and a two-dimensional vector ξ for the displacement of the principle point (see Fig. 3.2(a)). Additionally, the extrinsic parameters containing the spatial position and orientation of the camera to the object are included in this function:

$$\Phi_{obj} = \Phi_{obj}\left(\{x_{obj}, y_{obj}, z_{obj};\ c, \xi;\ a_1, \ldots, a_n\}; \{x_0, y_0, z_0;\ \omega, \varphi, \kappa\}\right) . \tag{3.1}$$

1 The measuring arm was built by FARO Technologies Inc. and the 3D scanner head was developed at the Fraunhofer IFF.



(a)



(b)

Fig. 3.2: Calibrating the 3D sensor. The sought intrinsic camera parameters and their coordinate systems are schemed in (a). The extrinsic camera position is obtained from a board of markers with known distances between them. In the same way, the position of the projected laser line is derived in two depth planes to compute the orientation of the laser plane relatively to the camera.

The computation of the parameters necessitates features from a calibration object with known dimensions. A feature is, for example, a set of coded markers on a plane as shown in Figure 3.2(b). The positions of the markers to each other have been previously determined by using a calibrated measuring device. On the basis of their positions in the captured image and the known real position information, the sought calibration parameters for Eq. 3.1 are determined.

Due to the lens aberration and the other initially unknown parameters, the rays from the coded marker to their pixel position on the camera chip do not intersect in a common point. Therefore, an adjustment procedure is applied. Its underlying mathematical principle is a bundle adjustment that estimates the intersection point by least-squares error minimization [GL92, TMHF00].

There are many different strategies that differ in the number of used images, the type of the calibration object (1D, 2D or 3D), and the used markers (lines, circles). Compared with classical techniques that use two or three orthogonal planes (see Fig. 3.2(b)), Zhang developed an efficient technique, which only requires the camera to observe a planar pattern at at least two different orientations [Zha99, Zha00]. Therefore, he used rectangular patterns to determine the projection parameters. Meng et al. adapted this procedure and used circular targets in combination with lines [MH03]. The advantage of their technique is that it needs to know neither any metric measurement on the model plane, nor the correspondences between points on the model plane. The parameter computation from ratios allows a flexible calibration and is additionally discussed by Wu et al. [WH06]. The disadvantage of these methods is that the scaling cannot be reconstructed, and thus a measured scale must be visible in the images [LRKH07].

For the camera calibration of both employed systems, the commercial software package CAP was used, which computes the parameters by a bundle block adjustment from multiple images.[2]

3.2.2 Laser Calibration

In a second step, the orientation of the laser plane to the camera is determined. Due to the previous camera calibration procedure, the spatial orientation and position of the camera and the transformation Φ_{obj} are known (see Eq. 3.1). As a result, a 3D coordinate can be determined for each point on a calibration plane (marker board). Thus, by extracting the 2D image positions generated by the laser line, its 3D representation can be computed (see Fig. 3.2(b)). By measuring in at least two different planes, multiple 3D lines in different depths are derived. Their common 3D plane is then determined by plane approximation [ARR99] (this approximation is discussed in more detail in Section 6.2.3).

Although the exact location of the laser origin is not needed for triangulation since it is based on the angular relationship to the plane, the location is useful to sort the points on the measured 3D contours by their projection angle. Because of the cylindrical lens, which splits the laser beam into a line (spatial light plane), the location of the light's origin is unknown. Therefore, an additional calibration step is applied, which uses a multi-step volume. This volume is scanned with a vertical laser alignment to artificially produce shadowing effects at the object's edges which causes disconnected line segments in different spatial planes as shown in Figure 3.3.

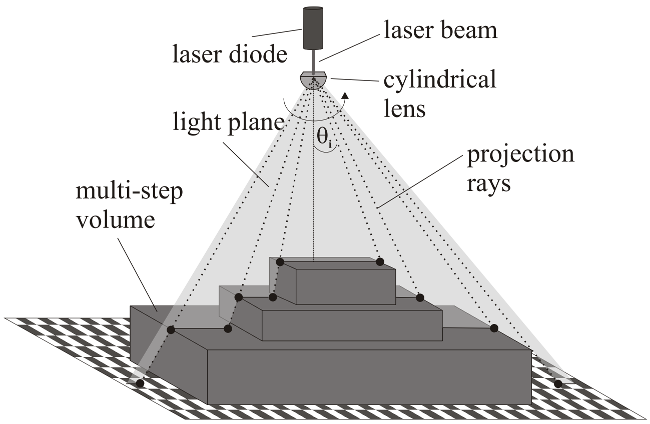


Fig. 3.3: The origin of the laser is obtained by scanning a multi-step volume (a) and utilizing the shadowing effects at the object edges. By intersecting the projection rays generated at these corners, the real laser origin is reconstructed.

The start and end points of these segments are manually segmented in the 2D image space. By applying the transformation Φ, their three-dimensional coordinates are obtained and finally used for the construction of inverse projection rays that point from

2 Combined Adjustment Program (CAP), developed by K^2-Photogrammetry.

the object edges to the laser origin. As a result, the common intersection point of these rays estimates the origin of the laser light. From the position and length of a segment, the beam width is obtained from its triangle relation. Because of the fixed alignment of camera and laser in a sensor, this procedure is only applied once.

Finally, the locomotor system is calibrated to the sensor by iteratively moving and measuring the markerboard. After each (defined) movement, the camera location is recalculated resulting in different spatial positions for the sought moving directions.

3.3 Point Cloud Generation

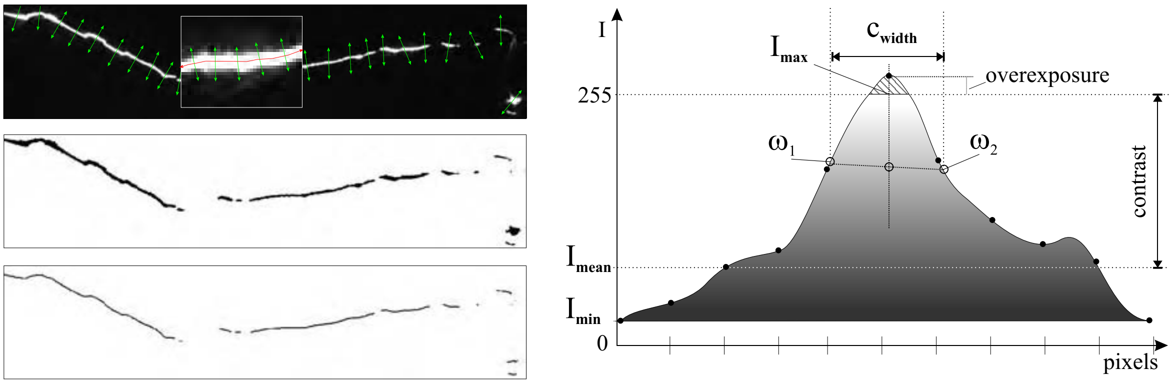
After having calibrated the digital cameras and having determined the spatial orientation of the laser plane, for each 2D pixel position the corresponding 3D coordinate can be determined by triangulation (see Sect. 2.1.1). Thus, for a 3D point cloud generation, the object is lighted and "intersected" by the light plane and the resulting contours, i. e., the corresponding set of 2D pixel positions, must be extracted. In order to reduce interferences by unfavorable illumination conditions, the scan is performed in a dark environment, where only the reflected laser light is visible. Therefore, an image processing pipeline is applied to separate this contour from the background.

3.3.1 2D Contour Extraction

Based on the principles of stereoscopic vision, the lines projected onto the object are deformed by the surface topology, and are thus enriched with depth information. The resulting contour is captured by a digital gray level camera and must be extracted, i. e., segmented from the background (see Fig. 3.4(a)(top)). Depending on the resolution of the camera devices and the width of the laser line, which is determined by its lens, the captured contour has a width of several pixels. But only the center is needed for the 3D reconstruction.

The medial axis of a plane object in the continuous case is defined as the set of points which are equidistant from at least two points on the object boundary [Par97]. In the discrete space, this definition cannot be directly applied, because the discretization generally produces jagged structures resulting in irrelevant skeleton branches [SPSP02]. For example, there is no unique discrete medial axis of line with an even pixel width. Thus, an approximation, called "thinning", is usually applied. By performing morphological operations on binary images, the foreground pixels are eroded until only a one-pixel wide contour is left. The algorithms of Zhang and Stentiford are typically used for that purpose [ZS84, SM83]. An approach for gray-level images is proposed in [MAM04]. The result of these procedures is an image with only medial points in the fixed-point arithmetic. But the more precise the segmentation, the more accurate is the projection from 2D pixel positions into 3D coordinates by triangulation. Therefore, the following alternative technique is proposed.

Due to diffuse reflections on the surface, there is a characteristic distribution of the intensities perpendicular to the contour. The gray level intensities at the edge of the contour are typically smaller than those at the central positions. Therefore, the path is analyzed perpendicular to its contour, from which different profiles are derived (see Fig. 3.4(a) (top)). The profile center is determined with subpixel accuracy by linear interpolation between the inflection points (see Fig. 3.4(b)).



(a) (b)

Fig. 3.4: Extracting the laser illuminated contour from the background (a). Based on a fixed threshold, the input image (top) is binarized (middle) and finally thinned (bottom). The gray level function of an extracted profile is shown in (b). The central position of a profile is found by detecting the inflection points ω_i based on the Laplacian edge detector.

In order to obtain the central (medial) positions for the entire contour, the following algorithm is applied: calculate binary image I' from input image I with threshold t (for efficiency a static $t = 50$ is defined); compute the medial axis points; apply the profile analysis; extract the subpixel contour centers (see Alg. 3.1).

In practice, there is often no unique maximum intensity value due to overmodulations and overexposures of the camera chip. Thus, it is necessary to derive the maximum value from the surrounding pixels by approximation. Therefore, the inflection points ω_i of the gray level intensity function for a profile are determined. This is achieved by previously applying an isotropic 3×3 Laplacian edge detector (see Fig. 3.5) from which the positions of the zero crossing of the 2nd derivatives (maximums of the first derivatives) are obtained [Par97]. The subpixel position between them is assumed to be the center of the profile and thus the center of the contour at this image position (see Fig. 3.4(b)).

Experience shows that the maximum intensity value is an important criterion for a balanced measurement system. The maximum should exist within the upper 10% of the gray level spectrum. This is achieved by adjusting either the intensity of the laser or the shutter time of the camera. Although the subixel interpolation for the maximum is valueable at at least a width of two pixels, the contour should be as thin as possible.

A quality measure is given by the contrast, defined as the variance around the mean value I_{mean}. For this application it is approximated as the difference $I_{max} - I_{mean}$ (see

Input: gray level image I containing the observed laser line and threshold t
Output: 2D subpixel positions of the contour line centers
calculate Binarization(I,t,B)
calculate Skeletonization(B,S)
calculate Laplace(I,L)
foreach *pixel position* p_i *in* S **do**
  if p_i *is foreground pixel* **then**
    calculate nextNeighbor(p_i,n_i)
    define profile direction $\vec{d_p}$ perpendicular to $\overrightarrow{n_i - p_i}$
    calculate Mean(I,d_p)
    foreach *pixel position* r_i *on the profile* **do**
      if $I(r_i) \geq I_{mean}$ **then**
        calculate FirstZeroCrossing (L,$p(\omega_1)$)
        calculate FirstSecondCrossing(L,$p(\omega_2)$)
        calculate centralPosition($p(\omega_1)$,$p(\omega_2)$)
        break;
      end
    end
    set p_i to background
  end
end

Algorithm 3.1: The scanline extraction algorithm.

Fig. 3.4(b)). Obviously, the higher the contrast, the more the line differentiates from the background. Another criteria is the slope at the inflection points. The higher the contrast and the thinner the contour, the steeper the slope.

Problematic cases appear at roughly textured surfaces such as stone or cast iron. Multiple intensity maximums may arise in such cases and can lead to severe artifacts and outliers in the resulting 3D points. The sources and influences of errors on a 3D point cloud during its 2D acquisition are discussed in the following.

3.3.2 Influences of Errors

There are many sources for erroneous scan data during the acquisition process. The most severe cases will be discussed below. As can be deduced from numerous example

0	1	0
1	-4	1
0	1	0

(a)

1	1	1
1	-8	1
1	1	1

(b)

Fig. 3.5: Laplacian image convolution masks for 4-neighborhood (a) and 8-neighborhood (b).

scans, most of the outliers and other erroneous points are caused by unwanted reflections. In these cases, the high energy laser beam is reflected from mirroring surfaces such as metal or glass. Therefore, too much light hits the sensor of the camera and so-called *blooming effects* occur.

In other cases, reflections may also miss the camera. At sharp edges only partial reflections appear. In addition, craggy surfaces cause multiple reflections and therefore indefinite point correlations. Furthermore, a part of the object may lie in the path of the projected laser line to the camera causing *shadowing effects*.

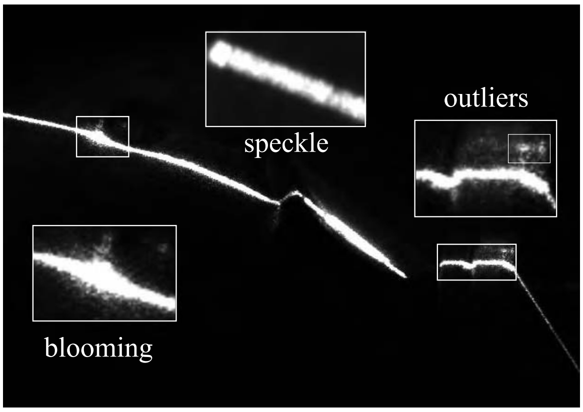


Fig. 3.6: Optical effects that affect the 2D data acquisition. Speckle effects cause spotted, fringy contour lines and small gaps. Blurring and blooming broaden the laser line in the image space and complicate the exact detection of the line center. Unwanted reflections on mirroring surfaces cause outliers.

Since the scan systems are typically used in industrial environments, some atmospheric effects (e. g., dust and vibrations) may affect the quality of the image obtained by the camera. Furthermore, aliasing effects in the 2D image processing and *speckle effects* lead to high frequent noise in the generated 3D data. In summary, the following effects already disturb a point cloud during the acquisition:

Shadowing Effects. These effects appear on object surfaces with a complex geometry. In this case, holes and salient parts prevent the camera from observing the laser line. Finally, this results in an incomplete point cloud leaving several “holes”.

Aliasing Effects. Aliasing effects emerge in the image processing step that detects and analyzes the contour. Due to the mapping of the continuous laser line to discrete pixel raster in the digital image, aliasing effects may occur. These effects lead to high frequency noise in the resulting 3D point cloud. The negative influence of this effect is reduced by the profile evaluation and the subpixel interpolation (recall Sect. 3.3.1 and Fig. 3.4).

Blooming Effects. Blooming is caused by the interaction of laser intensity and surface reflectivity. If the laser light is too intensive or if too much reflected light hits the sensor of a CCD camera, some cells in the chip cannot hold the generated energy which is then distributed to neighboring cells.

Blurring. Blurring is caused by an improper alignment of camera, laser and object. Because the depth of sharpness of the camera and laser line is limited by their lenses, the surface to be captured must be within this range. Blurring can also be a result of the blooming effect.

Speckle. These effects are caused by the coherence of the laser light. Due to the fixed phase relation, laser radiation is subject to interference whenever it is scattered from objects with diameters in the same order of magnitude as the wavelength [PS97]. Since almost every surface contains small-scale structures or dust particles, surfaces illuminated with laser light show speckled structures, which move with the direction of observation. Thus, these speckles are randomly distributed points in space, where both constructive and destructive interference take place (see Fig. 3.6).

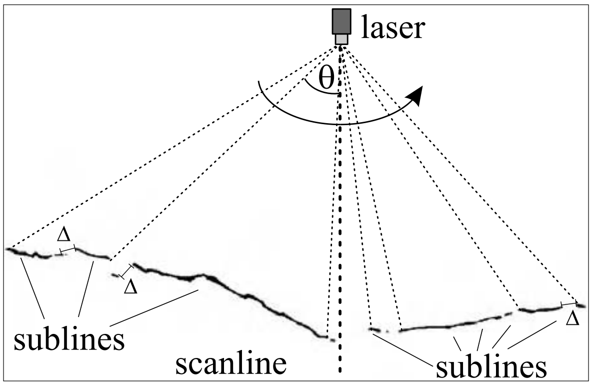
The combination of these effects is responsible for noise, outliers, gaps and holes, and thus the resulting 3D point data is partially erroneous. Closing larger gaps in the 2D image by interpolation is not reasonable, because even if the distance between two 2D pixel positions remains constant, their 3D representation may not. However, a lot of errors can be minimized by an optimal alignment of the sensor system and the object surface so that the number of unwanted reflections is as low as possible.

Curless et al. proposed a method which observes the degree of reflection over time [CL95]. Their technique allows to reduce errors from blooming and speckle effects by applying a space-time analysis. Due to the time-consuming acquisition of multiple images for one line it is not suitable for time-critical applications.

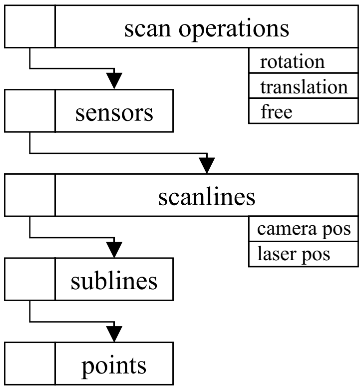
3.4 3D Point Cloud Structure

Based on the discussions before, we can assume that a laser scanned point cloud always has a structure. It consists of single contour line-based measurements projected into the 3D space by triangulation. In the following, one resulting 3D contour is called *scanline*. Due to shadowing and speckle effects, a scanline may be interrupted. Thus, the emerging 3D line segments are named *sublines* of a scanline. Finally, each subline contains a number of 3D points representing the scanned object's surface.

The existence of multiple 3D sensors necessitates an additional structuring by the sensors that have generated the scanlines. By moving a sensor or the object, multiple scanlines are generated and combined into a common point cloud. Therefore, each movement is named *scan operation*. The measuring machine allows defined successive translations and rotations and the flexible scanner for free movements. For each of these scan operations and for each contained scanline of a sensor, the actual camera and laser positions are known from the calibration stage. They can be used as an additional information in further algorithms and are stored for each scanline. Figure 3.7(b) illustrates the data structure representing these relations.



(a)



(b)

Fig. 3.7: Illustration of the scan data structure as derived from the measuring principle and the employed 3D scanners (a). Based on the laser origin, the points are sorted by their projection angle. The whole contour is named scanline, which is divided into a sorted set of sublines, if the distance Δ_i is larger than a threshold Δ_{max}. The hierarchical data structure derived from the measuring principle and the employed 3D measuring systems is shown in (b).

Depending on the preliminary image processing step that detects and collects the contour points, the sublines and their points are unsorted. For further processing and data analyses, it is desirable to sort the points along the real contour shape. Therefore, the information within the point data alone is not sufficient, since holes and gaps interrupt the connectivity. The solution for this problem is implicitly given by the laser position, because from its viewpoint the point set represents a continuous straight line. Due to the fact that all data lie within a plane, the point sorting can be performed by comparing the corresponding projection angles (see Fig. 3.7(a)). After this procedure, the points are sorted and the order of the sublines is corrected based on their start and end points. Since this procedure reflects the real measuring process, it is always correct.

The splitting procedure, defining the number of sublines, uses a manual threshold Δ_{max}. If the distance between the start and the end point of two successive sublines of a scanline is smaller than Δ_{max}, both sublines are merged. Thus, this parameter allows to control gaps within a scanline. For the used scanners the sampling density on a scanline (average Euclidean distance between two neighboring points) is 100–300 μm and Δ_{max} is defined as 1 mm. The sequence of scanlines within each operation is defined by the order of their acquisition. Due to the defined movements of the measuring machine, successive scanlines are topological neighbors. For the flexible laser scanner, there is no defined relationship between successively acquired scanlines.

In this context, the term *system information* is introduced. It describes the knowledge about:

- the type of the 3D scanner, i. e., its configuration and measuring principle,
- the type of movement between single (line) scans,
- the position and orientation of cameras, lasers and light planes,
- and finally the order of scanlines and sublines.

3.5 Noise Analysis

The combination of different external influences causes noise. For industrial measuring tasks it is of particular importance to reliably estimate the distribution of the noise on a scanline to be able to develop algorithms that can guarantee a certain accuracy. In order to apply local data evaluations and quantifications, the noise should be free of any systematic error. Usually, the measuring data is assumed to be normally distributed. Therefore, the deviations of the measured points are analyzed by using the χ^2 goodness-of-fit test in order to check the existence of a normal distribution.

3.5.1 Test Setup

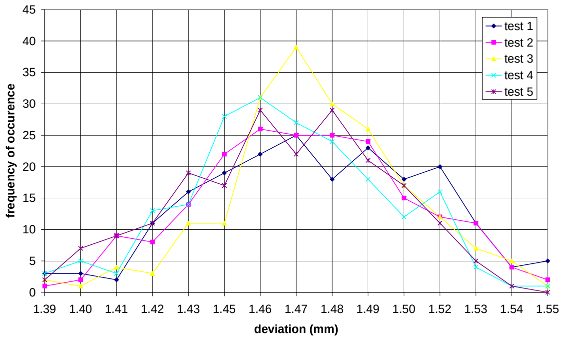
For the investigation of the noise distribution, a nominal object serves as an ideal reference. Its position and dimensions must have been determined by a system that exhibits a higher accuracy as the employed 3D laser scanners. Therefore, one planar face of a calibrated cuboid was used. In a first step, the planarity of this face has been checked by an external tactile coordinate measuring machine with an uncertainty of $2\,\mu$m.

In the second step, the face was sampled by 100 3D points from the tactile probe tip of the flexible measuring arm, which guarantees a global measuring uncertainty of $30\,\mu$m, while the uncertainty of the laser scanner is assumed to be about $100\,\mu$m. Then, a plane was approximated into these data ($\sigma = 0.049\,$mm) that finally serves as the wanted nominal geometry to which the measurements of the laser scanner are compared.

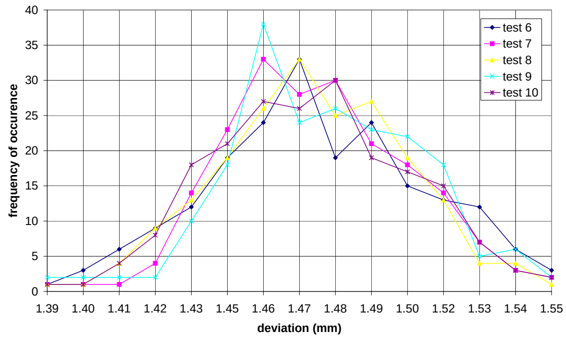
In the last step, the optical sensor, also mounted on the measuring arm, was used to generate 5000 laser scanned 3D points on the same face of the cuboid. Since the coordinate systems of both scan devices are the same, the orthogonal distances of the laser scanned points to the tactile sampled nominal plane give information about the distribution of the noise. To minimize the influence of systematic errors from the calibration between the tactile probe and the laser scanner, only local samples were taken. Finally, the resulting data set contains a nominal reference plane and a set of different sample points from the laser scanner.

3.5.2 Test Results

In order to analyze the distribution, the distances of the sample points to the reference plane are computed. Their observed frequencies are sorted and divided into $k = 15$ intervals (bins). Then, the number of points is counted for each interval k_i. The analysis was performed at 10 different locations on the reference plane, each with $n = 200$ points. The result is illustrated in Figure 3.8.



(a)



(b)

Fig. 3.8: Histograms indicating the distribution of the deviations from the nominal plane for 10 independent test measurements. The evaluation results are split into the two histograms shown in (a) and (b).

A test to approve the occurrence of a normal distribution is then performed by the χ^2-test. The assumed Gaussian normal distribution is given by the probability density function:

$$\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{3.2}$$

with two parameters for the mean (μ) and the variance (σ^2) of the measured distances $x_1, \ldots, x_n$. The test is then performed in the following steps:

1. Due to the unknown distribution, μ and σ must be estimated by their empirical measures, denoted as the empirical mean $\bar{x}$ and the empirical variance $(\Delta x)^2$ with:
$$\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j, \qquad (\Delta x)^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_j - \bar{x})^2 \,.$$

2. The measurements are divided into k intervals ($a_1 < a_2 < \cdots < a_k$) and the frequency of occurrence m_r for each $a_r \leq x_j \leq a_{r+1}$ is counted. Then, the probability p_r for the intervals is determined with:
$$p_r = \Phi\left(\frac{a_{r+1} - \bar{x}}{\Delta x}\right) - \Phi\left(\frac{a_r - \bar{x}}{\Delta x}\right)$$
with
$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_0^z e^{-\frac{1}{2}t^2} dt \,. \tag{3.3}$$

3. The test value, which characterizes the distribution c^2 is computed as a sum over the observations m_r and the expected values np_r by:

$$c^2 = \sum_{r=1}^{k} \frac{(m_r - np_r)^2}{np_r} \,. \tag{3.4}$$

4. Finally, the probability of error α is defined ($\alpha = 0.05$) and the corresponding value for χ^2_α is computed for $m = k - s - 1$ degrees of freedom, where s is the number of parameters of the assumed distribution. In this case (μ, σ), $s = 2$ and $m = 12$.

If the relation $c^2 \leq \chi^2_\alpha$ is true, it can be assumed that X is normally distributed with a probability of error of α. The results are shown in Table 3.1. Besides test number 9, all test values were smaller than the allowed value of $\chi^2_{0.05}(12) = 21.0$.

In summary, the test for a normal distribution of the scanline noise was successful with an error probability of 5%. The normal distribution has its maximum at the point $x = \mu\ (\bar{x})$, which also means that the measures become more centered around the mean value, the smaller the variance σ^2. This observation is of particular importance for many algorithms in this work that rely on the assumption of locally normal distributed noise.

Based on the normal distribution, the last step is the determination of the confidence interval $x^-_\alpha \leq x \leq x^+_\alpha$ with the bounds $x^-_\alpha = \mu - \sigma z_\alpha$ and $x^+_\alpha = \mu + \sigma z_\alpha$. The value z_α is obtained from the value $t_{\alpha,\infty}$ of the Students t-distribution ($z_{0.05} = 2.0$, for further reading see [PTVF02]).

test	1	2	3	4	5	6	7	8	9	10
min	-0.10	-0.10	-0.11	-0.09	-0.09	-0.10	-0.12	-0.12	-0.11	-0.11
max	0.03	0.04	0.04	0.04	0.05	0.03	0.05	0.04	0.04	0.03
$\bar{x}$	-0.03	-0.03	-0.03	-0.03	-0.02	-0.03	-0.03	-0.03	-0.03	-0.03
$(\Delta x)^2$	0.03	0.03	0.03	0.02	0.03	0.03	0.03	0.03	0.02	0.02
x^-_α	-0.08	-0.08	-0.08	-0.07	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08
x^+_α	0.03	0.03	0.02	0.02	0.03	0.02	0.03	0.02	0.02	0.02
c^2	8.49	14.49	15.66	19.61	10.03	7.53	6.76	7.53	21.27	5.81
$c^2 \leq \chi^2_\alpha$	√	√	√	√	√	√	√	√	○	√

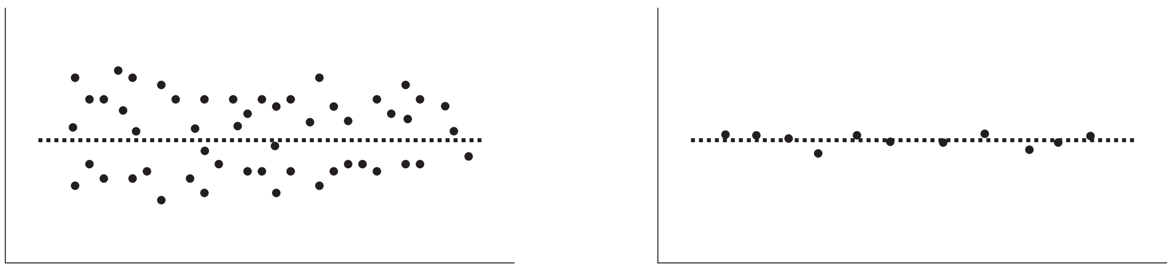
Tab. 3.1: Results of the performed noise analysis for 10 tests (in mm). Each test contained $n = 200$ sample points, whose distances to the reference planes were partitioned into $k = 15$ intervals. The probability of error was defined as 5% ($\alpha = 0.05$). The values x^-_α and x^+_α indicate the bounds of the confidence interval. The χ^2-test is successful, if the test value c^2 is smaller than $\chi^2_{0.05}(12) = 21.0$.

Due to the very local analysis, only the noise distribution is evaluated. Systematic errors from calibration and displacements of the kinematic system are not included and the uncertainty may vary on different surfaces. Due to the same measuring principle, the noise distribution is assumed to be equal for the flexible and the complex measuring device.

3.6 Accuracy vs. Resolution

When considering uncertainties in (optical) metrology, an often occurring discussion regards the terms *accuracy* and *resolution*. Both terms are often used to describe the quality of measurements from different points of view.

Usually, the term *accuracy* is not used in metrology, but the term *uncertainty*, since an uncertainty can be measured as a deviation from a known nominal value. The variance σ^2 and the standard deviation σ are measures that are typically used for this purpose. The term *resolution* usually means the density of the measured points with regard to the captured area. The sampling theorem plays an important role when defining the resolution, i. e., a function must be sampled with a frequency twice as high as the original frequency in order to allow for an exact reconstruction. Figure 3.9 illustrates the relation between accuracy and resolution.

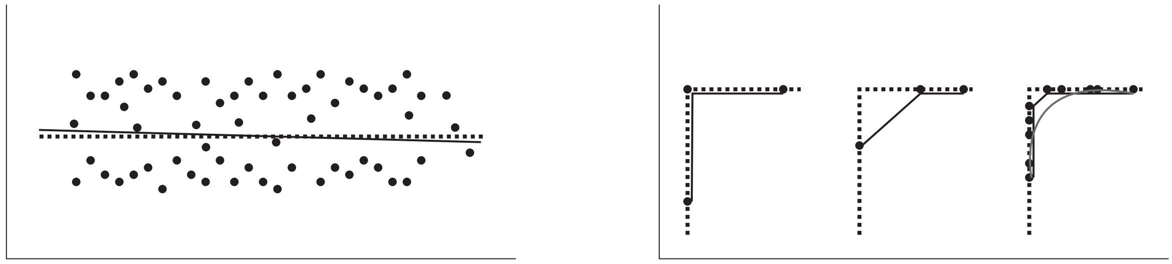


(a) (b)

Fig. 3.9: The relation between accuracy and resolution: high variance (low single point accuracy) and high sampling resolution (a). A high single point accuracy (low variance), but a low sampling resolution (b).

In fact, the lower the uncertainty, the higher is the single point quality. But single points are usually not used to measure quantities, since they can be affected by noise or they possibly represent outliers. Therefore, a set of neighboring measures is approximated by a geometry of known function (e. g., line, plane, etc.). The approximation minimizes the influence of single points affected by noise. In this case, the term *feature-based uncertainty* is often used. But therefore, a “sufficient” high point density or sampling resolution must be available to approximate a representative geometry, whereas “sufficient” depends on the geometry and the approximation procedure (typically 200–300 μm). Figure 3.10 illustrates the relation between point density and point distribution.

The point distribution also plays an important role, because even if the uncertainty is low and the density is high, an unfavorable distribution may disturb the resulting approximation (see Fig. 3.10(b)). A problematic case for all measuring systems is the capturing of sharp edges. Such edges contain all possible frequencies, and thus cannot be directly measured and reconstructed. Therefore, geometries in the near environment must be approximated by a known function, and a common intersection must be computed to quantify the properties of a sharp edge.



(a) (b)

Fig. 3.10: High variance and low single point accuracy but reconstructed line has a low feature-based uncertainty (a). A low density and unfavorable point distribution results in a poor approximation, even if the single point accuracy is high (b).

3.7 Summary

In this chapter, the two laser scanners employed for the point cloud generation were introduced. On the one hand, a measuring machine with defined motions between several scans was presented. On the other hand, a 3D sensor, mounted on a flexible kinematic measuring arm, was used. Since both devices employ the laser light-section and triangulation principle to produce 3D point clouds, the calibration procedures to obtain the interior parameters of the camera and the exterior to the laser were also discussed. Both systems are representative for the variety of light-section systems and principles.

Based on these discussions, system and measuring principle specific characteristics were extracted, which are useful for a further point cloud processing. This includes the scan data structure with its scanlines and sublines as well as additional information about the laser and camera position. Furthermore, the influence of different error sources was discussed, which enables to perform error-specific optimizations in the following chapters.

The majority of errors is brought into the data during the 2D image acquisition processing step caused by optical interferences. Furthermore, errors from the calibration of camera and laser and their orientation to each other affect the data. Besides outliers, it could be proved that contained noise is normally distributed and shows no additional systematics. The proof was performed by applying the χ^2-goodness-of-fit test. Finally, the last section discussed the often recurring relations between accuracy/uncertainty and resolution/density.

Due to the fact that the point cloud consists of a set of successively acquired scanlines, this structure is applied for a more efficient data processing in the following chapter.

Chapter 4

Curve-based Scan Data Processing

When reconstructing objects from laser scan data, usually very large data sets have to be processed. Therefore, it is often necessary to minimize the number of points while minimizing the loss of information at the same time. In addition, the generated point cloud usually contains a considerable number of errors. Most of these errors are directly dependent on the measurement system and the scanned object's surface. Outliers and other erroneous points are an important factor when discussing metering precision. Therefore, they have to be detected and removed from the point cloud or corrected in order to get a clean model that can be used as precise measuring data.

A scanning system usually provides more information than the point cloud alone. This additional information includes the camera and laser (projector) positions and their parameters as well as movements between several scan operations and comparable information. Furthermore, the applied measuring principle yields information about the data structure and other acquisition specific information. In contrast to usually existing methods, which assume a point cloud as an unstructured data set, this chapter exploits this additional system information and introduces alternative approaches for an optimized processing of point clouds from optical scanning systems.

Most optical scanners use the triangulation principle based on laser lines, fringe projection or photogrammetry. The result is a point cloud, which consists of sets of scanned lines. For photogrammetric approaches and similar measuring techniques, the grid structure of the captured images is a natural parameterization from which scanlines can be extracted. Thereby, each scanline is an individually captured measurement, depending on the corresponding projection and viewing conditions and the topology of the object surface. This structure is used to employ new techniques to efficiently handle optical scan data. Thus, the analysis of the entire point cloud considers the measurements separately before a global analysis is performed.

The major advantage of the proposed procedures is that not possibly erroneous meshes are considered and processed, but instead the measured point cloud is directly used. Compared to many algorithms that manipulate point clouds through an approximation with polygonal meshes, the goal of the algorithms here is to automatically correct each measurement individually and integrate the methods directly into the measurement process. This basic approach was introduced by Teutsch [Teu03]. This chapter seizes and extends this idea to solve problems in automation and industrial environments (e. g., dust, vibrations or changing lighting conditions) which require a high degree of robustness. Therefore, compromises must be found between fast and robust algorithms that meet the specific requirements of an application. Based on the extraction of the point cloud structure, these approaches and techniques are used to reconstruct a revised and optimized point cloud that is much better qualified for fast, robust, and precise measurements. Finally, the effectiveness of the proposed methods is evaluated based on exemplary point cloud sets from various models.

4.1 Scanline Approximation

After having discussed the point cloud structures, this section focuses on a general approach for the data approximation in order to analyze and correct single scanlines. Measurements and methods of visualization do not require hundreds of thousands or even millions of points and large data sets need a lot of computation time and memory. Precise measurements, on the other hand, need data that is free of noise that may be caused by external effects during a scan. Therefore, we aim at an analytical scanline description, which allows to analyze the measurements, to minimize the point number, and to clean the point clouds from artifacts. On the basis of the measuring principle and the system parameters, this section describes fast and effective methods to process scanlines derived from a 3D scanner.

The resulting point cloud consists of a number of individual measurements (scanlines). Therefore, it is reasonable to describe the whole point cloud as a set of sorted lines or curves to obtain an analytical description. Due to high frequent noise, artifacts and outliers, interpolation would not result in an adequate representation, since errors still remain. In contrast, an approximation allows a controlled smoothing and, in particular, an adaptive weighting of single measurements, e. g., depending on their quality.

The basic problem for a scanline approximation is the choice of the analytical description. One possibility is the interpolation by a global polynomial function and another one is the approximation by piecewise polynomials functions using B-splines. Basically, a single polynomial function of a certain degree n and a set of parameters p_i is defined by:

$$f(x) = \sum_{i=0}^{n} p_i x^i \,. \tag{4.1}$$

In general, n different points can be described by a unique polynomial of order n (degree $n-1$). Typical methods for polynomial interpolation are given by Lagrange and Neville [PTVF02]. A fast approximation is achieved by distance minimization with the least-squares method. Because of the dense data sets, the unique polynomial exhibits a corresponding high degree. With respect to computation time and numerical stability, the calculation of such a polynomial in $\mathbb{R}^3$ is not reasonable. Furthermore, each single point has the same influence on the resulting function. Thus, this kind of approximation is not robust enough against outliers. After all, functions in the form of $y = f(x)$ or $z = f(x, y)$ are depending on the axes of the coordinate systems, and thus are not flexible. Therefore, it is reasonable to describe curves in $\mathbb{R}^3$ with parametric functions in the form of $\vec{r}(t) = [x(t), y(t), z(t)]$. To minimize the influence of single erroneous points, the whole curve should consist of local, single connected curves with low degree. The sum of the conditions is satisfied by B-spline curves. Hence, for our purpose each subline of one discontinuous measured scanline is processed and approximated by a B-spline curve.

4.1.1 B-Spline Curves

B-splines are used to obtain an analytical description of a point set from which features (e. g., edges, curvatures) can be easily extracted. In addition, they can be used to close small gaps between neighboring sublines. Interpolating these gaps keeps the precision of measurements, if the distance between the corresponding sublines is less than a certain threshold (e. g., 2 mm). This threshold depends on the accepted inaccuracy of the closing segment. Otherwise, the interpolated B-spline segment would follow the assumed geometry insufficiently.

A B-spline curve with the position vector $P(t)$ of order k is defined over an ordered knot vector T as vectorial polynomial:

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t) \qquad t_{min} \le t < t_{max}, \qquad 2 \le k \le n+1\,, \tag{4.2}$$

with the control points B_i and the normalized basis functions $N_{i,k}(t)$, which are defined by the Cox-de Boor recursion formulas [dB72]:

$$N_{i,1}(t) = \begin{cases} 1 & \text{if } x_i \le t < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

$$\text{and} \qquad N_{i,k}(t) = \frac{(t - x_i)N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}}\,. \tag{4.4}$$

The curve $P(t)$ is a polynomial of degree $k-1$ on each interval $x_i \le t < x_{i+1}$ and offers a C^{k-2} continuity at the segment transitions. It has a smoothing character, since it exhibits the variation-diminishing property, thus, it does not oscillate about any straight line more often than its control polygon oscillates about the line [Rog01]. Finally, $P(t)$ lies within the convex hull of its control polygon B (see Fig. 4.1(a)). This property is important, because it allows the direct curve approximation using the dense measured data as control points. The introduced approximation error must be smaller than the noise. This assumption is valid as long as the point density of the measured data is lower than the contained error, which is typically true for optical metrology.

To ensure that the B-spline curve approximates the given points in an optimal way (B-spline fitting), new control points have to be generated from the measured points.

Therefore, the distance between the points on the curve P_i and the measured points M_i has to be minimized. A well-known and fast method to achieve this is to minimize the quadratic (Euclidean) distance:

$$\sum_{i=1}^{n} \|P_i - M_i\|^2 = min\,. \tag{4.5}$$

A point on the B-spline curve is computed using Eq. 4.2. This equation is expressed in matrix formulation:

$$\mathbf{P} = \mathbf{N} \cdot \mathbf{B}' \tag{4.6}$$

$\mathbf{P}$	–	in every row a curve point, in $\mathbb{R}^3$ 3 columns (x,y,z)
$\mathbf{N}$	–	for every point P_i one row with the basis functions from Eq. 4.4
$\mathbf{B}'$	–	in every row a new control point, in $\mathbb{R}^3$ 3 columns (x,y,z).

Since the control points should be manipulated, the equation must be rearranged:

$$\mathbf{B}' = \mathbf{N}^{-1} \cdot \mathbf{P}\,. \tag{4.7}$$

The distance between the points P and M should be zero:

$$\mathbf{P} - \mathbf{M} = 0, \quad \mathbf{P} = \mathbf{M}\,. \tag{4.8}$$

Finally, the calculation rule for determining new control points from the given measured data is given by:

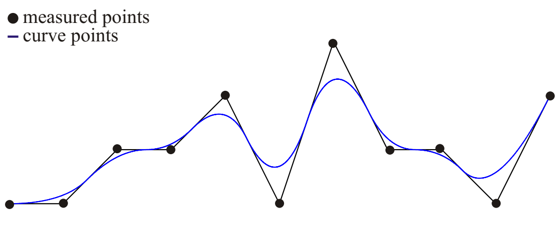
$$\mathbf{B}' = \mathbf{N}^{-1} \cdot \mathbf{M}\,. \tag{4.9}$$

This rule is only valid, if as much control points as measured points are generated. Usually, less control points are wanted. Thus, N is not square and the computation of its inverse is not possible. This problem is solved by computing the pseudoinverse, which is square. The new control points $\mathbf{B}'$ are then given by Eq. 4.10. By applying the $\mathbf{B}'$ to Eq. 4.2, the distance of the resulting B-spline points to the measured points $\mathbf{M}$ is minimized.

$$\mathbf{B}' = ((\mathbf{N}^T \cdot \mathbf{N})^{-1} \cdot \mathbf{N}^T) \cdot \mathbf{M} \tag{4.10}$$
$$((\mathbf{N}^T \cdot \mathbf{N})^{-1} \cdot \mathbf{N}^T)\text{- pseudoinverse of matrix N}\,.$$

This approach is called the least-squares method [PTVF02]. For solving minimization problems, so called l-norms are used. Since the described method considers the squared distance, it represents the l^2-norm. In many cases, a minimization problem can be reduced to a linear equation system by using the least-squares method. An approach to solve this problem by using different distance norms as a linear programming problem is presented in [HBL97].

When generating new control points to re-sample the original data, the sampling theorem must be considered. Therefore, the number of control points must not be greater than the half of the number of input points (+2 to force the curve to the endpoints of the control polygon). On the one hand, the corresponding result in Figure 4.1(b) shows that the distance is minimized, but on the other hand that the curve does not optimally follow the given original scanline data (e. g., overshooting, oscillation). Furthermore, the matrix-based minimization is time-consuming and the new curve is allowed to leave the convex hull of the original scanline data, which could significantly worsen the approximation. Therefore, the standard minimization procedure is not applicable for scanline approximation.



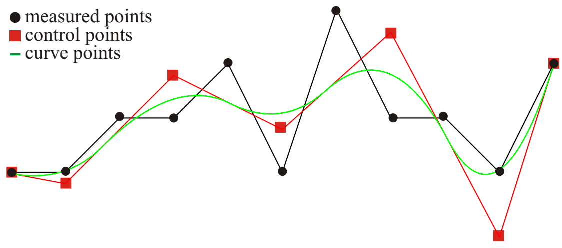


Fig. 4.1: B-Spline approximation strategies. Taking the measured points as input, the resulting curve approximates them (a). By computing new control points, the curve nearly interpolates the measures.

B-splines can be forced to pass through the given points by adjusting their knot vectors. Therefore, the knot value, which corresponds to a given point is used k times, but this results in a C^0-continuous connection at the segment transitions. This procedure is only applied to the first and last point in order to keep the principle subline structure.

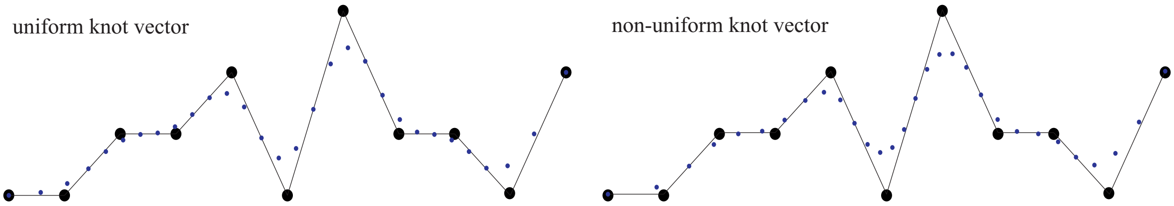
A related class of curve reconstruction approaches uses the Moving Least-Squares (MLS) algorithm. These methods locally fit curve functions while minimizing the distance to the measured data or to previously extracted curve features (e. g., edges). This is typically achieved by computing a function-specific weight for every given point [Lee00]. Problems may occur at the transition between successive curve segments, i. e., if a certain degree of continuity is required. In principle, an MLS reconstruction of scanlines with other functions is also possible, but it was found that the B-spline approximation is much more insensitive to high frequent noise.

4.1.2 Optimizing B-spline Representations

For further processing, it is desirable to generate regularly spaced points. Therefore, the values of the parameter t of the B-spline curve have to be adjusted. This can be achieved by choosing a knot vector with parameter distances that are proportional to those of the control points (chord distances) using the following ratio:

$$\frac{t_{i+1} - t_i}{t_{i+2} - t_{i+1}} = \frac{\|d_{i+1} - d_i\|}{\|d_{i+2} - d_{i+1}\|} \,. \tag{4.11}$$

The effects of uniform and non-uniform parameterization are compared in Figure 4.2. The resulting points on the curve are distributed more regularly and represent the geometry much better. Especially larger point distances and curve segments are sampled more precisely.



(a) Irregular point distribution of a B-spline with uniform knot vector.

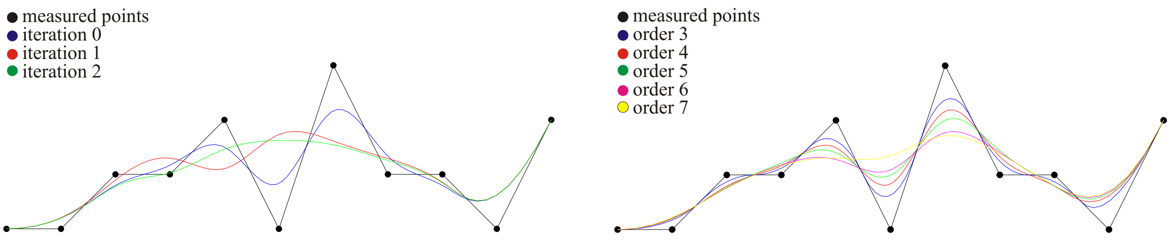
(b) More regular point distribution of a B-spline with non-uniform knot vector.

Fig. 4.2: B-Spline approximation with different knot vectors resulting in different point distributions on the curves.

This representation is also useful to vary the point density on the approximated curve. For example, by calculating 10 regularly spaced points on a curve with an arc length of 10 mm, a point density of 1 mm is achieved. For performance reasons, the real arc length s (see Eq. 4.12) is estimated as the sum of all distances between neighboring points.

$$s = \int |r'(t)| dt = \int \sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2} dt \tag{4.12}$$

Due to the limited spatial resolution in the 2D image acquisition step, aliasing effects appear and result in high frequent noise within the 3D points. Therefore, the data has to be smoothed. Smoothing a B-spline curve can be performed in different ways. The simplest one is to increase the curve order, and, thus to enlarge the segments intervals. This results in a much higher computation time and the smoothing behavior cannot be controlled (see Fig. 4.3(b)). Furthermore, the degree at specific control points can be elevated by adjusting the knot vector.



(a) (b)

Fig. 4.3: Smoothing a B-spline in two different ways. The iterative smoothing procedure is illustrated in (a) compared to the successive elevation of the B-spline order in (b).

Another approach is the manipulation of the control points. They can be moved or even removed to achieve a smoother result. Since the removal of measured data is not desirable, an iterative B-spline smoothing is applied. This method is based on minimizing the bending energy of a thin, elastic bar with a constant cross-section as proposed in [Had98]. The corresponding term for a B-spline $P(t)$ is as follows:

$$E = \int (\ddot{P}(t))^2 \quad dt = min\,. \tag{4.13}$$

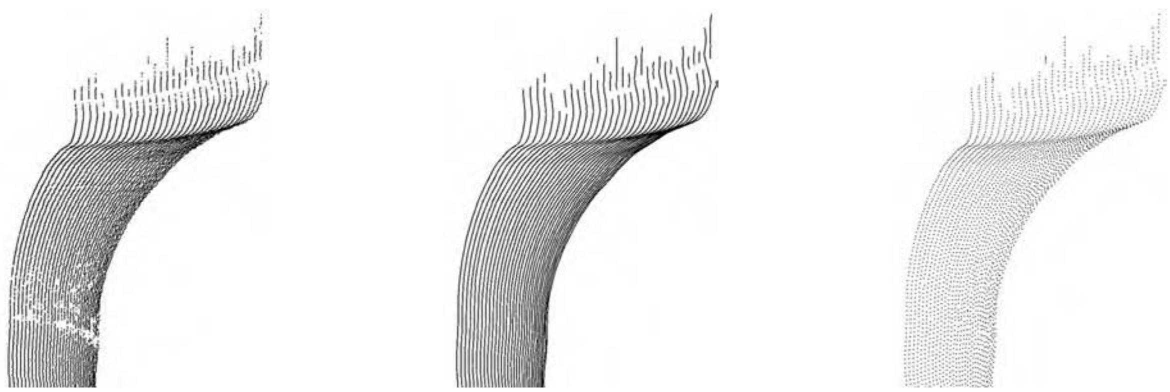
Each considered control point $\tilde{b}_i$ is manipulated depending on the control points $\bar{b}_i$ from the last iteration (see Fig. 4.3(a)):

$$\tilde{b}_i = -\frac{1}{16}\bar{b}_{i-3} + \frac{9}{16}\bar{b}_{i-1} + \frac{9}{16}\bar{b}_{i+1} - \frac{1}{16}\bar{b}_{i+3}\,. \tag{4.14}$$

Furthermore, metrology requires that a smoothing operation can be limited to a certain assumed uncertainty. Therefore, a tolerance δ is added to control the movement of control points with respect to the metering precision. Within each iteration the tolerance is checked with Eq. 4.15. This guarantees that the new coordinates are not moved more than δ units.

$$\tilde{b}_r^* = \begin{cases} \tilde{b}_i, & \text{if } \|b_i - \tilde{b}_i\| \leq \delta \\ b_i + \delta \cdot \frac{\tilde{b}_i - b_i}{\|\tilde{b}_i - b_i\|}, & \text{if } \|b_i - \tilde{b}_i\| > \delta\,. \end{cases} \tag{4.15}$$

The results of smoothing and thinning an exemplary point cloud using the proposed procedures are illustrated in Figure 4.4. Furthermore, Eck and Hadenfeld propose



(a) (b) (c)

Fig. 4.4: Steps for smoothing and thinning using proposed methods. The originally measured points are shown in (a) with their closed and smoothed B-spline approximation with the same number of points (b) and the final point set with reduced point number (c).

an alternative method to remove control points of a B-spline curve [EH95]. Their algorithm implicitly detects redundant control points, which are not needed for the reconstruction. Furthermore, they minimize the error when adding or removing control points in general. Schumaker and Stanley present another method for a shape-preserving knot removal with respect to a given tolerance [SS96]. This method is efficient, but limited to quadratic splines.

4.2 Approximation Error

Since the B-spline curves approximate into the given sublines, deviations between the curves and the measured points occur. For a reliable data evaluation based on these curve approximation, this deviation must be estimated. Therefore, the minimal distance of a point on a measured subline to the B-spline curve must be determined. Due to the parametric curve definition, the minimal distance cannot be directly derived. The problem is solved by either adaptively refining the running curve parameter or by generating significantly more points on the curve than existing input points.

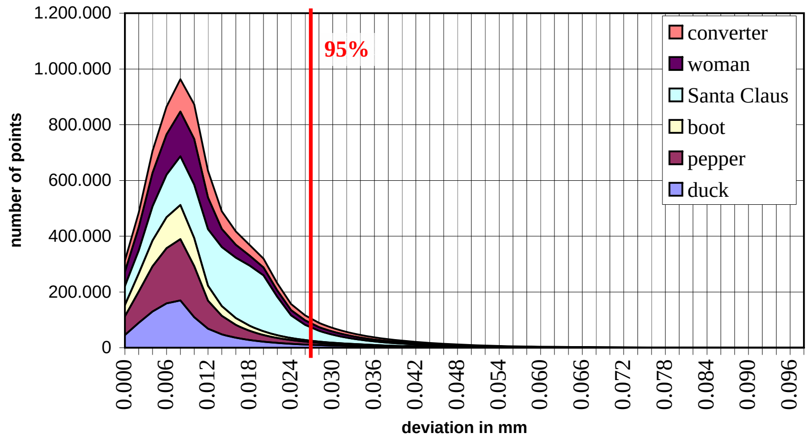


Fig. 4.5: Histogram showing the distribution of the deviations. For all models, most deviations (95%) are smaller than 30 μm. Particularly due the definition of the b-spline knot vectors, which force the curve through the start and end points of a subline, many zero deviations occur. The number of larger deviations ("outliers") is negligibly small.

For this analysis, the latter approach was chosen with a number of curve points 10 times higher than measured points, which is sufficient to estimate the deviation. A higher point density would reduce the distance even more. Due to the smoothing character of B-splines, the test was performed at different objects that contain planar regions as well as curved regions with edges. The results are visualized in Figure 4.5 and given in Table 4.1. For the used models, the introduced error by curve approximation is smaller than the assumed measuring uncertainty of the scanners. The number of deviations larger than this uncertainty ranges between 2–9 ‰, except the converter model which contains a lot of outlying points that cause higher deviations. But even in this case ($< 0.6\%$ outliers), the approximation is acceptable.

The amount of the deviation depends on the point distribution on a subline and the B-spline definition. The higher the curve order, the smoother is its path and the possibly

model	duck	pepper	boot	Santa Claus	woman	converter
curves	10.871	18.715	9.670	26.746	13.272	14.010
points	998.092	1.297.044	724.657	2.281.961	1.155.575	965.983
< 0.01 mm	4.51 %	5.22 %	5.62 %	2.96 %	4.47 %	4.05 %
< 0.02 mm	70.37 %	72.75 %	73.44 %	34.70 %	62.55 %	52.92 %
< 0.05 mm	99.57 %	99.74 %	99.58 %	99.49 %	98.96 %	97.59 %
> 0.10 mm	0.02 %	0.05 %	0.09 %	0.05 %	0.06 %	0.59 %
$\bar{x}$(mm)	0.011	0.010	0.012	0.015	0.011	0.014
σ(mm)	0.008	0.010	0.007	0.009	0.010	0.011

Tab. 4.1: Deviations for the B-spline approximations of sublines. In most cases the deviation of 99% of all points is smaller than $50\,\mu m$. The avarage displacement is always $\approx 10\,\mu m$, while the assumed total measuring uncertainty of the scan system is $\approx 100\,\mu m$. The test was automatically performed for ≈93.000 scanlines. The models are illustrated in Figure 4.11 and Figure 5.5.

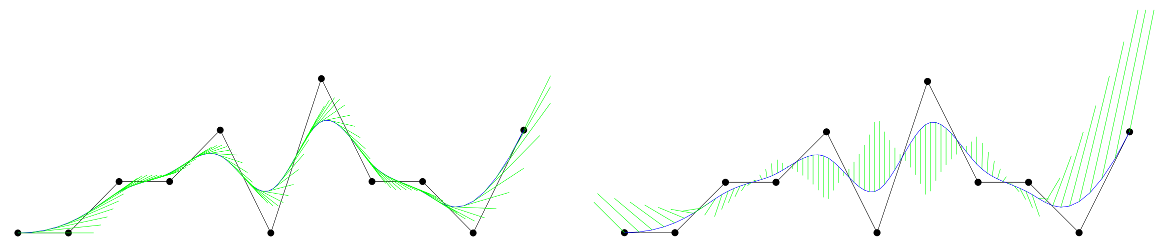
higher are the deviations. Since the curve is determined by the measured points, their distribution is also of particular importance. A dense point set can be approximated much smoother than a coarse sampled part. This discussion refers to the problem of accuracy and resolution previously discussed in Section 3.6. If the sampling rate and the point density are high enough to represent the surface (sampling theorem), and if the variance of the contained noise is smaller than the density, no additional error is introduced by the approximated curve. But concrete values depend on the application, the surface, the noise level and the curve parameterization. Actually, points on a subline have an average distance of about 200 μm and the normal distributed noise has a variance of 20–30 μm (see Table 3.1).

Due to the B-spline definition that requires the curve to stay within its control polygon, noise and single occurring peaks are smoothed. The approximated curves run within the normal distributed noise, and thus do not introduce a noticeable error. The quality of the approximation with regard to the (unknown) original cannot be evaluated, but indeed the approximation does not worsen the given data. For the analyzed data sets the observed error is negligible, since single point features with sizes smaller than about $50\,\mu m$ are not evaluated anyway.

4.3 Scanline Analysis

A prior analysis step is required for the correction of scanline defects, a controlled smoothing and the determination of local geometric properties, such as edges and curvature. For this purpose, derivatives are employed to identify changes in the function with respect to one of its variables. For parametric curves $r(t)$ in $\mathbb{R}^3$ the derivative $r'(t)$ describes a vector containing the changes in the three dimensions.

The required B-spline derivatives are obtained by formally differentiating the basis functions N of Eq. 4.2. Specifically:



(a) (b)

Fig. 4.6: Computing the directions of the first and second derivatives of a B-spline curve in order to analyze the local behavior of the curve shape. The end positions of the illustrated curve are forced to pass through the given points by adjusting the knot vector. They are singular points and cannot be derived.

$$P'(t) = \sum_{i=1}^{n+1} B_i N'_{i,k}(t) \quad \text{and} \quad P''(t) = \sum_{i=1}^{n+1} B_i N''_{i,k}(t) \tag{4.16}$$

with the first derivative ($N'_{i,1}(t) = 0$):

$$N'_{i,k}(t) = \frac{N_{i,k-1}(t) + (t - x_i)N'_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N'_{i+1,k-1}(t) - N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}} \tag{4.17}$$

and the second derivative ($N''_{i,1}(t) = 0$):

$$N''_{i,k}(t) = \frac{2N'_{i,k-1}(t) + (t - x_i)N''_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N''_{i+1,k-1}(t) - 2N'_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}} . \tag{4.18}$$

The second derivative provides information about the smoothness (tangential behavior) of the curve's shape, due to its relation to the curvature κ and the normal vector N, i. e., $r'' = \kappa N$. For example, the effectiveness of the iterative optimization and smoothing procedures in Section 4.1.2 are analyzed in Figure 4.7. In this case, 100 iterations with a tolerance δ of 50 μm were applied to the measured scanline.

4.3.1 Analyzing Curvature Patterns

The most important property in differential geometry is the curvature κ. It describes the local properties and relations between the first and second derivative, and thus, the precise curve shape. Most of the erroneous sublines exhibit conspicuous curvature patterns. For example, within short distances lots of turnarounds and sharp edges occur, indicated by high curvatures. In general, the curvature of parameterized curves is given by:

$$\kappa = \frac{\|r'(t) \times r''(t)\|}{\|r'(t)\|^3} . \tag{4.19}$$

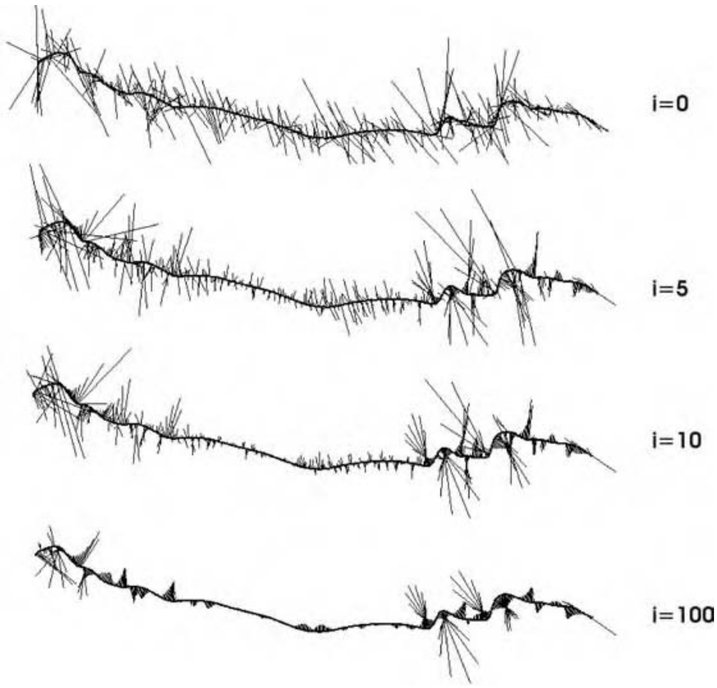


Fig. 4.7: Evaluating the length and direction of the second derivatives of the approximated B-Spline curves within the iterative smoothing process. The curve's shape is basically preserved and high-frequent noise is significantly reduced.

If the curve is parametrized over its arc length s instead of the arbitrary parameter t, the curvature is directly given by the length of the second derivative:

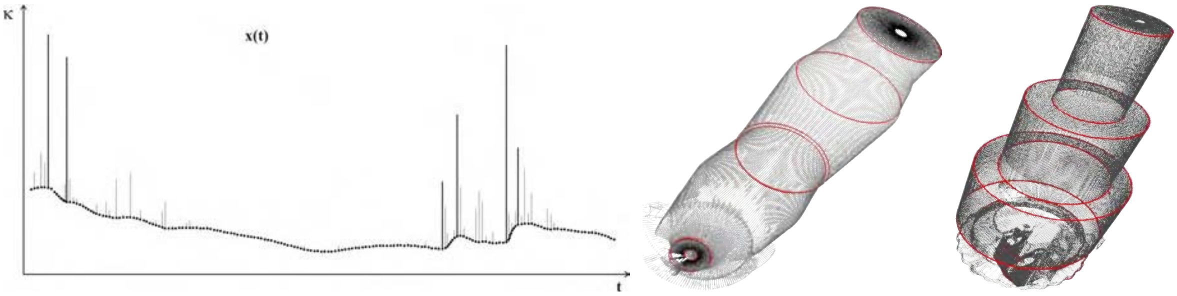
$$\kappa = |r''(s_0)| \,. \tag{4.20}$$

The direct computation of the curvature with these equations is the big advantage of parameterized curves against their polygonal representations [PT95].

The amount of scanline curvature with respect to surface edges (subjective impression) was empirically analyzed using the models illustrated in Figure 4.11 and Figure 5.5. In this study it was found out that the value $\kappa > 0.2$ indicates sharp edges reliably (for 4^{th} order B-Splines). These values can also be used to detect potentially erroneous sublines, because it was observed that sublines with at least 40% of their points having a curvature $\kappa > 0.2$ can be considered as errors and can be deleted. The mentioned parameters are stable in a range of $\pm\,10\%$, depending on the surface properties, noise, etc. (see [TIT+05]).

Sharp edges partially describe the shape of an object. Thus, in addition to using this information for error detection, it can also be exploited for automatic feature detection (see Fig. 4.8(b)). Further algorithms benefit from this curvature information, e. g., for an improved polygonal mesh construction at the surface edges.

However, the curvature can also be used to control the interpolated geometry. While B-splines smooth sharp edges, curvature can be used to control the curve shape at the edges. In addition, the weights from an evaluation of the acquired 2D and 3D data can



(a) Curvatures on a B-spline curve. (b) Derived object parts. (c) Derived object parts.

Fig. 4.8: Feature detection based on significant curvatures ($\kappa > 0.2$ in black lines) (a); identification of object structures using curvature (e. g., circles (b), circular parts (c), ellipses); $\kappa > 0.2$ in red (b).

be used to manipulate the path of the curve depending on the quality of each vertex in the image and corresponding coordinate in the point cloud. Therefore, rational curves to weight single points, are applied in Section 4.4.

4.3.2 Quality Evaluation

The first step for optimizing a point cloud is to optimize the projection and the viewing conditions. Therefore, the quality of the point cloud has to be quantified with respect to the position of laser and camera. Improving the recording conditions leads to less erroneous 3D points. Afterwards, the remaining errors can be detected and evaluated much easier.

A straightforward approach for minimizing the number of points and correcting the point cloud is to test for the focus space. All parts of a scanned point cloud that do not lie within this area should not be used, and thus they can be deleted. In addition, a simple test is used for sublines that consist of only a few points. During the tests it was found out, that sublines that had no neighbors and consisted of less than 10 points could safely be assumed to be errors.

3D Data Quality

Because the scan procedure uses optical sensors, the quality directly depends on the viewing and projection properties. The smaller the angle between surface normal and direction of projection or viewing, α_p and α_c respectively, the better the surface was seen. In addition, the triangulation angle (on the surface) between projection vector $\vec{p}$ and the camera viewing vector $\vec{c}$ is optimal when the angle $(\alpha_p + \alpha_c)$ defined by them is $\frac{\pi}{2}$. With this constraint, there are less intersecting errors (e. g., convergent rays) and less numerical errors when computing the location of the surface point. The more $(\alpha_p + \alpha_c)$ deviates from $\frac{\pi}{2}$ and the larger the angles are, the worse are the viewing conditions of the surface point for camera and/or laser. The employed laser scanner consists of two optical sensors. An upper and lower sensor ensure that the entire surface is captured. The quality of a measured point is evaluated depending on its sensor properties (see Fig. 4.9).

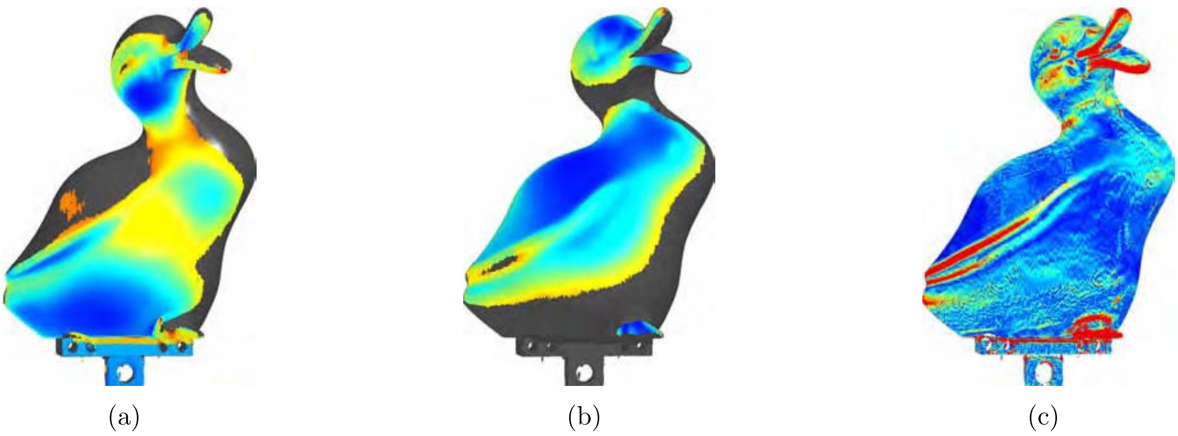


Fig. 4.9: Quality evaluation at the example of point clouds showing: the quality depending on projection/viewing angle for the lower (a) and the upper (b) sensor (0=blue to $\frac{\pi}{2}$=red) and curvature with highlighted significant edges ($\kappa > 0.2$) derived from the B-spline curve approximation per single scanline (c) (blue: $\kappa = 0$ to red: $\kappa > 0.2$).

For these quality analyses, the surface normals have to be known. In general, they are derived from polygonal meshes. However, for our approach, a polygonal mesh is not needed. A plane is approximated (see Sect. 6.2.3) over a defined neighborhood for each point (within $\pm$ one scanline) and its normal is used as the normal for the considered point. The neighborhood is determined very fast, since the sorting of the scanlines is a known system parameter. To guarantee a consistent normal orientation, the camera position is used as well. Obviously, the surface normal has to point towards the camera that recorded this point. Therefore, the correct surface normal orientation can be computed by evaluating the angle between normal vector $\vec{n}$ and viewing vector $\vec{c}$. The normal is oriented correctly if this is $\leq \frac{\pi}{2}$, otherwise $\vec{n}$ is reversed.

Hence, an important step to optimize such a laser scan system is to perform an initial scan and evaluate the result based on the position and alignment of the projection system. Depending on the result, the viewing conditions of the object can be optimized. In addition, for each point a corresponding quality for the viewing condition can be computed and used for further error compensation. Besides the quality in 3D, each surface point can additionally be evaluated by quantifying its quality in the 2D image processing. The intermediary results are numbers, scaled between 0 and 1. The limits in 3D are implicitly given by the worst and best possible projection. In 2D the limits are defined by empiric and statistic tests (e. g., contrast).

2D Quality Evaluation

In addition to the previously discussed viewing conditions, there are factors in the image processing stage that may influence the resulting point cloud. These factors are influenced by environmental effects such as lighting, laser light energy, surface type, and camera resolution. For example, a high contrast of a laser line in the recorded image results in a high stability and precise detection. In addition, the line thickness is an important parameter (see Fig. 3.4(b)). Thick lines make it harder to find its exact middle and small features may be buried. Therefore, the line profile at each vertex has to be considered in the point quantification as well. The higher the slope of an edge orthogonal to the line, the better its middle can be detected as a maximum in the profile (recall Sect. 3.3.1).

Additional Quality Estimation

In the analyses it was found out that erroneous sublines have specific characteristics: on the one hand, they are very short with many high curvature points. High curvatures can be detected by analyzing the approximated curves. On the other hand, sublines may be caused by reflections. These lines are projected somewhere in space and, therefore, have no neighboring scanlines. These aspects are used for quantification as well.

4.4 Reconstruction using NURBS Curves

The application of Non-Uniform Rational B-Spline (NURBS) curves is similar to that of normal B-splines. In addition, for each control point B_i, a weight h_i is specified (see Eq. 4.21). Therefore, the basis functions must become rational functions $R_{i,k}$ and the knot vector is non-uniform. Versprille was the first to discuss rational B-splines [Ver75].

The weight determines the influence of the control point on the curve's path. The higher it is, the more the curve converges against the control point. For the scanline approach, it is derived from each point's quality that has been determined in the sections before (see Fig. 4.10). Additional descriptions may be used as well [PT95, Rog01].

$$P(t) = \frac{\sum_{i=1}^{n+1} B_i h_i N_{i,k}(t)}{\sum_{i=1}^{n+1} h_i N_{i,k}(t)} = \sum_{i=1}^{n+1} B_i R_{i,k}(t) \tag{4.21}$$

A rational B-spline curve also exhibits the variation-diminishing property, and for all $h_i > 0$ the curve lies within the union of convex hulls formed by k successive control polygon vertices. Each rational basis function is positive or zero for all parameter values, i. e., $R_{i,k} \geq 0$. The sum of the rational basis functions for any parameter value t is one, i. e., :

$$\sum_{i=1}^{n+1} B_i R_{i,k}(t) = 1\,. \tag{4.22}$$

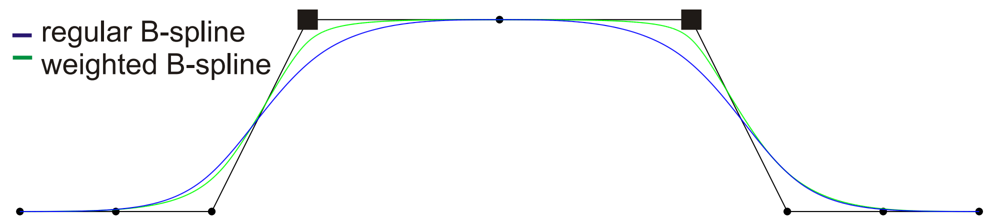
To reconstruct an optimized scanline, and thus an optimized point cloud, the following rules to weight the control points depending on the curvature properties and the normalized quality values from the prior quality quantification are proposed:

$$W = \begin{cases} 1 & \text{for all points not rateable} \\ 1 + w_c + w_l + 0.5w_t + 1 & \text{for } \kappa > 0.2 \\ 1 + w_c + w_l + 0.5w_t + \kappa & \text{otherwise} \end{cases} \tag{4.23}$$

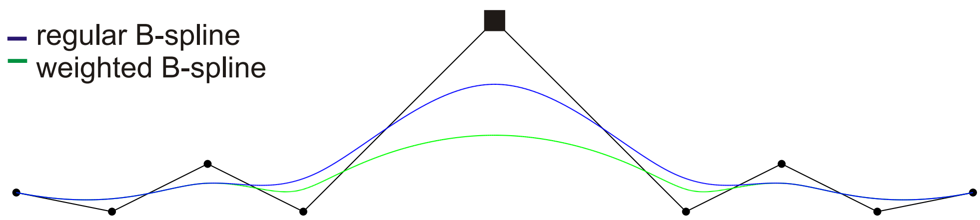
w_c, w_l - weights for camera viewing angle and laser projection angle.
w_t - weight for triangulation angle (lower due to its relationship to w_l and w_c).

For example, the maximum viewing and projection angles for the normalization are $\frac{\pi}{2}$, which result in the lowest possible weight. Basically, there is no maximum weight for a control point, because the curve computation is based on the ratios between the weights.

The effect of weighting control points depending on their quality is schemed in Figure 4.10. Higher weights (e. g., for edge and high curvatures) force the spline curve to the control points. In contrast, lower weights reduce the influence, e. g., for low quality points and outliers.



(a)



(b)

Fig. 4.10: Two examples for correcting a scanline using NURBS curves. Weights are applied to the thick points. The effect of assigning higher weights to significantly curved regions and edges is illustrated in (a). In contrast, the effect of lower weights for artifacts and outliers is shown in (b).

These scanline procedures (i. e., smoothing, quality estimation, curve approximation) are finally applied to real data sets as shown in Figure 4.11. Thus, the following section discusses the corresponding results at the example of different point cloud models.

4.5 Case Studies

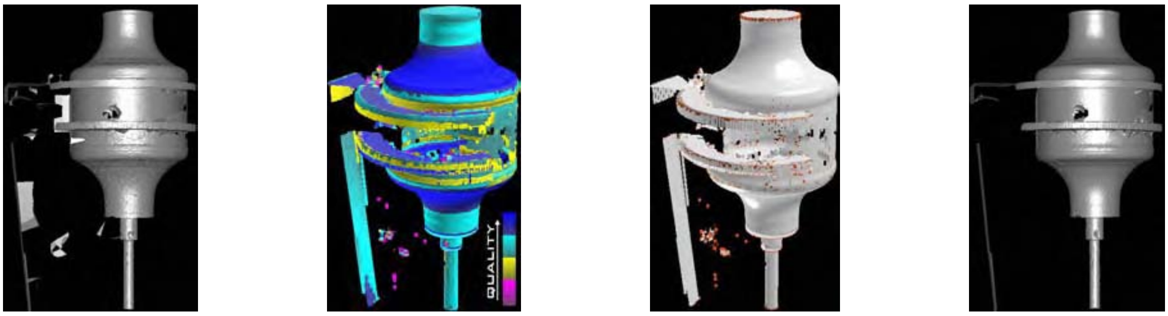
The effectiveness of the presented approaches partially depends on the properties of the surface to be measured. For this reason, the methods were tested on different point clouds captured from different surface types and surface topologies. To demonstrate the results, Figure 4.11 shows four phases of the approach. For a better visibility of the manipulations made, each first image per row shows a simple and non-optimized triangulation of the raw, erroneous data. The second image illustrates the quality of each measured point miscolored for one sensor (laser and camera). Significant curvatures respectively object structures used for the NURBS-approximation are displayed in the third image.

After processing the whole data with the presented methods, the obtained optimized point cloud is shown again as polygonal representation in the fourth image. For demonstration, a technical element (catalytic converter for vehicles), a clay model of Santa Claus and of a duck and a green pepper as an organic object were chosen. These models are considered to be representative, because they exhibit different surface properties that can cause different degrees of noise. Since an optical system is used, their reflectivity plays an important role. Furthermore, the object shapes vary from widespread to locally highly curved regions, and the point clouds contain holes, gaps and outliers.

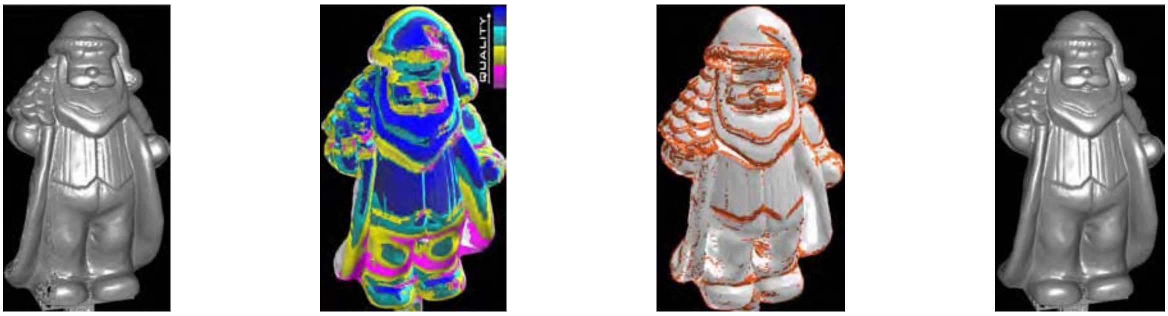
The metal, and thus reflective surface of the converter caused a lot of errors that could largely be detected and corrected (see Fig. 4.11(a) and 4.11(d)). In particular, the originally craggy scan of the surface of the converter could be smoothed by taking the estimated metering precision of the scan system into account.

Considering the model of Santa Claus, the clay surface caused only few artifacts that could largely be removed. However, the noise in the areas of his face and his legs could be notably reduced (see Fig. 4.11(e) and 4.11(h)). The overall quality of the 3D data from the duck could be increased, especially to be seen at the reflections on its sides. The organic surface of the green pepper was partly translucent. By adapting the laser energy, a good measurement was possible. Nevertheless, the measured data was noisy, but could be smoothed sufficiently. Attention should be paid to the parameters of the B-spline approximation. Closing small gaps can improve the consistency of the surface, but real existing gaps were also closed (see holes in the object carrier in Figures 4.11(l) and 4.11(p)).

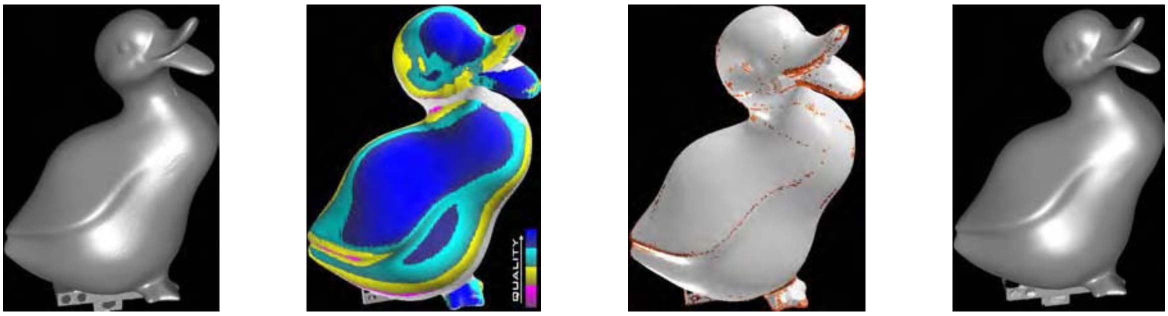
The proposed methods are stable and the mostly correct surfaces are not changed substantially. In contrast, the applied modifications are all within the error tolerance of the tested laser scan system. In addition, the density of example point clouds could, on average, be reduced by 60% without appreciable loss of information.



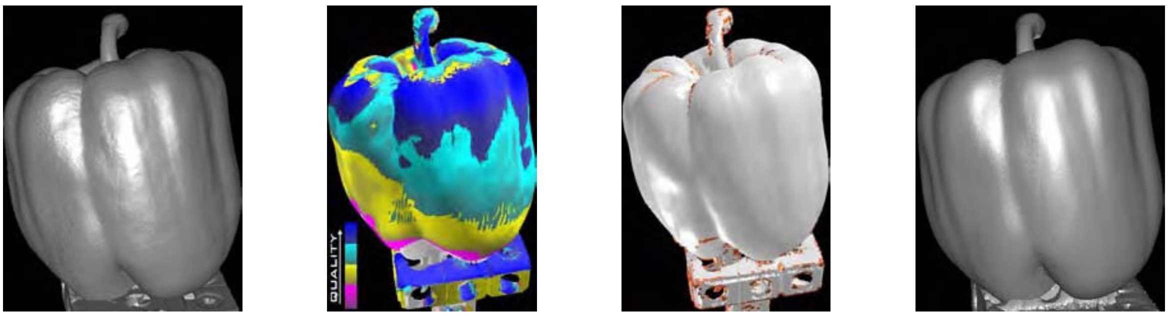
(a) Original. (b) Evaluated. (c) Significant curvatures. (d) Optimized.



(e) Original. (f) Evaluated. (g) Significant curvatures. (h) Optimized.



(i) Original. (j) Evaluated. (k) Significant curvatures. (l) Optimized.



(m) Original. (n) Evaluated. (o) Significant curvatures. (p) Optimized.

Fig. 4.11: Examples for triangulated point clouds, before and after processing. For each row: triangulated original raw data, evaluated quality (camera viewing angle), significant curvatures ($\kappa > 0.2$), triangulated optimized data.

4.6 Summary

In this chapter, methods were presented to automatically evaluate, quantify, and correct point clouds generated by an optical 3D scanning system. Techniques were proposed that are based on the system's parameters in 2D (e. g., contrast and line thickness) and in 3D (e. g., camera and laser positions, focus area, etc.) to estimate the quality of each single point. Furthermore, point clouds were approximated by a sorted set of B-spline curves to iteratively smooth and close gaps. Edge information from these curves was derived, and scanlines finally reconstructed by using NURBS curves with respect to quality and curvature of each single point on it. The adjustment made to the point clouds can be controlled by tolerances, thus the metering precision of the whole system is not declined. The main goal of the considerations was to analyze and correct the 3D point sets. The proposed methods are based on a general optical measuring principle that is also used in other scanning systems, and thus can easily be adapted to other scanning systems.

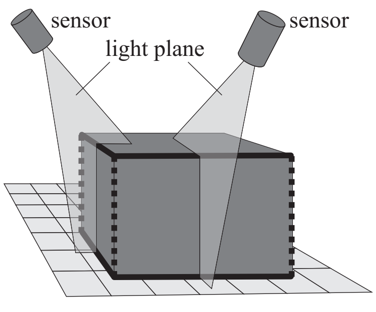
The described methods and the acquired systematic coherences allow further considerations of how to use the parameters obtained. The quality values from the point cloud quantification can be used for an automated object and system device positioning to improve the general scanning properties. This can be achieved through an iterative process by measuring and evaluating until an optimum is reached. Furthermore, triangulation algorithms can use the smoothed and optimized point set as well as the edge information and the computed surface normals for more robust meshing. In addition, the edge information can be used for an automated feature recognition.

While the curve processing approach exploits the measuring principle that bases on line scans, the following chapter discusses the additional comprehension of movement information between scanlines. Therefore, the algorithms developed here are extended to B-spline surfaces to enable more detailed interrelated considerations and interpolations on neighboring scanlines.

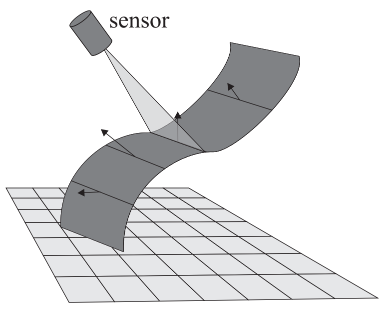
Chapter 5

Surface-based Scan Data Processing

Duplicating unique works of art, archaeological findings or work pieces for the industrial mass production are usually applications for 3D scanners. A manual scanning procedure can be time-consuming, especially for large or complex objects with many small and important surface features. To assure that all areas of the object are captured within a partial scan, a fast preview method is desirable, which assists the user to control and evaluate the scanning process and the results. However, a set of local scanline approximations, as proposed in the previous sections, is obviously not sufficient for an evaluation of global neighborhoods (see Fig. 5.1(a)). The curve approximation extracts information by exploiting the principles of the light sectioning method which projects line patterns. By using the additional information that is given from the measuring system, multiple scanlines can be considered. Thus, the positions and movements of the locomotor systems are helpful, because the scanlines are implicitly sorted as a sequence of single measurements, while the object is moved in front of the sensors.



(a)



(b)

Fig. 5.1: Illustration of problematic cases for line scans when edges are parallel to the light plane. Thick edges are captured, the dotted ones are not. The computation of scanline features (e. g., curvatures) does not have a global character (b), the captured line is straight, but the surface is not.

In addition to the scanline processing, the goal is here to utilize this additional information and generate grids that preserve the scanline neighborhoods. This allows to consider surface properties, which cannot be derived from singles curves alone. For example, edges cannot be detected by scanlines if the surface curvature is perpendicular to the scanline (see Fig. 5.1(b)). A fast grid-based preview also allows to check for gaps, shadowing effects and interfering reflections. As a result, the scanning procedure becomes faster and more comfortable and thus, the results are even better. Furthermore, surface features often have to be detected and evaluated in order to check the geometry and adjust the scanning procedure.

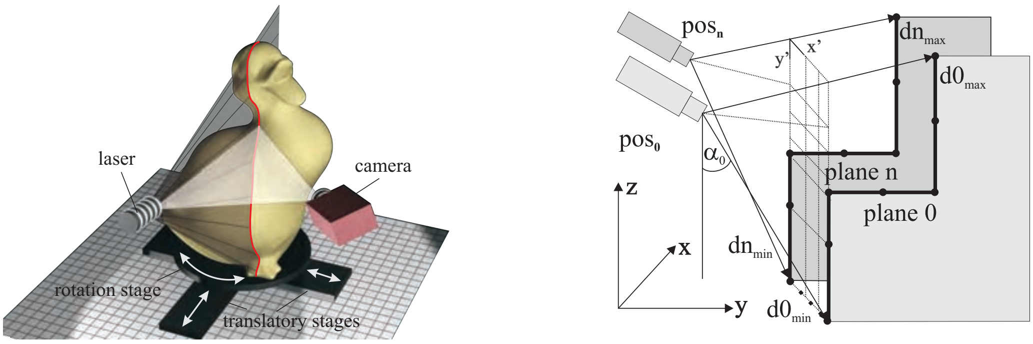
This chapter introduces a new approach for the fast visualization and evaluation of large point clouds derived from different structured light 3D scanners. By using information derived from the underlying measuring principle, regular quadrilateral grids are nearly computed in real-time. These grids are additionally used for partial NURBS approximations. Based on the derived parameterized surfaces, higher order properties are computed (e. g., surface curvatures).

5.1 Grid Generation

In many cases, algorithms require surface types of a certain connectivity. Usually, triangle meshes are used. The process of generating other mesh types from a given one is called remeshing. For example, Hormann et al. propose a method to generate quadrilateral meshes from triangles [HG00]. They present a processing pipeline from unstructured point sets over triangle meshes to quad meshes, and finally fit cubic B-spline patches to achieve smooth surfaces [Hor03].

In this section, we seize this idea. But in contrast, the grid generation approach aims to construct a regular quadrilateral "mesh" without pre-calculating triangle meshes but by employing model information from the scan system. Therefore, a simple technique is used, which is known as mapped meshing [Owe98]. At first, a grid and its dimensions are automatically defined. The horizontal dimension (x') of the grid is set equal to the number of scanlines. The vertical sampling (y') depends on the object height (for example y'=300 for an object height of 300 mm). Furthermore, the relation between two scanlines is a known system parameter (translation and/or rotation). Depending on the scanning direction and the number of sampling steps, the sorting and the distances between each two scanlines is determined. Additionally, the point sorting on a scanline is checked, depending on the actual laser or projector position (see curve processing in the previous chapter).

The main approach is the projection of the pre-defined grid onto the points of a scan operation. To achieve a regular distribution and an unambiguous mapping, the projector position is used. This is a safe location for each scanline from which all points have been seen without self-intersections and shadowing effects. Based on the perspective projection, a unique direction of projection and the corresponding projection angle is assigned to each point of a scanline (see Fig. 5.2(b)). The next step finds a corresponding neighbor for each point with respect to all scanlines of the scans. Therefore, it is assumed that points that have been seen from the same direction of projection are at



(a) (b)

Fig. 5.2: Data acquisition principle using a laser sensor and a movable object on rotational and translatory stages (a). Each set of scanlines within one movement is called a scan operation and is interpolated by a single grid. The grid construction with the minimum and maximum projection angles with respect to one scan operation is illustrated in (b).

least near or neighbors, which is valid, if the surface is slightly changing from scanline to scanline [TBTW05].

In practice, neighboring scanlines have different point densities and distributions. Additionally, the length and shape are changing. Thus it must be assured that all points from the same direction of projection for all scanlines i of one scan are mapped to one horizontal line in the grid. Therefore, each projection angle α_i is required. It is derived from the point on the scanline $p(i,k)$ and laser position $p_L(i)$ that belongs to the same scanline. The angle is computed in relation to the up-vector $\vec{v}_{up}$ (for this system the z-direction). The direction of projection $\overrightarrow{d_P}(p(i,k))$ points from the laser position to the actual point.

$$\alpha_i = acos(\vec{d_P}(p(i,k)) \cdot \vec{v}_{up}), \quad \text{with} \quad \vec{d_P}(p(i,k)) = \frac{p(i,k) - p_L i}{\|p(i,k) - p_L i\|} \tag{5.1}$$

Considering all scanlines i of one operation, there is a minimum and a maximum projection angle. Those two angles limit the grid in the projection space. All grid positions $g(i,\alpha_j)$ are calculated, based on the defined dimensions, by linear interpolation between those two angles. Then, for each point $p(i,k)$ the projection angle α_k is computed, which is unique for its scanline i. Finally, p is back projected and a 3D coordinate is assigned for the corresponding grid position (Eq. 5.2). As a result, each scan operation (set of scanlines after rotation or translation) is interpolated by a regular row/column grid. In contrast to standard terrain modeling, angular relations are used, because there is no common background plane and the back projection of rotational scans could not be handled and would destroy the scanline structure.

$$\begin{aligned} & g(i,\alpha_j) = p(i,\alpha_k) \quad \text{with} \\ & |\alpha_j - \alpha_k| < |\alpha_j - \alpha_{k-1}| \wedge |\alpha_j - \alpha_k| < |\alpha_j - \alpha_{k+1}| \end{aligned} \tag{5.2}$$

The grid generation procedure is additionally given in Algorithm 5.1.

Input: sets of X scanlines from one scan operation, desired vertical dimension Y
Output: regular row/column grid with dimension (x,y)
foreach *scanline S of the operation O* **do**
  obtain laser position L_{pos}
  foreach *point p on a scanline* **do**
    calculate and store projection angle $\alpha_p^i(p_i,L_{pos})$
  end
end
calculate α_{min}=MIN(α_p) and α_{max}=MAX(α_p)
generate gridMatrix(x,y)
foreach *knot(x_i,y_i)* **do**
  grid[x_i][y_i]=*dummy*
end
foreach *scanline S of the operation O* **do**
  foreach *point p on a scanline* **do**
    y_{pos}^i=$(y \cdot \alpha_p^i)/(\alpha_{max} - \alpha_{min})$
    grid[x_i][y_{pos}^i]=p_i
  end
  x_i=$x_i + 1$
end
Algorithm 5.1: Algorithm for the generation of a regular grid for one scan operation.

Changing point densities and gaps in the point cloud cause knot points without valid 3D positions. To keep the grid consistent, a dummy point with a zero weight is assigned to them. In further considerations, such knots are not evaluated. Additional degenerate cases may arise at positions where scanlines of a rotational operation overlap. These positions are detected by computing the intersection line of the laser planes. This case can also be avoided by choosing the rotation axis outside the measuring range. For grid consistency, a zero weight is assigned to them as well.

Finally, the initially unstructured point clouds are now structured and interpolated by a regular row/column grid. Therefore, the spatial dimension was reduced from 3D to 2.5D by using the projector/laser positions for a unique grid mapping. A comparative study for surface parameterization methods, i. e., mapping one surface into another, is given by Floater and Hormann [FH05].

5.1.1 Grid Smoothing

The discretization of the measured points may result in single gaps within the computed grid. Thus, the initial basic grids possibly have to be repaired and smoothed. Gaps are located as unused dummy knots in the grid structure (zero weight). Small gaps in a 2×2 region are closed by linear interpolation between surrounding valid knots. Because of the very regular structure of the row/column grids fast algorithms can be applied. For example, a smoothing is applied with the recursive Laplacian operator

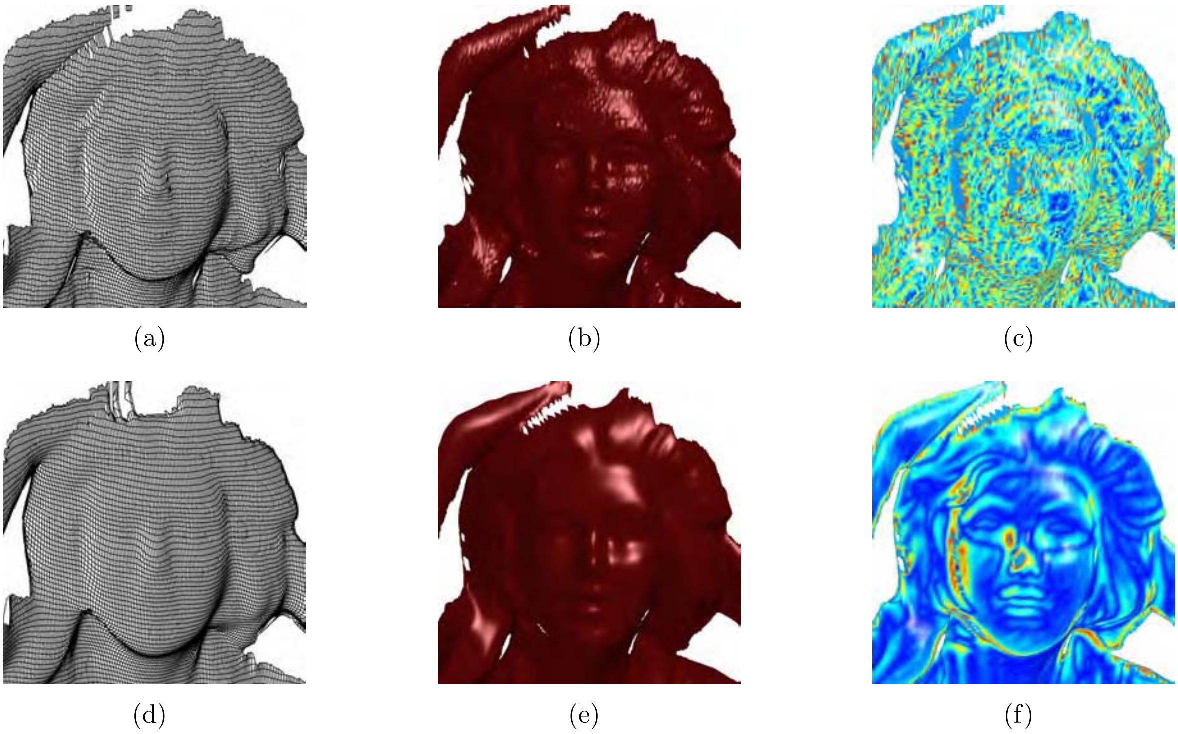


Fig. 5.3: Result of the grid generation process: projected grid (a), shaded representation (b) and the color coded length of the second derivatives (c), indicating the grid's fairness (from blue over green and yellow to red). The first row shows the initial grid and the second its smoothed representation.

(Eq. 5.3). For each grid point x all N topological neighbors are considered and the mean value is assigned. The influence of these points on the new position of the actual point is controlled with the weighting parameter λ. As a result, the grid structure becomes more regular, and noise is reduced. As proposed in [KVPL04], a smoothing within a 3×3 neighborhood is applied to the model in Figure 5.3, which is typically sufficient to achieve a fair grid.

$$x^r = x^{r-1} + \lambda \sum_{i=1}^{N} \frac{x_i^{r-1} - x^{r-1}}{N}, \quad 0 < \lambda < 1 \tag{5.3}$$

Furthermore, applications often require that a smoothing operation can be limited to a certain assumed uncertainty. Therefore, a tolerance δ is added to control the movement of control points with respect to the metering precision. Within each iteration, the tolerance is checked with Eq. 5.4. This guarantees that the new coordinates are not moved more than the given δ units. Usually, 4 iterations are sufficient to achieve a smooth result, and the maximal allowed displacement δ is reached within 10 iterations.

$$x_i^r = \begin{cases} \tilde{x}_i^r, & \text{if } \|\tilde{x}_i^r - x_i\| \leq \delta \\ x_i + \delta \cdot \frac{\tilde{x}_i^r - x_i}{\|\tilde{x}_i^r - x_i\|}, & \text{if } \|\tilde{x}_i^r - x_i\| > \delta \end{cases} \tag{5.4}$$

The result of the grid generation and the effect of the smoothing operations are illustrated in Figure 5.3. An objective quality evaluation is given by the maximums of the first derivative [BSG+03]. They denote the strength of an edge and serve as indicator for the smoothness and fairness of the interpolated grid (for a more detailed discussion see Sect. 5.1.2).

5.1.2 Grid Analysis

Up to this point, the grids consist of linearly connected line segments, which represent the point cloud, and the surface respectively. For the computation of surface features derivatives are needed. Therefore, this section discusses discretization and difference methods for regular grids.

The first derivative of a smooth function u with the grid spacing h can be calculated in many different ways. The simplest methods to approximate the first derivative are the forward ($\delta_h^+ u(x_k)$) and backward ($\delta_h^- u(x_k)$) differences:

$$\delta_h^+ u(x_k) = \frac{1}{h}(u(x_{k+1}) - u(x_k)) \tag{5.5}$$

$$\delta_h^- u(x_k) = \frac{1}{h}(u(x_k) - u(x_{k-1}))\ , \tag{5.6}$$

which are one-sided differences of first order. Higher order differences are obtained by repeated operations of the difference operators (see Eq. B.2).

The central (symmetric) difference $\delta_h^0 u(x_k)$ and the second derivative $\delta_h^2 u(x_k)$ are of second order:

$$\delta_h^0 u(x_k) = \frac{1}{2h}(u(x_{k+1}) - u(x_{k-1})) \tag{5.7}$$

$$\delta_h^2 u(x_k) = \frac{1}{h^2}(u(x_{k+1}) - 2u(x_k) + u(x_{k-1}))\,. \tag{5.8}$$

The differences at the grid borders are undefined due to the missing neighbor knots. Thus, they are set to zero.

For the grids these equations are extended to the two-dimensional case, where the differences are applied in the x- and y-direction, $\delta_{h,x}^+$ and $\delta_{h,y}^+$ respectively. The sum of both differences results in the Laplace operator $\Delta u = u_{xx} + u_{yy}$:

$$\Delta_h u(kh, lh) = (\delta_{h,x}^2 + \delta_{h,y}^2)u(kh, lh) = \frac{1}{h^2}(u_{k-1,l} + u_{k+1,l} + u_{k,l-1} + u_{k,l+1} - 4u_{kl})\,. \tag{5.9}$$

By using those differences, all partial derivatives $u_x, u_y, u_{xx}, u_{yy}, \Delta_u$ are computed. The mixed partial derivative in x and y (u_{xy}) is approximated by the product $\delta_{h,x}^0 \delta_{h,y}^0$:

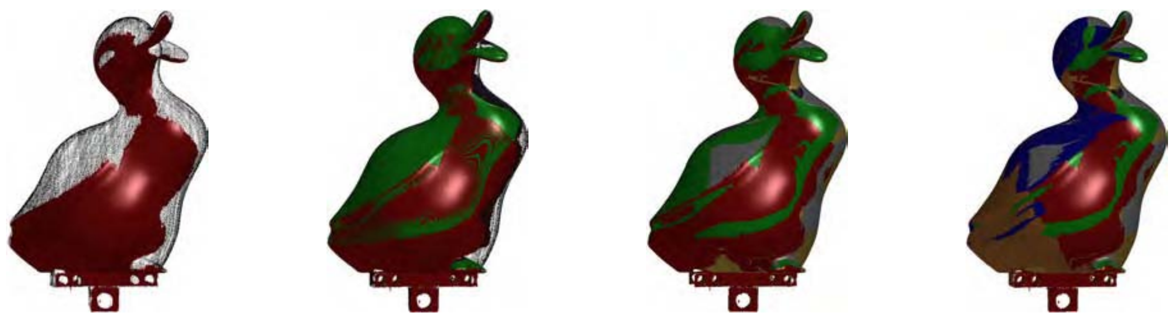
$$u_{xy}(kh, lh) \approx \frac{1}{4h^2}(u_{k+1,l+1} + u_{k-1,l-1} - u_{k+1,l-1} - u_{k-1,l+1})\,. \tag{5.10}$$

In the last column of Figure 5.3, the zero crossings of the second derivatives, which are the maximums of the first derivative, are employed to visualize the regularity and homogeneity of the projected grids.

There are several other approaches to detect high frequency features. For example, KOBBELT ET AL. propose to perform an automatic edge and corner recognition in triangle meshes. By using a simple heuristic based on the local normal vector orientations, they compute the opening angle of the normal cone for a vertex. If this angle is smaller than some threshold, they expect the surface to have a sharp feature [KBSS01]. Since this method is designed for high-quality industrial CAD models, it is not suitable for data affected by noise and other artifacts. In this case, neighboring normal vectors may exhibit strong variations in their directions and the proposed heuristic would fail. If a prior smoothing operation is performed, it may work for scattered data as well.

5.1.3 Grid Registration

Because of uncertainties from the calibration procedures of simple 3D scanners, there are small deviations between overlaying grids from different sensors and they do not match exactly. For other systems, movement information between scan operations may not be available. Therefore, the Iterative Closest Point (ICP) algorithm [RL01] is usually employed to match the data. This algorithm computes a transformation that minimizes the disparity function X in which x_j for j= 1,..,Nx is a set of points on the surface X and y_j is a point on the surface Y that corresponds to the point x_j (e.g., the closest point on Y). To find this closest point, one triangulates the surface Y and projects the point x_j onto the surface.



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

Fig. 5.4: Illustration of the generated grid for one scan operation (a) and the successive combination with other grids (b,c). The last figure (d) shows the result of the finally registered grids.

$$d(T(X), Y) = \sqrt{\sum_{j=1}^{N_x} |T(x_j) - y_j|^2}, \quad y_j = C(T(x_j), Y) \tag{5.11}$$

This is usually achieved by computing the intersection of a line passing through the point that is perpendicular to the surface. For the grids, the correspondence problem can be solved easily. The information on neighboring points within the regular grid needs no triangulation. In our algorithm we start the first iterations with a pre-orientation based on a point-to-point query. This is achieved by building a kd-tree from the points of the static grid. The nearest neighbor for the dynamic point cloud is then efficiently found by searching the tree. Once the correspondence is established, the transformation that minimizes (Eq. 5.11) is computed by using the least-squares method. The process is repeated until convergence (ICP is guaranteed to converge [JH02]). Furthermore, the distances between overlaying grids are only small and thus, only a few (<10) iterations are needed. An alternative method, which uses a non-rigid alignment on the basis of thin plate splines at the overlay borders is presented by Brown et al. [BR04]. Their method is designed for initially unaligned meshes.

5.1.4 Intermediate Results

After the fast generation of the grids, high quality previews of the actual scan operations were produced fully automated and in real-time (see Fig. 5.5). This approach supports the user to detect gaps and check for the point cloud completeness, and additionally illustrates the correct alignment of several surface patches derived from different scan operations and sensors. The resulting models in Figure 5.5 were generated with two laser sensors, one on top and one on the bottom to capture the entire surface. The sensors were static and the objects were moved by using one rotational and two translatory axes. As a result, the point clouds consist of several scan operations (set of rotational or translational movements), which consist of scanlines and sublines.

The models in Figure 5.5 represent the variety of typical objects for optical 3D metrology. There are industrial surfaces (steel, aluminum (i-k)), clay(b-e), organics(f), gypsum(l), ceramics(a) and synthetic material(g,h). Furthermore, there are lateral(d), rotational(l) and combined scans (all others) and different topologies which are homeomorphic to a sphere(c) and a torus(e) etc.

Exploiting this structures from the measuring system and the measuring principle allows very efficient procedures. A performance evaluation for the presented models in Figure 5.5 is given in Table 5.1. Of course, the total computation depends on the number of input points and scan operations, respectively. Compared to the data acquisition which takes several minutes, the grid generations only take 500 ms, and thus, it can be termed as real-time. The computation time includes a smoothing in a 3×3 neighborhood with 4 iterations. The ICP-based grid was not applied for the scan result preview and for the evaluation of the calibration accuracy. For completeness, this would result in an extra time of approximately 12 sec. per model.

The grid generation is an excellent method for the visualization, but it is not suitable for the robust computation of surface features. The approximation of derivatives with differences requires an equidistant grid spacing, which is not guaranteed. Furthermore, the grid only consists of linearly connected line segments. Higher order analyses require a larger neighborhood and region of influence to assure a certain continuity. To overcome this problem, the preview grids are used as input for a NURBS surface approximation in the next step, which allows a more comfortable and robust computation of the surface curvature.

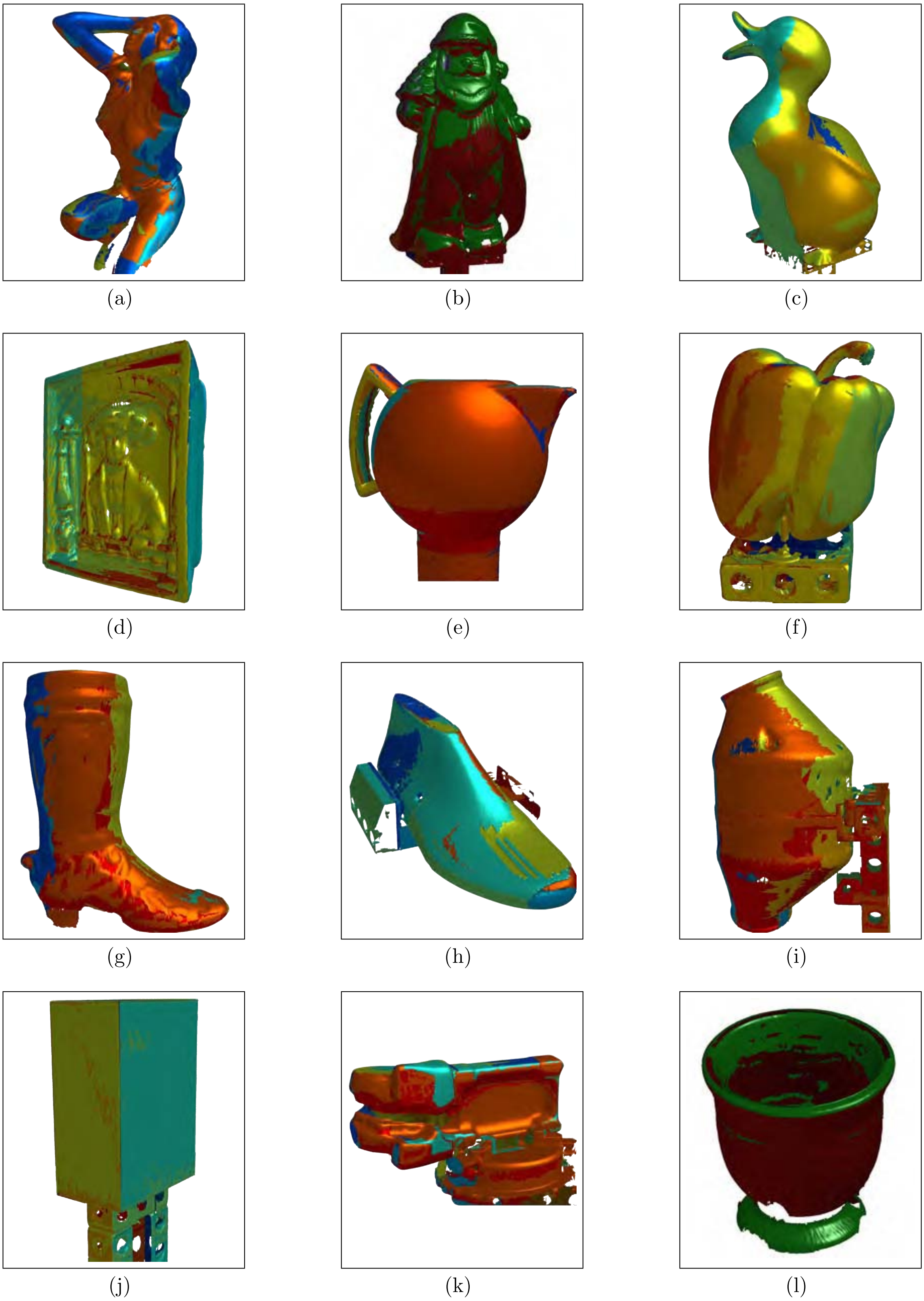


Fig. 5.5: Examples for generated grids with the proposed methods. The scans contain rotational and translational operations between the scanlines. The grid for each of those scan operations is assigned a different color.

model	points	grids	polygons	total time	time/grid
woman (a)	1.201.829	8	742.956	515ms	64.4ms
Santa Claus (b)	6.833.877	32	2.674.852	2.573ms	80.4ms
duck (c)	1.355.041	8	635.066	428ms	53.5ms
cast tile (d)	1.605.784	12	738.217	555ms	46.3ms
can (e)	1.216.448	10	544.692	458ms	45.8ms
pepper (f)	1.581.587	16	793.905	526ms	32.9ms
boot (g)	927.901	8	403.959	307ms	38.4ms
shoe (h)	953.765	8	404.519	247ms	30.9ms
converter (i)	1.488.444	8	516.023	389ms	48.6ms
cube (j)	1.186.406	8	403.297	281ms	35.1ms
casting (k)	1.656.360	8	879.487	683ms	85.4ms
eggcup (l)	747.468	2	356.063	190ms	95.0ms

Tab. 5.1: Performance evaluation for the construction of the initial grids with respect to the models illustrated in Figure 5.5. For each model, the number of captured 3D coordinates is given in the second column. For each connected set of scanlines (scan operation) a grid is constructed and their number is shown in the third column. The total resulting number of generated quadrilateral polygons for each model is given in column four and the computation times (Pentium4 3.2GHz) in the last columns.

5.2 Parametric Surfaces

Robust measurements on the interpolated surfaces require a precise analytical description. The described grid construction approach as well as triangulation methods (e. g., Delaunay, Marching Cubes, Ball-Pivoting) generate piecewise linear meshes. To determine surface features on such models with differential geometry methods interpolations and approximations within a certain neighborhood about the considered region have to be used. Thus, a local region is assumed to represent a higher degree surface patch (e. g., quadrics). The algorithms are typically applied to all points/polygons, but in most cases without regard to neighboring patches. For perfect input data sufficient results are achieved, but real measuring data is characterized by noise, uneven sampling and uncertainties and, thus, fitting procedures fail.

The optimal solution for this problem is a parametric surface, such as B-spline patches. B-spline surfaces exhibit the same smoothing character concerning noise and well behavior as B-Spline curves. Furthermore, they give the possibility to measure at arbitrary positions in the parametric space and guarantee a constant continuity between connected subpatches. Especially rational (weighted) B-spline surfaces are practical in this case, since they enable to define control point specific influence values. The corresponding procedures are discussed in this section.

The generated quadrilateral and regular row/column grids introduced in the last section serve as control net for a B-spline surface approximation. The zero weights caused by unused knots or existing gaps in the underlying grid are passed to the rational B-Spline surface. The influence of these points on their environment becomes zero and the consistency of the control net is kept. The initial weights for valid grid positions are set to one. Due to the sampling theorem, the control net dimensions should not exceed the half of the number of scanlines and points per line in this case.

5.2.1 NURBS Surface Approximation

A non-uniform rational B-spline surface S of degree p, q is defined by the recursive basis functions N, the weights w, and the points of the control net B.

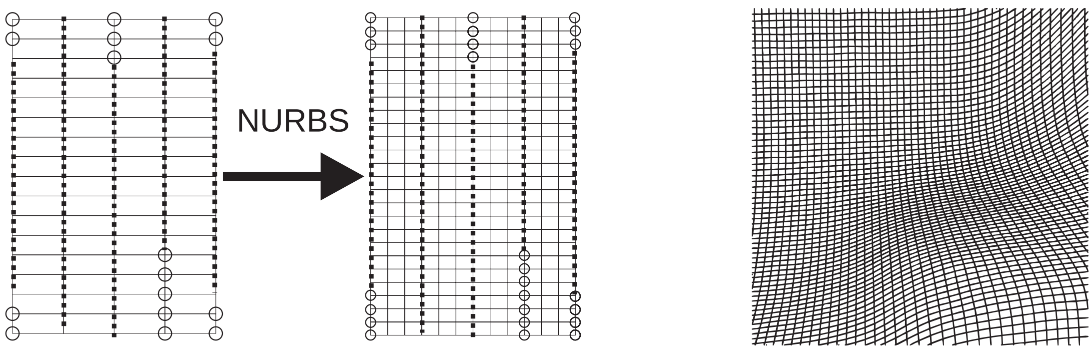
$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}B_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} \tag{5.12}$$

The recursive basis functions for both parametric directions u and v can have a different degree. They are given by using the following equation:

$$N_{i,p}(u) = \frac{u - i_i}{u_{i+p} - u_i} N_{i,p-1} + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{5.13}$$

$$N_{i,0}(u) = \begin{cases} 1 & u_i \le u \le u_{i+1} \\ 0 & \text{otherwise.} \end{cases} \tag{5.14}$$

The analysis with differential geometry methods mostly requires the calculation of at least the 1st and the 2nd derivative of the surface. Thus, 3rd degree (4th order) polynomial surfaces are practical, which is also typical for CAD applications [FvDFH00]. Furthermore, the influence of one point of the control net is defined by the order of the surface. Higher degree surfaces may be calculated to increase the influence region and for test purposes. Therefore, the implementation of the NURBS surface algorithms is variable in degree. Figure 5.6 illustrates the procedure to generate B-Spline surface from the grids.



(a) (b)

Fig. 5.6: Approximating the basic grid (with marked unused knots) (a) by a NURBS surface (b), and an example of a real grid after back projection to the 3D space (c).

The approximated surfaces have a smoothing character, depending on their degree and the resolution of the input grid. Since the sampling theorem requires half the number of control points than measured points, the resolution for the B-spline surface is chosen twice as high as the control points. Up to this point, the weights are only used to mask invalid grid positions. But the information about local curvature from the scanline processing and the quality of single points with regard to their viewing and projection properties can be applied to the B-spline surface, too. Therefore, the weights from Eq. 4.23 are multiplied with the initial grid weights, which are zero or one. The NURBS surface may also be used to locally increase the point density or interpolate points between neighboring scanlines.

5.2.2 Surface Analysis

The shape analysis (e. g., edges, bumps) of parameterized geometrical surfaces is usually based on Gauss' fundamental theorem surface theory. The parametric form of a surface is defined with the three Cartesian coordinates x,y and z. A surface is defined by the relation r=r(u,v) with the real parameters u,v and the relations x=x(u,v), y=y(u,v) and z=z(u,v). For the determination of metric surface properties, there are three types of so-called fundamental forms. The most important are the first and second form (since the third can be expressed in terms of these). The first Gaussian fundamental theorem for curved surfaces is explicitly given by the Riemannian metric (Eq. 5.15) and provides metric properties such as length and area. The second fundamental form is the symmetric bilinear form with respect to the tangent space of the first and provides metric surface properties such as mean and Gaussian curvature. The second form is given in Eq. 5.16. The coefficients of both forms are given in Eq. 5.17.

$$ds^2 = Edu^2 + Fdudv + Gdv^2 \tag{5.15}$$

$$-dNdr = Ldu^2 + 2Mdudv + Ndv^2 \tag{5.16}$$

While the first fundamental form contains all the metrical information of the surface, the second fundamental provides information about the surface curvature. The parameters of Eqs. 5.15 and 5.16 are given as the dot products between the first and second surface derivatives and the normal vector N:

$$E = r'_u r'_u, \quad F = r'_u r'_v, \quad G = r'_v r'_v, \quad L = r''_u N, \quad M = r'_{uv} N, \quad N = r''_v N\,. \tag{5.17}$$

Derivatives of Rational B-Spline Surfaces

The calculation of the derivatives of rational B-spline surface Q is much more complicated than in the non-rational case. The ratios of the weights must be included in the computation for the basis functions [PT95]. The corresponding terms are given by Eqs. 5.18 to 5.22.

$$Q_u = \frac{\bar{N}}{\bar{D}}\left(\frac{\bar{N}_u}{\bar{N}} - \frac{\bar{D}_u}{\bar{D}}\right) \tag{5.18}$$

$$Q_w = \frac{\bar{N}}{\bar{D}}\left(\frac{\bar{N}_v}{\bar{N}} - \frac{\bar{D}_v}{\bar{D}}\right) \tag{5.19}$$

$$Q_{uw} = \frac{\bar{N}}{\bar{D}}\left(\frac{\bar{N}_{uv}}{\bar{N}} - \frac{\bar{N}_u}{\bar{N}}\frac{\bar{D}_v}{\bar{D}} - \frac{\bar{N}_v}{\bar{N}}\frac{\bar{D}_u}{\bar{D}} + 2\frac{\bar{D}_u}{\bar{D}}\frac{\bar{D}_v}{\bar{D}} - \frac{\bar{D}_{uv}}{\bar{D}}\right) \tag{5.20}$$

$$Q_{uu} = \frac{\bar{N}}{\bar{D}}\left(\frac{\bar{N}_{uu}}{\bar{N}} - 2\frac{\bar{N}_u}{\bar{N}}\frac{\bar{D}_u}{\bar{D}} + 2\frac{\bar{D}_u^2}{\bar{D}^2} - \frac{\bar{D}_{uu}}{\bar{D}}\right) \tag{5.21}$$

$$Q_{ww} = \frac{\bar{N}}{\bar{D}}\left(\frac{\bar{N}_{vv}}{\bar{N}} - 2\frac{\bar{N}_v}{\bar{N}}\frac{\bar{D}_v}{\bar{D}} + 2\frac{\bar{D}_v^2}{\bar{D}^2} - \frac{\bar{D}_{vv}}{\bar{D}}\right) \tag{5.22}$$

The nominators $\bar{N}$ and denominators $\bar{D}$ in these equations as well as the partial derivatives with regard to the ratios are given in the appendix (see Eq. B.3). Basically, the recursive derivatives in the two parametric directions $N'_{i,k}(u)$, $M'_{j,l}(w)$, $N''_{i,k}(u)$, $M''_{j,l}(w)$ are given by the following Eqs. The recursive derivatives for $k = 0$ are zero.

$$N'_{i,k}(t) = \frac{N_{i,k-1}(t) + (t - x_i)N'_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N'_{i+1,k-1}(t) - N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}} \tag{5.23}$$

$$N''_{i,k}(t) = \frac{2N'_{i,k-1}(t) + (t - x_i)N''_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N''_{i+1,k-1}(t) - 2N'_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}} \tag{5.24}$$

5.2.3 Surface Curvature

Of fundamental concern in computer-aided design and reverse engineering are techniques to determine and visualize the fairness (smoothness) of surfaces in order to detect even the smallest deviations or to check for sharp edges or surface tensions. Since the derivatives are very sensitive to data affected by noise and outliers, their quality decreases with the number of derivatives. Usually, 4th-order surfaces which provide a C^2-continuity everywhere, are used. Nevertheless, they can exhibit unfair bumps, flat spots or undulations. One of the best mathematical techniques for determining surface fairness uses orthogonal nets of minimum and maximum curvature and of Gaussian curvature. At any point P on the surface S, the curve of intersection of a plane containing the normal vector n to the surface at P and the surface has a curvature κ (see Fig. 5.7). As the plane is rotated about the normal, the curvature changes, whereas a minimum and a maximum curvature arise.

The curvatures in these directions are called the principle curvatures, k_1 and k_2 respectively. The directions are orthogonal and they comply with the relation $d_k = \lambda e_1 + \mu e_2$ while:

$$\lambda^2(FN - GM) + \lambda\mu(EN - GL) + \mu^2(EM - FL) = 0\,. \tag{5.25}$$

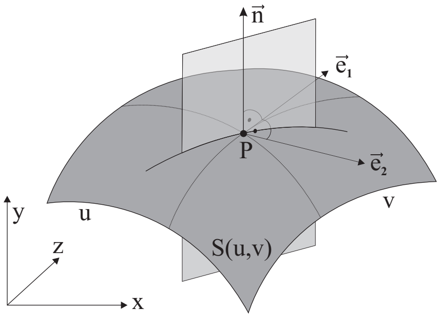


Fig. 5.7: Computing the curvature of surfaces. A biparametric surface S(u,v) is intersected by a plane. The resulting contour of intersection is a curve on the surface. The curvature is evaluated at the point P with regard to the surface normal $\vec{n}$.

The Gaussian curvature is defined as the product of the principle curvatures $K = k_1 k_2$, and the mean curvature is their average $H = 0.5(k_1 + k_2)$. For parametric surfaces, K and H are given in terms of the fundamental forms:

$$K = \frac{LN - M^2}{EG - F^2}, \qquad H = \frac{LG - 2FM + EN}{2(EG - F^2)}\,. \tag{5.26}$$

Given both quantities, the principle curvatures are derived from the quadratic equation $k^2 - 2HK + k = 0$:

$$k_1 = -H + \sqrt{H^2 - K} \tag{5.27}$$
$$k_2 = -H - \sqrt{H^2 - K}\,. \tag{5.28}$$

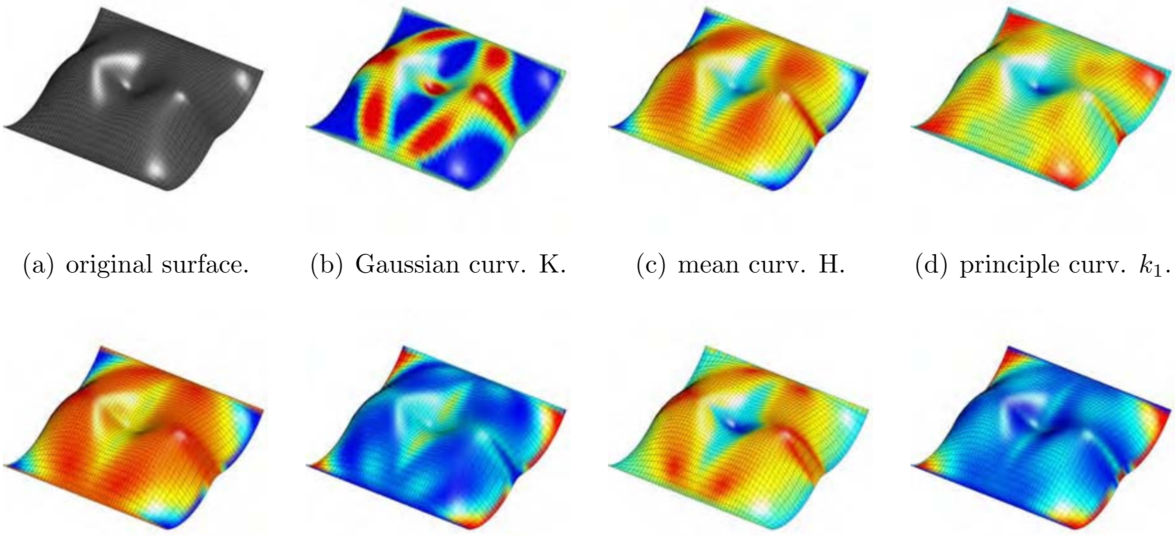
From Equation 5.26 follows that K is independent against transformation into other canonical coordinate systems, while H is not. Thus, K and only $|H|$ are real geometrical quantities.

The Gaussian curvature K and the mean curvature H are important surface features. Depending on the values of k_1 and k_2 one can derive geometric information about the shape. For example, a positive K value indicates bumps and a negative value pits. Other combinations allow to detect saddles, ridges, elliptical and cylindrical locations [RTvVV02].

Furthermore, Isenberg introduced the fold index [Ise04]. The desired measure should emphasize strong ridges and long, narrow valleys rather than treating every ideal ridge or rut equally. A surface point on one of these ideal features would have a negative first principal curvature (k_1) along with a second principal curvature (k_2) having the value zero (i. e., a point on a ridge) or k_1 with the value of zero and positive k_2 (i. e., a point on a rut). Then, the fold index F is given by:

$$F = \frac{2\mathcal{K}}{1 + \mathcal{K}^2}, \quad \text{with} \quad \mathcal{K} = \frac{k_2 + k_1}{k_2 - k_1}, \quad (k_1 \geq k_2)\,. \tag{5.29}$$

Koenderink and van Doorn developed a single-value, angular measure to describe local surface topology in terms of the principal curvatures [KvD92]. For their shape



(a) original surface. (b) Gaussian curv. K. (c) mean curv. H. (d) principle curv. k_1.

(e) principle curv. k_2. (f) fold index F. (g) shape index S. (h) degree of curv. C.

Fig. 5.8: 4th order NURBS surface with 5×5 control polygon and different types of surface curvature measures. All measures are color-coded between their minimal and maximal values from blue over green and yellow to red.

classification S each point lives in a Cartesian $k_1 \times k_2$ plane. Furthermore, they define the degree of curvature C. Both measures are given by:

$$S = \frac{2}{\pi} \arctan \mathcal{K}, \quad (k_1 \geq k_2, \quad k_1, k_2 \neq 0), \tag{5.30}$$

$$C = \sqrt{\frac{k_1^2 + k_2^2}{2}}. \tag{5.31}$$

In this case, C is the square-root of the deviation from flatness. Points with the same S value but differing C values can be thought of as being the same shape but only more stretched.

Since computation time is an important criteria, there is a more efficient way to compute the Gaussian curvature. An appropriate equation is proposed in [Rog01]:

$$\kappa_g = \frac{AC - B^2}{|Q_u \times Q_v|^4}, \quad \text{with} \quad \begin{bmatrix} A \\ B \\ C \end{bmatrix} = [Q_u \times Q_v] \cdot \begin{bmatrix} Q_{uu} \\ Q_{uv} \\ Q_{vv} \end{bmatrix}. \tag{5.32}$$

Although less operations are needed, this equation is numerically unstable. The reason was found in the 4th exponent of the length of the (unnormalized) normal vector $\vec{n} = Q_u \times Q_v$ within the equation. Because the length directly depends on the values of coordinates x,y and z, the 4th exponent may lead to numerical overflows and exponential multiplication of existing numerical errors. The intrinsic computation of the Gaussian curvature as the second derivative of a ratio of lengths is taken from

Thurston [Thu97]. Furthermore, the computation of surface curvature for irregular triangle meshes is discussed by Rusinkiewicz [Rus04] and Schulz [Sch05].

Three exemplary illustrations for the application of the surface curvature to the acquired scan data are given in Figure 5.9. The derived information is used to visually inspect complex data for completeness, gaps, bumps and pits on the one hand (see Fig. 5.9(a), 5.9(b)), and to automatically detect edges for object segmentation on the other hand (see Fig. 5.9(c)).

(a) (b) (c)

Fig. 5.9: Mean curvature of a scanned face (a), a casting (b) and the Gaussian curvature on scanned rivet on a plate (c). In the first step, the data was interpolated by a smoothed grid and in the second step it was approximated and analyzed on the basis of a rational B-spline surface.

5.3 Summary

In this chapter, an efficient procedure for the automated preview generation from several scan operations was presented. Furthermore, a more global data analysis, in contrast to the curve processing in the chapter before, was applied in order to check for surface features, geometric discontinuities, gaps and holes. By exploiting as much given information as possible from the measuring principle and the scanning system, the number of parameters could be minimized to employ these methods for automated processes.

The first section discussed methods for the establishment of a data sorting. Therefore, the position of the laser/projector has been exploited. The points on a scanline were sorted according to their projection angle in the same way as proposed for the curve processing. In the next step, the movements of the sensors were exploited to establish an order between single scanlines. Based on the projection angles, an approach for the construction of regular row/column grids was presented, whereas its parameterization is defined by the number of involved scanlines and the unique projection angles for single point coordinates. The grids were smoothed and registered using an ICP algorithm. Furthermore, the initial construction process was computed in real-time. While the scanline approaches can be employed to both measuring systems, the grid construction requires an ordered data acquisition as it is done by the complex measuring device.

The second section proposed a method for a rational B-spline surface approximation to the initial grids. These parametric surfaces were used to compute different surface curvatures in order to check for surface fairness. Additional algorithms for the evaluation of the surface topology were applied. They were based on shape descriptors, derived from the principle curvatures.

Within the proposed procedures, absolutely no user interaction or parameter adjustments are needed. All necessary information is derived from the measuring system and the underlying principle.

Due to the limitation that defined and uniformed movements between scanlines may not be given in every case, the following chapter considers the global point cloud. However, the scanline structure exists anyway, thus the point set can be enriched with additional information. In this regard, the next chapter discusses general 3D point set managing, optimization and evaluation.

Chapter 6

Point Cloud Processing

When analyzing laser scan data, usually very large data sets with millions of points have to be processed. Furthermore, optical 3D scanning techniques yield point clouds that usually contain several errors caused by system and measuring principle specific characteristics. Additionally, due to overlaying scans, redundant information with different quality is generated. Many applications do not require such a high point density, and additionally, the computation time for the data analyses is often limited in practice. Therefore, it is necessary to optimize the data and minimize the number of points while minimizing the loss of information at the same time. The basis for an efficient data processing are data structures which support the efficient retrieval of neighborhood information from the point set.

In the previous chapters, the point cloud representations were based on scanlines and grid meshes from scan operations, which exploited implicitly given neighborhood information. But, by using the flexible laser scanner, as presented in Section 3.1.2, and other system configurations, no exploitable information between points from single scanlines or operations may be derived. Thus, local neighborhood information must be constructed from the basic 3D point coordinates.

In connection to the discussions before, this chapter considers another level of abstraction, where local point information from different scans is derived and finally used for point cloud analysis and optimization. In particular, data structures, point cloud simplification, smoothing and nearest neighbor problems are discussed.

6.1 3D Data Organization and Representation

First of all, the set of 3D coordinates must be organized and structured in order to analyze local neighborhoods between multiple scanlines and scans from different sensors, etc. Therefore, a data structure is needed, which allows to efficiently search for points within the entire point cloud. There are different algorithms available for the structuring of 3D points, based on tree and graph representations or on the clustering of data. The most common data structures are introduced in this section.

An intuitive method for structuring 3D data is partitioning, where nearby points are assigned an unique partition or cluster. This approach is called spatial partitioning. A more efficient representation is provided by octrees, which subdivide the space into a hierarchically connected set of cubes. Later, the application of binary search trees, such as kd-trees and range trees, are discussed.

6.1.1 Spatial Occupancy Enumeration

In spatial-partitioning representations, an object is decomposed into a set of adjoining, non-intersecting primitives, which may vary in type and size. One of its most general forms is cell decomposition, in which the object is decomposed into identical cells arranged in a fixed regular grid [FvDFH00]. The cells are called voxels (voxel elements) in analogy to image pixels in the 2D case. The most common cell type is the cube. In this context, spatial-occupancy enumeration is of interest, where only the presence or absence of a single cell at each position in the grid is controlled. An object is then represented by the cells which are occupied and which are not.

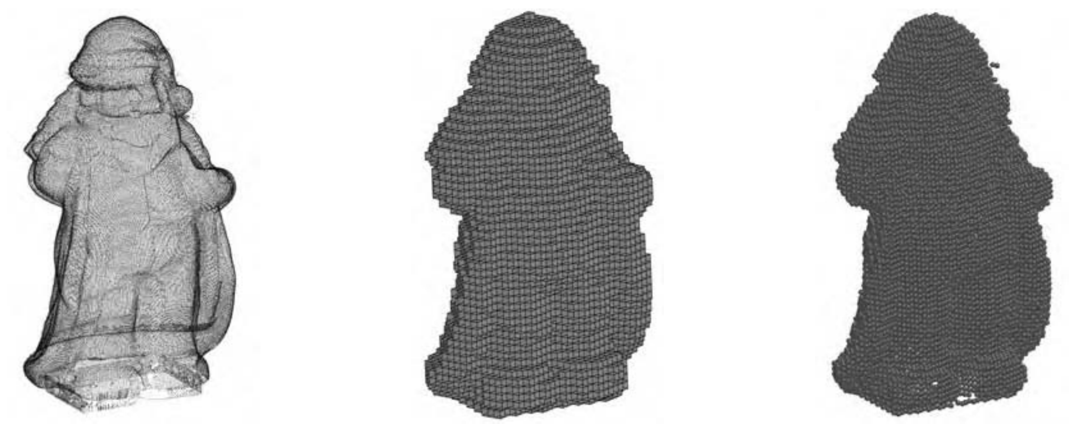
$$V_p(P) = \frac{1}{c_s} \cdot \begin{bmatrix} P_x - m_x \\ P_y - m_y \\ P_z - m_z \end{bmatrix} \tag{6.1}$$

Therefore, the used 3D space, limited by the axis-aligned bounding box, is subdivided into a set of voxels. The resolution depends on the real size of the object and a voxel/cube size. The mapping transformation from object coordinates to voxel space is given in Eq. 6.1, with the position of a 3D point $P(x, y, z)$ in the voxel space V_p, the minimal coordinates m of the bounding box and the cube size c_s. Each point of the point cloud is assigned to one cube with a complexity of $O(1)$, since the bounding box, the resolution in the voxel space and the total number of voxels are known. Locating all points within a cube is also done in $O(1)$ time, since the data structure stores all voxels of the bounding box. In the next step, empty voxels are marked as eliminated and the remaining ones represent the scanned surface (see Fig. 6.1).

The advantage of this procedure is the fast access to voxels, but the obvious disadvantage is the high memory consumption. Cells may be created as small as desired to increase the accuracy. However, up to n^3 cells are needed to represent an object at a resolution of n voxels in three dimensions. Thus, for every possible voxel within the bounding box an associated data structure must be generated. For example, if a cubic volume is subdivided in 100 steps, one million voxels must be generated.

6.1.2 The Octree

An alternative data structure is an octree. This is a hierarchical variant of the spatial-occupancy enumeration, designed to optimize the storage requirements. The fundamental idea behind this structure is the divide-and-conquer power of binary subdivision.



(a) (b) (c)

Fig. 6.1: Spatial partitioning of a point cloud. The original surface points are illustrated in (a). A spatial-occupancy representation based on cubic and spherical cells is shown in (b) and (c).

They are used to partition a three-dimensional space by recursively subdividing it into eight octants. Thus, each internal node has up to eight children. The recursion stops at a user-defined cutoff depth, a minimum cube size or when an octant contains only one point or is empty. As a result, only occupied cubes are stored, which significantly reduces the storage requirements. For every node, a pointer to its children and to the father ensures that the tree can be hierarchically traversed. Many computer graphics applications (e. g., Marching Cubes triangulation) utilize this data structure, which supports the implementation of divide-and-conquer strategies [CC06].

For further processing, the voxel centers are typically used as representation for the corresponding surface part. However, when analyzing the measuring data, no information should be discarded a priori this way and thus, the octree recursion should not have a predefined cutoff depth, which then may lead to very deep cascaded trees.

One of the most important applications is the determination of local neighborhoods. For example, finding the nearest or the m-nearest points to a given point is a typical task. An octree node has neighbors in 26 possible directions: 6 neighbors along a face, 12 neighbors along an edge, and 8 neighbors along a vertex. While the computing of the nearest neighbor is limited to these possibilities, finding the m-nearest neighbors in a given radius has to consider much more possible neighboring nodes, which can be wasteful and time-consuming. This is due to the fixed hierarchy and the fixed cubic cell size, which is relatively inflexible. A more flexible and sophisticated approach for spatial neighbor queries is given by a kd-tree, which is introduced in the next section.

6.1.3 The kd-Tree

A kd-tree is a general, multidimensional search tree for k dimensions. For 3D data the corresponding tree is usually called three-dimensional kd-tree instead of $3d$-tree. They are a special case of binary space partitioning (BSP) trees. Kd-trees use splitting planes that are perpendicular to one of the coordinate system axes (so-called hyperplanes), which is a specialization of the general BSP methods, in which arbitrary splitting planes can be used. A point set in a kd-tree is subdivided into axis-aligned non-intersecting cuboidal regions. The construction algorithm is as follows:

At the root, the point set is split into two subsets of roughly the same size by a hyperplane perpendicular to the x-axis. At the children of the root (depth 1) the partition is based on the y-coordinate and at node depth two, in the next level, on the z-coordinate. Then the algorithm starts over again at the x-coordinates. The recursion stops when there is only one point left, which is then stored at a leaf (see Fig. 6.2).

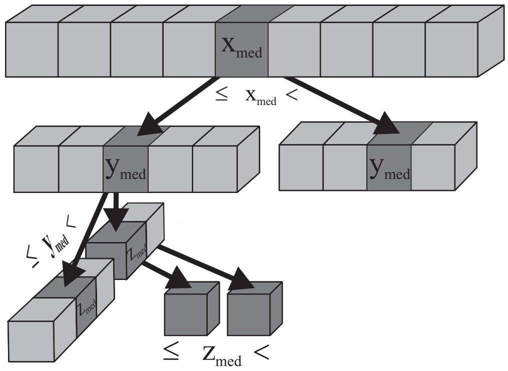


Fig. 6.2: Binary space partitioning using a kd-tree. The subsets over the components of a coordinate (gray) is successively split at its median (dark gray) until only one coordinate is left. The median value itself belongs to the subset with the smaller values.

The splitting plane in each recursion is defined by the median value of the component in a depth. To find the median of a set of values, one can proceed by sorting the set and selecting the central value in the array. This is a process of order $N \log N$. Since the sorting yields much more information than just the median, this procedure is wasteful. The fastest method for finding the median is partitioning [PTVF02], exactly as it was done in the quicksort algorithm [Hoa61, Hoa62]. Selecting a partition element, one marches through the data set, forcing smaller elements to the left and larger ones to the right. But both subsets themselves remain unsorted, and thus its operation count scales as N rather than $N \log N$ for the complete sorting. By definition, the median value belongs to the subset with the smaller values.

Because the three-dimensional kd-tree for a set of n points is a binary tree with n leafs, it uses $O(n)$ storage and the construction time is $O(n \log n)$. The query algorithm visits those nodes whose regions are properly intersected by the query range $[x : x', y : y', z : z']$, and traverses subtrees that are rooted at nodes whose region is fully contained in the query range. At a leaf, a last check ensures that the point fulfills the search criteria. The query time is bounded by $O(n^{1-1/k} + p_r)$, where p_r is the number of reported points [dBvKOS00]. Due to the fact that the data set is recursively subdivided at the central (median) value, the kd-tree is balanced. The depth is then approximated by the round off integer $[\log_2 N]$, which is small even for a lot of points.

To analyze the local neighborhood of a given point, nearby points must be searched. The neighborhood may be a cube, a cuboid or even a sphere. A kd-tree implements orthogonal range queries. Therefore, the range is usually defined as a cube around the given point and the query reports all points within this cube. A query, which is independent from the coordinate axis is implemented by a spherical range, where the maximum distance to the given point is limited by a predefined radius. This is achieved by discarding all points in the bounding cube, whose Euclidean distance is larger than the radius. However, the additional cost for this check is not as high as it may seem, since a validity check to the point at a leaf must be performed in any case. The theoretical kd-tree properties are verified at the example of real data in Table 6.1 to evaluate its practical performance.

model	points	constr. in ms	max. depth	NH 0.1 in ms	NH 0.2 in ms	NH 0.3 in ms	area in mm^2
shoe	404.583	358	20	729	983	1.281	80K
eggcup	423.707	505	19	812	1.324	1.979	12K
boot	724.657	767	21	1.142	1.639	2.222	62K
cube	745.554	864	20	1.895	2.715	3.526	60K
converter	965.983	956	21	1.630	2.218	2.878	114K
duck	998.092	967	20	1.389	2.068	2.733	85K
can	1.076.512	1.159	21	2.268	3.800	5.590	40K
woman1	1.155.575	1.256	21	1.607	2.298	3.122	178K
medal	1.174.931	1.358	21	17.406	34.203	56.785	3K
pepper	1.297.044	1.456	21	2.274	3.659	5.226	44K
cast tile	1.439.192	1.646	21	2.837	4.704	6.920	82K
casting	1.656.358	1.793	22	4.416	7.156	12.042	51K
golf ball	2.152.348	1.472	22	16.509	44.792	84.225	4K
Santa Claus1	2.281.961	2.596	22	3.961	6.310	9.297	116K
Santa Claus2	5.704.530	7.032	23	12.838	24.283	39.676	116K
woman2	5.813.599	7.378	25	11.511	20.520	32.641	178K
Santa Claus3	12.757.244	10.557	25	42.300	94.950	318.509	116K

Tab. 6.1: Performance evaluation for the employed kd-tree. For each model, the number of input points, the construction time for the kd-tree and maximum depth are listed. In the following columns the processing time for performing a neighborhood query in three different spherical regions with radii from 0.1, 0.2 and 0.3 mm is shown. For the rating of low and high processing times, the last column also displays an approximation for the model's surface area. Some objects appear several times, because they were captured in different resolutions or with different scanners. For the evaluation, a standard PC (Pentium 4 3.2GHz, 2GB RAM) was used.

6.1.4 The Range Tree

A further improvement to the query time is provided by range trees. The idea behind this data structure is the following: if a kd-tree performs a one-dimensional range query on each component of a coordinate, then a combined query for all components will increase the performance. Thus, the main tree is a binary search tree on the x-coordinates. Each node v in the main tree has an associated data structure $\mathcal{T}_{assoc}(v)$, which is a binary search tree on the y-coordinates, and again, every node in $\mathcal{T}_{assoc}(v)$ points to an associated binary search tree on the z-coordinates. Such a range tree uses $O(n \log^2 n)$ storage and it can be constructed in $O(n \log^2 n + p_r)$ time. Points that lie in the rectangular query range, are reported in $O(\log^3 n + p_r)$ time, where p_r is the number of reported points (for further reading see [dBvKOS00]).

In practice, the implementation of a three-dimensional range tree requires much more memory, bookkeeping and internal operations, which reduces the effective performance benefits. Therefore, the best compromise was found by the kd-tree. Due to the efficiency with regard to memory consumption, computation time and flexibility it is used as basic data structure for the following algorithms.

6.2 Point Cloud Analysis

Usually, procedures have to know about the point cloud structure and the data distribution in order to adaptively compute the parameters for internal algorithms. As it can be derived from Table 6.1, the time-consumption for (spherical) neighborhood/range queries may increase significantly, even if the search radii are slightly increased. The volume of the sphere, for which neighbors are reported, depends on the radius with $\frac{4}{3}\pi r^3$. By assuming that the surface is locally planar, this region reduces to a circle. Its area depends on the radius with πr^2. Thus, the number of reported points and traversed tree-nodes increases rapidly, in the worst case at least with a quadratic function of the search radius. The worst case is a radius, which is significantly higher than the point density. Thus, the choice of the optimal search radius with respect to computation time is of importance for neighborhood queries in general.

6.2.1 The Optimal Neighborhood

Searching for neighbors is a basic algorithm in point cloud processing. There are two factors that mainly describe a neighborhood for further algorithms. The number of points and the quality of local representation.

Local analyses typically require a certain number of points k to estimate local shape functions (e. g., fitting geometries) depending on the number of unknown parameters. But the number of neighboring points is not necessarily a sufficient criterion, if the considered region is too small or too large and does not represent the surface part. Especially in scanned point clouds with noise and uneven sampling densities, the point number fluctuates locally. Therefore, the point number needs to be coupled to a neighborhood radius, which is chosen adaptively. The radius range should be defined by the data density and expected sizes of the minimal and maximal features that should be detected. Furthermore, the noise level of the considered feature plays an important role. If the ratio between variance and tolerance to the fitted function becomes too large, the approximation will fail.

To estimate surface normals, a local plane is fitted to neighboring points. Therefore, Mitra et al. proposed an adaptive method to compute the optimal radius with respect to the amount of points and their distribution for normal vector estimations [MNG04]. Assuming that noise has zero mean and standard deviation σ_n, they minimize a bound for the estimated angle between the normal vectors of the fitted plane and the true surface with a probability of $1-\varepsilon$. The optimal radius r is obtained by:

$$r = \left(\frac{1}{\kappa}(c_1 \frac{\sigma_n}{\sqrt{\varepsilon\rho}} + c_2\sigma_n^2)\right)^{\frac{1}{3}} , \tag{6.2}$$

where ρ is the local sampling density, κ is the local curvature, and c_1 and c_2 are constants. The algorithm takes σ_n as input and iteratively evaluates r. In the first iteration ρ and κ are evaluated based on empirically chosen $k(=15)$ nearest neighbors and then the radius r is obtained from Eq. 6.2.

Another application is discussed by Ohtake et al. [OBA+03]. They locally fit 3D quadrics and bivariate quadratic functions to construct multi-level partition of unity (MPU) implicit surfaces. These functions exhibit more degrees of freedom, and thus are more suitable to represent curved surface parts. The radius r is adaptively increased until sufficient k neighbors are found to solve the surface equations (10 for 3D quadrics and 6 for the bivariate quadratic polynomials). The initial minimum radius r_{min} is defined as 0.1 mm. Typically, a constant value must be added to k in order to compensate uneven sampling and noise.

Thus, the choice of the optimal radius r depends on the application and what function the local surface part is assumed to have. However, there is another fact that must be considered. Even an adaptive number of points or radii only have a local character. If a small and a large object have the same point density, the optimal local point number is the same, but the covered regions represent different scales of the surfaces (see Table 6.1). A solution would be given by adjusting these parameters with regard to the ratio between the local and global surface area. But in most cases a local analysis must be performed before information on the global behavior can be obtained.

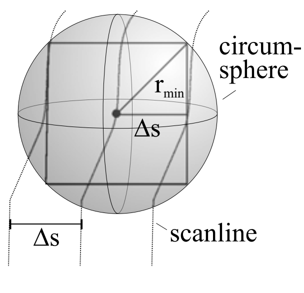


Fig. 6.3: Computing the minimum radius r_{min} for the sphere containing the neighboring scanlines.

For the used 3D scanners, there is some a priori information that can be used to adaptively determine a minimum radius. The movement and displacement between two scanlines are known system parameters, which typically range between 0.1 and 0.5 mm for the examined models. Thus, in any case, the minimum radius r_{min} must at least be larger than the distance between two scanlines Δs. Starting from a given point on a scanline, a cubic region can be defined, which touches the neighboring scanlines. Then, the circumsphere of this region contains a sufficient number of points from the neighboring scanlines (see Fig. 6.3). Thus, the minimum radius r_{min} of this circumsphere is defined by:

$$r_{min} = \sqrt{2}\Delta s + c\,, \tag{6.3}$$

with the constant c to compensate uncertainties (usually $c = 0.1$ mm). For the flexible laser scanner, the point density is usually higher and the scanlines are oriented in an arbitrary manner. In this case, r_{min} is adaptively increased until at least 20 points are covered. The starting value must be defined by a value larger than the uncertainty of the scanner, which is also $c = 0.1\,mm$.

6.2.2 Data Density

As already discussed in the section before, the local point density is an important parameter for applying certain methods to unevenly sampled surfaces. It allows the adaptive adjustment of the search radius on the one hand, and it implicitly provides information on the quality and completeness of a scan on the other hand.

Edges and shadowing effects can cause gaps and holes. Moreover, and unfavorable viewing and projection conditions in the data acquisition stage lead to different sampling densities. The more parallel the laser plane and the surface are, the larger becomes the distance Δs between two scanlines on the surface $\Delta s'$ (see Fig. 6.4(d)). This effect depends on the unknown surface function and the positions of the laser. It especially appears at strongly curved regions, which are hardly accessible for the optical sensors.

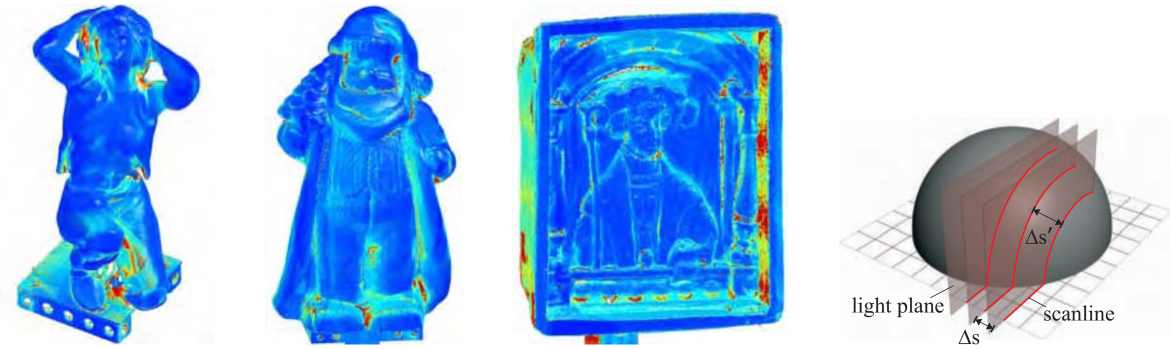
To compute the point density, these problematic surface parts are automatically determined (see Fig. 6.4). In further steps, it is possible to repeat the scan in those areas in order to improve the quality of the scan. The point density is also useful to identify outliers, which are often characterized by a significant low density in their neighborhood (e. g., ≤ 2 neighbors).

For computing the density ρ, the radius r for the neighborhood is computed as proposed in the section before. The number of points within the neighborhood is counted and the resulting number N_p is divided by the area of the local plane within the circumsphere to normalize the result:

$$\rho = \frac{N_p}{\pi r^2}\,. \tag{6.4}$$

where π can be omitted in principle, since it is just a scaling factor.

For the examined models from Table 6.1, the local point density does not vary much in most of all surface parts. Thus, the computation performance can be increased



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

Fig. 6.4: Illustration of the resulting point densities ρ for three different models with Eq. 6.4. The colors indicate a decreasing density from blue to red (a). The influence of the object's geometry on the scanline density is shown in (d).

by estimating the globally best radius. Therefore, the radius of 1% of all points is computed, provided that at least 20 points are covered. Their median value serves as global radius estimation for the entire point cloud.

6.2.3 Surface Normal Estimation

The local orientation of the surface is described by its normal vectors. Since the surface function is unknown in most cases, it must be approximated. Besides polygonal meshing, from which this information can directly be derived, there are different methods to compute the normals from the point cloud.

Hoppe introduced an algorithm, which locally fits planes to the points in a fixed size neighborhood [HDD+94]. The orientation of the plane is assumed as the normal vector for the considered surface point. Mitra et al. proposed a method to adaptively compute the optimal radius for the neighborhood [MNG04] in contrast to a fixed number of neighbors. Additionally, Pauly et al. observed that the fitting plane for a point p should respect the nearby points more than the distant points [PKKG03]. Hence, the neighboring points are assigned weights based on their distances to p by using a Gaussian function.

For an ideal, noise-free point set Amenta et al. proposed a Voronoi-based method for normal estimation [AB99]. For a given set of three-dimensional points P, the Voronoi diagram and its dual Delaunay triangulation of P are computed. They showed that the line through a point p and the farthest vertex in the Voronoi cell can approximate the normal at p. Since this property does not apply for noisy data, Dey and Goswami extended this idea [DG04]. They observed that certain Delaunay balls remain relatively large and can take the role of polar balls. Therefore, they redefine the pole for a point $p \in P$ as the furthest vertex of its Voronoi cell whose dual Delaunay ball is large by comparing the radius of the nearest neighbor distances with a user-defined parameter. In [DLS05], Dey et al. compared the different methods and found out that the last method is the most robust, but also the significantly slowest one. The main reason is that it employs a Delaunay triangulation procedure to the entire point cloud, while the other methods operate very locally.

For the scanned point clouds in this work, a plane fitting is performed, as well. Based on the minimum radius, the corresponding neighbors for a considered point p are determined. The plane fitting procedure is then based on a least-squares orthogonal distance fitting (ODF) as proposed by Ahn [Ahn04]. Therefore, the model parameters are determined by minimizing the square sum of the shortest distance between the model feature and the measurement point. There are different categories of least-squares fitting, i. e., algebraic fitting and geometric fitting. These are differentiated by their respective definition of the error measure to be minimized. A plane which is given by the algebraic equation $ax + by + cz + d = 0$ is an algebraic fit, if its parameters (e.g. the coefficients) are determined in a least-squares optimization. In contrast, the plane is a geometric fit, if the sum of the squares of the distances to the given points is minimal. This procedure is numerically more stable. Therefore, the plane containing the point X_0 and normal vector n can be described as:

$$(X - X_o)^T n = 0 \quad with \quad \|n\| = 1\,. \tag{6.5}$$

The square sum of the orthogonal distances from each measured point $\{X_i\}_{i=1}^m$ to this plane is then defined by:

$$\sigma_0^2 \equiv \sum_{i=1}^{m} [(X_i - X_o)^T n]^2 = n^T \left[\sum_{i=1}^{m} (X_i - X_o)(X_i - X_o)^T \right] n = n^T \mathbf{M} n\,, \tag{6.6}$$

with the central moments tensor as the symmetric square matrix $\mathbf{M}$:

$$\mathbf{M} \equiv \begin{pmatrix} M_{xx} & M_{xy} & M_{zx} \\ M_{xy} & M_{yy} & M_{yz} \\ M_{zx} & M_{yz} & M_{zz} \end{pmatrix}\,, \tag{6.7}$$

where the components of $\mathbf{M}$ are given in relation to the mean values (X_o, Y_o, Z_o) by:

$$\begin{array}{rclrclrcl} x_i &=& X_i - X_o, & y_i &=& Y_i - Y_o, & z_i &=& Z_i - Z_o, \\ M_{xx} &=& \sum_{i=1}^m x_i^2, & M_{yy} &=& \sum_{i=1}^m y_i^2, & M_{zz} &=& \sum_{i=1}^m z_i^2, \\ M_{xy} &=& \sum_{i=1}^m x_i y_i, & M_{yz} &=& \sum_{i=1}^m y_i z_i, & M_{zx} &=& \sum_{i=1}^m z_i x_i\,. \end{array} \tag{6.8}$$

In the next step the matrix $\mathbf{M}$ is decomposed by using the singular value decomposition (SVD) of matrices. The SVD method is based on the following theorem of linear algebra: Any $M \times N$ matrix $\mathbf{A}$, with $M \geq N$, can be written as the product of an $M \times N$ column-orthogonal matrix $\mathbf{U}$, an $N \times N$ diagonal matrix $\mathbf{W}$ with positive or zero elements (the singular values), and the transpose of an $N \times N$ orthogonal matrix $\mathbf{V}$ (for further reading see [PTVF02]). Since $\mathbf{M}$ is a symmetric 3×3 square matrix, the decomposition results in:

$$M = \mathbf{V_M} \mathbf{W_M} \mathbf{V_M}^T\,. \tag{6.9}$$

As a result, the diagonal matrix $\mathbf{W_M}$ contains the principle central moments, and the orthogonal matrix $\mathbf{V_M}$ contains the principle axes of central moments. The fitting plane is finally defined by the mass center $\bar{X}$ and the principle axis $\mathbf{v_{Mj}}$ with the corresponding smallest moment $\mathbf{w_M}$. Specifically:

$$(X - \bar{X})^T \mathbf{v_{Mj}} = 0\,. \tag{6.10}$$

Since $\mathbf{W_M}$ contains only three values (on its diagonal), the smallest one is easily found, and the corresponding column in $\mathbf{V_M}$ can be extracted.

Solving this linear least-squares problem by using the SVD method is numerically the most reliable, especially for ill-conditioned matrices, where elements in the matrices are much smaller than others (for further discussion see [NW99]).

The orientation of the estimated normals from the plane fitting depends on the position of the plane within the coordinate system. For the analysis and for an esthetic visualization a consistent orientation is necessary. By exploiting given scanline information of the camera or the laser position, the normal vector orientation problem becomes trivial. An approximated normal $\vec{n}$ for a point p only needs to be flipped, if it points contrary to the camera or laser (p_c/p_l). Specifically:

$$\vec{n} = -\vec{n}, \quad \text{if} \quad \vec{n} \cdot \frac{p_c - p}{\|p_c - p\|} < 0\,. \tag{6.11}$$

6.2.4 Graph Representations for Unstructured Point Sets

Without additional system information, usually a global data structure is necessary that encodes the neighborhood to ensure that neighboring tangent planes are consistently oriented. Although the scanning system provides the necessary information, this problem illustrates an often recurring application for neighborhood algorithms.

The problem can also be described as follows: Given the sample point set P, sought is a set of consistently oriented normals. Therefore, any two neighboring data points p_i and p_j are taken. If the unknown surface is smooth, and if the sample set P is dense, nearby points will have tangent planes $\mathcal{P}_t$ that are close to being parallel, i. e., if $\mathcal{P}_t(p_i) = (o_i, \vec{n}_i)$ and $\mathcal{P}_t(p_j) = (o_j, \vec{n}_j)$ then $\vec{n}_i \cdot \vec{n}_j \approx \pm 1$. Since the orientation of the tangent planes should be consistent, either $\vec{n}_i$ or $\vec{n}_j$ needs to be flipped, if $\vec{n}_i \cdot \vec{n}_j \approx -1$. Additionally, $\mathcal{P}_t(p_i)$ should be consistently oriented with all neighboring tangent planes.

This normal orientation problem can be modeled as a graph optimization. The graph contains one node N_i for each tangent plane $\mathcal{P}_t$, and N_i will have edges to all nodes that correspond to neighboring data points of p_i. The cost on an edge $E(i, j)$ is defined as $\vec{n}_i \cdot \vec{n}_j$, which is maximal for parallel normals. Thus, tangent plane orientations are needed that maximize the total cost of the path. Since it has been shown that this kind of problem is NP-hard (see Kruskal [Kru56]), an approximation must be used.

Therefore, the surface is assumed to be single connected. A reasonable starting point is to construct the Euclidean Minimum Spanning Tree (EMST) for the set of tangent plane centers $\{o_1, o_2, ..., o_n\}$ (in this case o_i corresponds to p_i). An edge $E(i, j)$ is added to the tree if either o_i is in the neighborhood of o_j or vice versa. Specifically, the Euclidean minimum spanning tree of the point set P is the maximal tree $EMST(P) = (P, E)$ such that $E \subseteq P \times P$ and the sum of all edge lengths $\sum l(e_k)$ is minimum, where $l(e_k) = |p_i - p_j|$ (early algorithms considering the EMST were also discussed by Prim [Pri57] and Yao [Yao75]). Thus, the constructed graph is a connected graph that encodes geometric proximity in the Euclidean norm in $\mathbb{R}^3$ of the tangent plane centers o_i. Such a graph is also called a Riemannian Graph [HDD+94] (see Fig. 6.5).

(a) (b) (c)

Fig. 6.5: Achieving a consistent orientation of the initial normal vectors (a) by flipping the normals (b) depending on their parent nodes in the Riemannian Graph (c). An enlarged segment of the graph with highlighted main path in red and its subgraphs in blue (b). Some seemingly overlaying subgraphs in the image are due to the projection from 3D to 2D image space.

After constructing the Riemannian Graph, a heuristic to iteratively propagate the tangent plane orientation is needed. If the propagation is solely based on the geometric proximity, the resulting surface can be severely distorted due to noise and uneven density. Particularly at sharp edges, where significant changes of neighboring tangent planes may appear, this approach can fail. Thus, the best way is to propagate the orientation along directions of low curvature. Therefore, a cost $1 - |\vec{n}_i \cdot \vec{n}_j|$ is assigned to the edge $E(i, j)$ in the Riemannian graph, which is always positive and low, if the tangent planes $\mathcal{P}_t(p_i)$ and $\mathcal{P}_t(p_j)$ are nearly parallel. Then, the propagating procedure traverses the Euclidean Minimal Spanning Tree of the resulting graph. The starting position for the propagation must be a point with known correct tangent plane orientation. Since such a position is not known, a possible approximation is to take the tangent plane center with the largest z-coordinate, and to assign the $+z$ direction to the corresponding normal vector. During traversal, each node is assigned an orientation consistent with that of its parent node, i. e., if $\mathcal{P}_t(p_j)$ is the next tangent plane for $\mathcal{P}_t(p_i)$. Then, the direction of $\vec{n}_j$ is reversed, if $\vec{n}_i \cdot \vec{n}_j < 0$.

There are two important other graph representations for a point set P which were discussed by Attene and Aurenhammer [AS00, Aur91]. The first one is the Nearest

Neighbor Graph (NNG), which is the maximal graph $NNG(P) = (P, E)$ so that the edges $E \subseteq P \times P$ and p_j is the point of P closest to p_i. A similarly defined construct is the Gabriel Graph GG. Two points are called Gabriel points, and are connected by an edge E if the sphere having this edge as diameter is empty. Thus, the Gabriel graph of P is the maximal graph $GG(P) = (P, E)$ defined by $E \subseteq P \times P$ and the smallest sphere for p_i and p_j does not contain any other point of P. Aurenhammer also noticed that the Gabriel Graph just consists of those Delaunay edges that intersect their dual Voronoi edges.

6.3 Point Cloud Optimization

Repeated scans in the same area of an object's surface and the choice of the sensor alignment can cause overlaying point clouds. The introduced redundancy can be helpful to locally optimize the point cloud on the one hand. But on the other hand, a large amount of points significantly reduces the processing speed of further algorithms. This section discusses suitable methods to process the redundant information in order to optimize and simplify a point cloud with respect to quality and curvature information as derived in the chapters before.

6.3.1 Adaptive Smoothing

As discussed in the data acquisition section, rough and specular surfaces can cause high-frequent noise for optical sensors. Furthermore, the local point distribution variates depending on the shape and the distance to the sensors. By applying smoothing algorithms, this variance can be reduced. Overlaying scans can also produce errors which origin from an imprecise calibration of the sensors to the axes movements. For an aesthetic visualization and for more robust post-processing algorithms, a smoothing procedure must be employed.

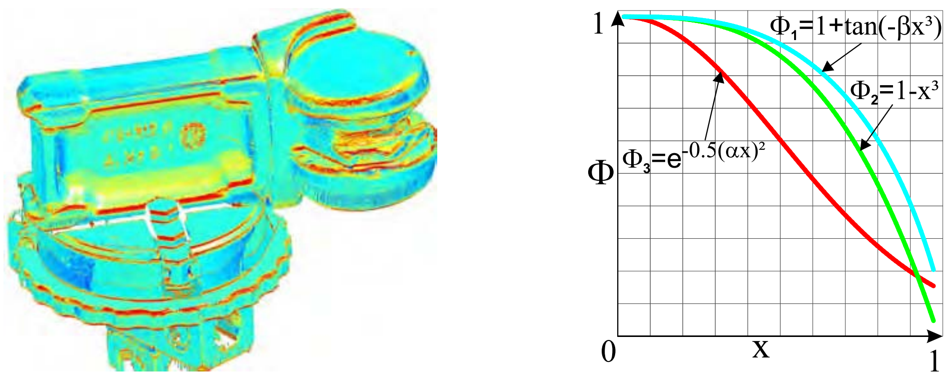
When smoothing noisy data, edges should be preserved and the quality of a point, regarding to its viewing conditions, should have a notable influence on the resulting point. The smoothing procedure locally operates in a neighborhood defined by the k-nearest neighbors (typically k=20) and the minimal radius r_{min}, depending on the scanline distance Δs, as proposed in the section before. As long as the number of points in the neighborhood is smaller than k, the search radius r is increased, starting from r_{min}. A smoothed point is derived from all of its neighbors by applying a weighting function that depends on the distance of the neighbor to the considered point d_i and a weight ω_i, regarding the scanline curvature κ_i and quality of the viewing conditions q_i (see Fig. 6.6(a)). Since a low quality can cause noise and thus a higher curvature, the weight ω_i is defined as the normalized sum of these measures by:

$$\omega_i = (1-\alpha)\bar{\kappa}_i + \alpha(1-\bar{q}_i) \quad \text{with} \quad 0 \le \alpha \le 1\,, \tag{6.12}$$

where α allows to manipulate the ratio between the influence of the viewing quality (viewing and projection angles) and the edge values (typically $\alpha = 0.5$). $\bar{\kappa_i}$ and $\bar{q_i}$ are the normalized values of κ_i and q_i with:

$$\bar{q_i} = \frac{q_i}{\pi}, \quad \bar{\kappa_i} = \begin{cases} \frac{\kappa_i}{\tau}, & \text{if } \kappa_i < \tau \\ 1, & \text{otherwise}. \end{cases} \tag{6.13}$$

This normalization is based on the following observation: Since the viewing quality q_i is defined by the viewing angle, its range is limited between 0 and π. The scale τ defines the curvature value, that indicates significant edges. It is an empirical measure, and for the scanline curvature based on 4^{th} order NURBS curves it was found out that $\tau = 0.2$ is optimal (see Sect. 4.3.1).



(a) (b)

Fig. 6.6: Illustration of the weights for single points based on their scanline curvature κ_i and the value q_i for the viewing angle (a). The smoothing effect at the edges (in red) is lower than in planar areas. The paths of three influence functions Φ are shown in (b). The empirically chosen parameters $\alpha = 0.2$ and $\beta = 0.7$ guarantee an influence between 5-20% at the edge of the neighborhood (x=1).

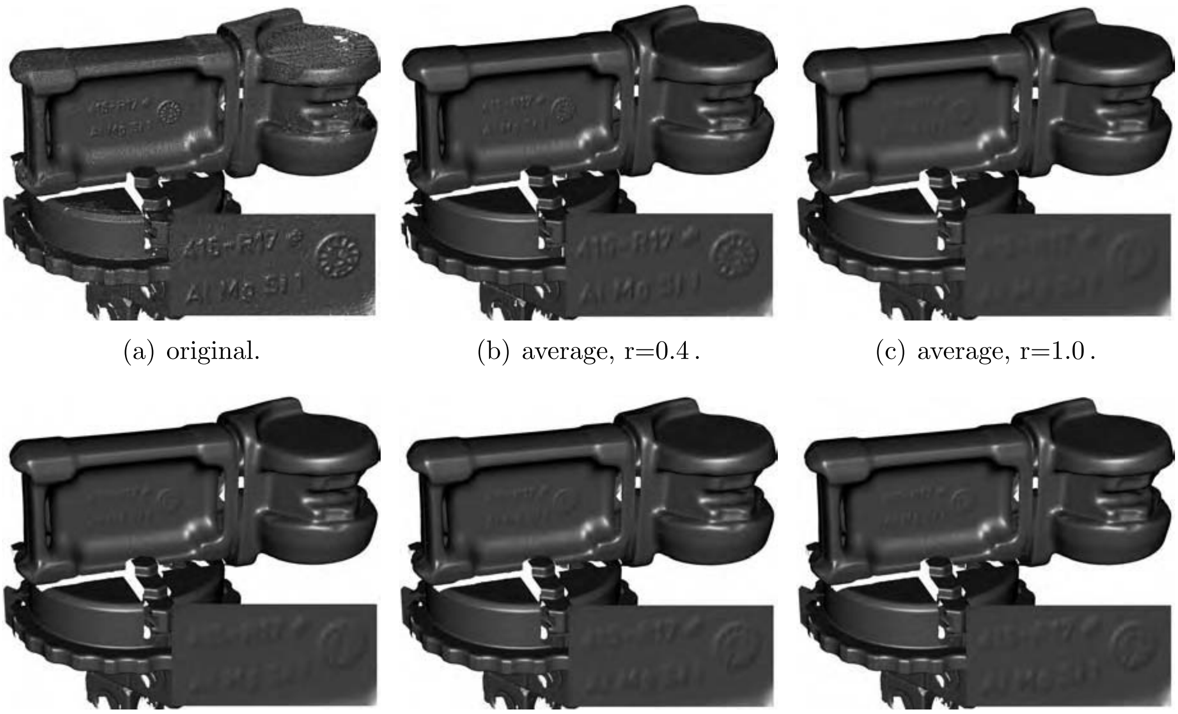
After having defined a weight for each neighbor, an influence function Φ is added. Since the number of points within the bounding sphere of the neighborhood nonlinearly increases with the radius r, the influence function should penalize points near the edge of neighborhood more than nearby points. Based on the function Φ, its neighbors p_i and their weight ω_i, the resulting smoothed point p_s is computed by:

$$p_s = \frac{1}{w_{sum}} \sum_{i=1}^{n} p_i \omega_i \Phi(\bar{d_i}), \quad \text{with} \quad w_{sum} = \sum_{i=1}^{n} \omega_i \Phi(\bar{d_i}), \tag{6.14}$$

where the measure $\bar{d_i}$ is the adaptively normalized distance d_i to the neighbor p_i. Since the maximum for d_i is limited by the radius r of the neighborhood, $\bar{d_i}$ is defined as:

$$\bar{d_i} = \frac{d_i}{r} \quad \text{with} \ \ d_i = |p_i - p|. \tag{6.15}$$

Together with the average function, three different nonlinear influence functions $\Phi_1 - \Phi_3$ were applied in order to attain the desired result. The functions and their paths are illustrated in Figure 6.6(b).



(a) original. (b) average, r=0.4. (c) average, r=1.0.

(d) Φ_1, r=1.0. (e) Φ_2, r=1.0. (f) Φ_3, r=1.0.

Fig. 6.7: Triangulated models to visualize the effect of smoothing a point cloud in different stages. The original noise point cloud (a) and the smoothed representation based on an average filtering (b). The result of the quality and edge weighted approach is shown in (c). The slightly better results by applying the influence functions Φ_i from Figure 6.6(b) are shown in the bottom row.

Different radii between 0.4 and 1.0 mm were applied to evaluate the weighted smoothing. The effects are shown at the example of the casting model in Figure 6.7. The automatically determined radius r_{min} was 0.4 mm. In this case, no significant difference between the weighted and the average function was found, since the covered region was (optimally) small. But for a radius of 1 mm the weighting effect becomes visible. The influence functions slightly differ in the influence of very near or far points.

In summary the curvature and quality-based weighting with an influence function performed well and exhibited a significant improvement compared to the simple average, since edges are retained while more planar regions are smoothed. Due to its strong slope, the exponential function(Φ_3) operates more locally, and thus gives more influence to nearby points than the tangent function (Φ_1). The best compromise with regard to the computation time is given by the less complex cubic function (Φ_2).

In order to control the smoothing, a test function is additionally applied, which checks if the distance of a smoothed point p_s to its original p exceeds a tolerance t. The value of t depends on the measuring uncertainty or the accepted inaccuracy (e. g., t=0.1):

$$p_s = p + t \cdot \frac{p_s^* - p}{\|p_s^* - p\|} \,. \tag{6.16}$$

Without having scanline curvature, LANGE ET AL. propose a method for point cloud fairing using an anisotropic geometric mean curvature flow [LP05]. Their method solves a parabolic PDE with boundary constraints to obtain an anisotropic Laplacian operator. Unfortunately, this approach requires many iterations and a user-defined parameter called edge quotient that enables to emphasize corners. Furthermore, based on the given normal vectors of a point cloud, a smoothing is also achieved by generating an implicit volume model whose zero level isosurface interpolates the given points and associated normal vectors [Nie04b].

In addition to point cloud smoothing, there are also many smoothing algorithms for polygonal meshes. These methods benefit from the known local edge connections, e. g., to relax the polygons [BKP07, Gib98]. Furthermore, NEALEN ET AL. introduce a framework for triangle shape optimization and feature preserving smoothing of triangular meshes that is guided by the uniformly weighted Laplacian and the discrete mean curvature normal [NISA06]. A comparative overview on polygonal mesh smoothing is given in [BHP06].

6.3.2 Adaptive Correction

The quality of laser scanned 3D point clouds is mainly determined by the direction of projection and the viewing direction of the camera onto the object's surface. This fact can be exploited in order to adaptively remove redundant information. After registering the point clouds from different scan operations and sensors, the same small neighborhood region N often has been multiply sampled and contains sample points in different quality (see Fig. 6.9). Points of lower quality downgrade the influence of points with high quality when applying neighborhood-based operations to these regions. Therefore, low quality points should be removed from the merged point set.

To minimize the number of points that are removed, the minimal neighborhood radius r_{min} should be selected for merging. Useful definitions for r_{min} are either based on the distance between points or two scanlines Δ_s (Eq. 6.3) or the expected uncertainty of the 3D scanner σ_M. For the analyzed point sets, Δ_s is larger than the measuring uncertainty in most cases, which causes relatively large neighborhood, and thus the removal of too many points. The uncertainty (0.1 mm) is more suitable for this purpose, but may be too small if Δ_s is large. A further adaptive measure is given by determining the typical (average) distance of two points on the considered scanline Δ_p. Since the directions of all three measures are different, they are interpreted as a vector whose length is the radius r_c for the neighborhood in which the correction is performed.

$$r_c = \sqrt{\Delta_p^2 + \frac{\Delta_s^2}{4} + \sigma_M^2} \tag{6.17}$$

For the normalized quality values q_i of all points p_i in the resulting neighborhood $N(r_c)$, the average $\bar{m}_q$ is computed. This value serves as a threshold which defines that

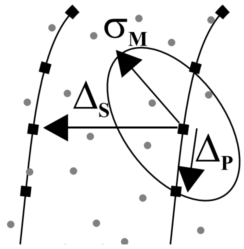


Fig. 6.8: Illustration of the distance measures used to compute the optimal radius in the correction procedure.

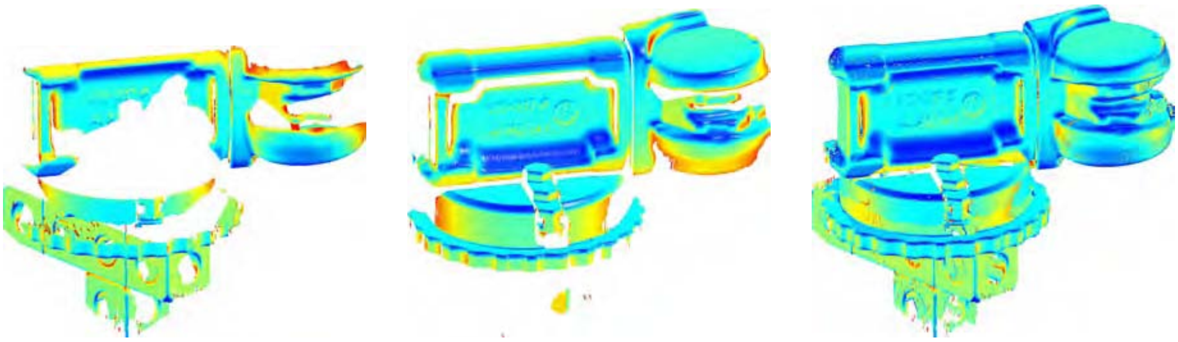
Input: sets of 3D points P from different scan operations
Output: merged and reduced point set
foreach *point* p_i **do**
  if p_i *is not already removed* **then**
    calculate Neighbors(p_i,N,r) in P
    calculate Quality(n)
    foreach *neighbor n from N* **do**
      if *Quality(n)<averageQuality(N)* **then**
        remove(n)
      end
    end
  end
end

Algorithm 6.1: Algorithm for the quality-based merging of point sets from different scan operations.

a point p_i in $N(r_c)$ should be removed, if its quality is lower than the threshold $\bar{m}_q$. Specifically:

As a result, the redundancy from the regions N is avoided by removing low quality points (see Fig. 6.9(c)). For the flexible laser without constant distances between the scanlines Δ_s is set to zero.

It was also noticed that an adaptive correction for points with low quality on the basis of neighboring high-quality points by weighting is not reasonable. On the one hand, the uneven sampling would still remain and on the other, the necessary low weight for the considered point causes only a weighted interpolation and smoothing between the neighbors with negligible influence of the point itself.



(a) (b) (c)

Fig. 6.9: Quality-based merging of redundant surface parts from different scan operations and sensors. The data sets obtained from the lower and upper sensor with their corresponding viewing quality are given in (a) and (b). The merged result is shown in (c).

6.3.3 Adaptive Simplification

Multiple scanning of the object's surface is often necessary to assure the capturing of all interesting surface parts. This often results in very large point clouds, with no significant increase of information due to a higher density at overlaying regions. After having merged overlaying points, the point density usually is still very high. In order to increase the computation performance of the following algorithms, a point-based simplification is applied. Usually, there is a differentiation between uniform and adaptive non-uniform procedures. To ensure a constant, uniform distance d_u between neighboring points, all points in the neighborhood of a point p with $r = d_u$ are removed (see Fig. 6.10(a)). The advantage of this approach is the high computation speed, but its disadvantage is that it does not regard local surface properties. But especially this adaptivity allows to remove more points in planar regions, than at edges and in curved regions.

For visualization purposes, PAULY ET AL. presented an iterative method [PGK02]. They compute the local surface variation obtained from a covariance analysis of the k nearest neighbors. From the eigenvalues λ_i, derived from the eigenvectors of the covariance matrix, they determine the corresponding normal vectors and achieve a consistent orientation with the procedure described in Section 6.2.4. The surface variation σ for a point p is then defined by λ_0 as the deviation from the plane, spanned by the mass center of the neighborhood (with size n) and the normal vector.

$$\sigma_n(p) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{6.18}$$

For example, a zero deviation $\sigma_n(p)$ indicates that all points in the neighborhood of p lie in a plane, and if all eigenvalues have the same length, i. e., $\sigma_n(p) = 1/3$, a completely isotropically point distribution can be assumed. Points that cause the smallest error, are removed from the set by an edge collapse operation in an iterative manner.

When using point cloud simplification as preprocessing for surface measurements, the input is a desired distance value d_a between two neighboring points that is only allowed to decrease in strongly curved regions and at edges. The necessary local surface properties for the adaptivity of the approach is then given by the curvature measures κ, that have been efficiently derived from the scanline analysis (see Fig. 6.10(b)). Therefore, for each point p of the point set P all neighbors in the neighborhood d_a are identified with the help of the kd-tree. For each p_i, its value $\bar{\kappa_i}$ from Eq. 6.13 is used to determine the linear scale s_r that describes how much points must remain in the neighborhood of p_i, whereas p_i itself is never removed:

$$s_r = N_p(d_a) \cdot \bar{\kappa_i}\,. \tag{6.19}$$

If s_r is always set to zero, all points except p_i are removed and the procedure equals a uniform simplification. Otherwise, points with low quality $\bar{q_i}$ are removed first. To ensure that edge points are not removed by processing planar neighborhoods, the procedure is applied to the edge points first. This is achieved by dividing the point set into two subsets containing significant edge points on the one hand, and all remaining points on the other hand. The computation performance is increased, since points are not deleted from the kd-tree but labeled as removed. Although the tree becomes unbalanced, the whole procedure is much faster than re-balancing the tree by removing a point. At the end of the procedure, the tree is simply restored by unlabeling the knots.

(a) (b) (c)

Fig. 6.10: Point cloud simplification with r=0.5. The result of the uniform simplification is shown in (a), the scanline curvature (b) is used for an adaptive simplification (c). For a better visualization, illustration (c) shows the resulting density in gray levels instead of single points.

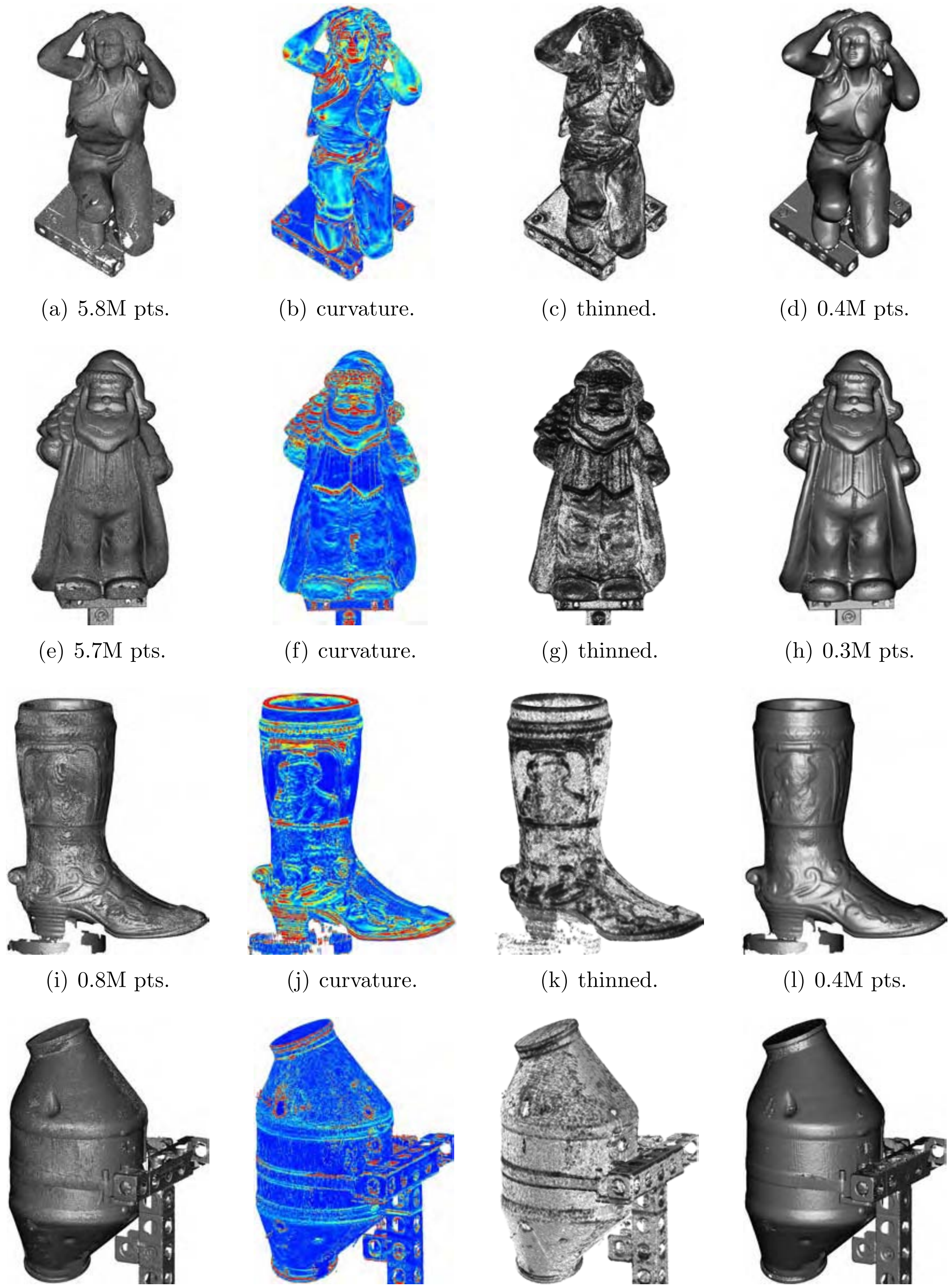
6.4 Case Studies

The proposed methods were applied to different models in order to evaluate their effectiveness. Figure 6.11 illustrates the models and the processing pipeline. The polygonal approximation of the initial point sets in the first column shows the influence of noise, redundancy and uneven sampling. The most problematic case is an overlay of noisy point sets, since the local surface is then represented by multiple point layers. This effect is significantly reduced by the correction step, which solves the redundancy by removing low quality points in local neighborhoods. The correction is additionally supported by a following smoothing procedure, which is adapted to the scanline curvature (second column) to preserve edges while smoothing noise. Since the smoothing performs a weighted averaging, an uneven sampling is implicitly corrected, too. In order to increase the computation speed of the following local point processing operations, the number of points has been adaptively reduced. The models in Figure 6.11 also show different kinds of curvature of small and large areas with sharp and more smooth edges. Since the simplification procedure is also based on this curvature information, it reduces the point number depending on the strength of the curvature (third column). The point clouds processed this way show a significantly increased quality of their polygonal approximation (last column), although the point density is also significantly reduced.

6.5 Summary

In this chapter, different strategies for the management, analysis and processing of point clouds derived from 3D scanners were presented. The methods are fast and robust and adaptively derive their parameters from the data, without requiring a tesselation of the underlying surface. The employed kd-tree only stores pointers to the complex scan data structures, which is memory efficient and provides immediate access to the additional system information generated during the scan process. The smoothing and point-based simplification can significantly increase the performance when creating polygonal approximations of large data sets. Because on the one hand, the number of points to be processed is reduced, and on the other the smoothing operations reduce topological distortions due to noise. The presented methods also benefit from utilizing the given system information, like quality measures, uncertainty estimations, and scanline distances to increase the degree of automation and their adaptivity.

In the last three chapters, modular tools for managing and evaluating 3D point sets were discussed. These methods employ as much structural information from the acquisition principles as possible to increase the efficiency and adaptivity. Their application to real measuring tasks is discussed in the next chapter.



(a) 5.8M pts. (b) curvature. (c) thinned. (d) 0.4M pts.

(e) 5.7M pts. (f) curvature. (g) thinned. (h) 0.3M pts.

(i) 0.8M pts. (j) curvature. (k) thinned. (l) 0.4M pts.

(m) 1.0M pts. (n) curvature. (o) thinned. (p) 0.3M pts.

Fig. 6.11: Case studies for the proposed methods. The first column displays the triangulated models of the initial raw point clouds. The second one shows the scanline curvature of the merged point clouds from different scan operations/sensors. The third shows the density after the adaptive simplification (r=0.5) and the last column points out the triangulated models of the smoothed and simplified point clouds.

Chapter 7

Practical Applications

Mass production of industrial workpieces requires continuous manufacturing control to ensure constant quality. Therefore, assembly inspection is used to detect errors at an early stage. Many conventional inspection methods are still in use, especially in the automotive and supplier industry. Traditional inspection with gauges is rather subjective and time-consuming. Moreover, such inspection only reaches an OK or not OK decision and measurement of actual dimensions is impossible. Furthermore, it is also impossible to specifically control and improve the manufacturing process. Therefore, measurement systems were developed that reliably inspect the geometric metrics of a manufactured item based on given reference data. The systems presented have been developed in cooperation with other people at the Fraunhofer IFF and manufacturers. The modular scan data algorithms introduced in the previous chapters were integrated to evaluate the data with respect to the measuring tasks.

This chapter gives a brief overview on two exemplary industrial applications and the measuring systems, where the methods presented in the previous chapters have been successfully applied. The first system captures and evaluates the geometry of catalytic converters automatically within an offline process. The second system evaluates a large number of geometric measures at wheel rims within an online procedure, i. e., during the manufacturing process. Both systems employ line-based laser triangulation sensors and multi-axis locomotor systems to obtain the 3D measures.

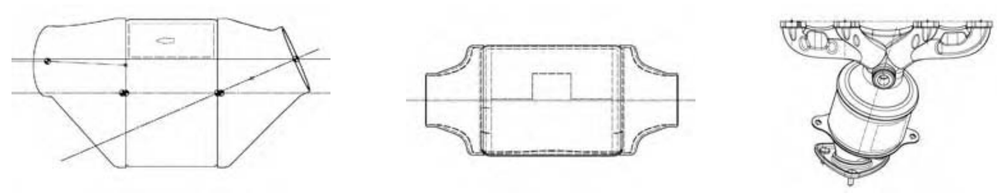
7.1 Geometry Evaluation of Catalytic Converters

Since January 1993 all new cars in Germany are equipped with catalytic converters to reduce pollutant emissions. Twelve million cars with converters are produced in Europe every year. Reduced space in the engine compartment and the underbody make the exact outside geometry of a converter very important. This geometry determines the spatial curve of the connected exhaust pipes, and thus the mounting of the exhaust system in a car. Therefore, quality assurance is an important task, and a significant quality criterion is compliance with geometric metrics. Variations detected can be directly used to correct the production.

This section describes the measuring system and the corresponding methods for automated three-dimensional geometry inspection. This example is used to illustrate the process of 3D data acquisition and evaluation under industrial working conditions. Problems that arose during measurement are analyzed, and suitable solutions are proposed. Problems include outside influences as well as other sources of error, e. g., converter materials or the measuring principle itself. Furthermore, several system parameters are identified that can be used in particular to qualitatively evaluate and correct the point clouds as proposed in the chapters 1 and 2.

7.1.1 Functional Description

A converter consists of a stainless steel shell and a catalytic substrate. The shell consists of three main parts: a cylindrical centerpiece with the catalytic substrate and two funnels to connect the exhaust pipes. The funnels are welded to the centerpiece. Different types of converters are illustrated in Figure 7.1(a). The position and orientation of the funnels relative to the centerpiece are sought. The measuring system has to be able to digitize converters with a total length of 600 mm without contact. The system's measuring uncertainty should not exceed 100 μm. The system has to extract the spatial coordinates of the inlet and the outlet funnels from the measured 3D point cloud.

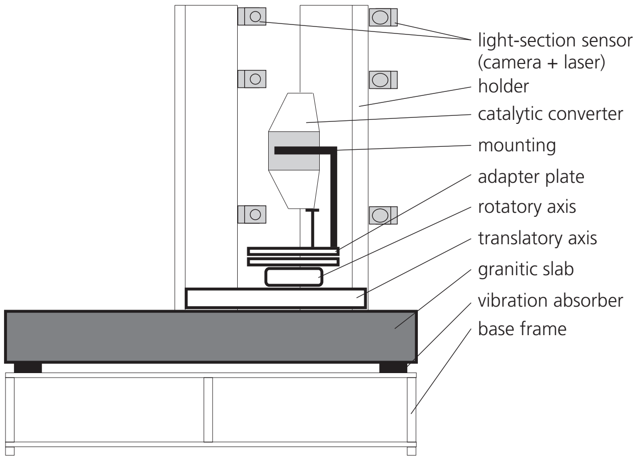


(a) (b) (c)

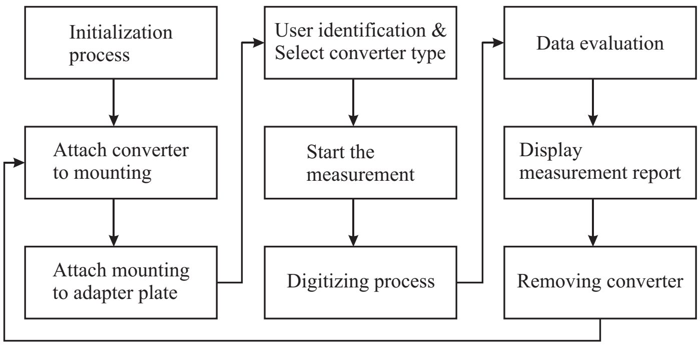
Fig. 7.1: Different types of converter geometries. The principle constituents are the inlet and outlet funnels and a cylindrical center piece. In contrast to the standard shapes in (a) and (b), a more complex type with several attachments is shown in (c).

In addition to the complex measuring system (recall Sect. 3.1.1 and Fig. 3.1(a)), this extension consists of three sensors. The system also uses structured laser light and the triangulation principle to capture the 3D shape of a test item (recall Sect. 2.1.2). Additionally, a multi-axis locomotor system moves the item in front of the sensors to capture the entire surface. The sensors are rigidly mounted together with the locomotor system on a hard stone slab (granite) with passive vibration absorbers (see Fig. 7.2(a)) in order to tolerate ambient influences. A housing protects sensitive components against dust, external mechanical influences and incidence of outside light.

In order to capture the (open) inlet and outlet funnels, they are sealed with a cylindrical endpiece. A mounting, specifically for each converter, makes sure the converter remains clamped during digitizing.



(a)



(b)

Fig. 7.2: Schematic configuration of the converter measurement system (a) and the schematic measuring procedure (b).

Digitizing and measuring operations are performed in a defined order (see Fig. 7.2(b)). Afterwards, a measurement report is generated and used by a machine operator to adjust production. The report is additionally stored in a database to log the development of values. Apart from setting and removing the converters, no other user-interaction is needed.

7.1.2 Data Analysis

The result of the digitizing process is a set of 3D points captured from the shell of the converter. This passage describes the extraction of the geometric properties and features. The data evaluation has to complete two tasks: aligning the converter with the reference coordinate system (e. g., the car coordinate system), and determining the positions of the inspected features. For purposes of simplification, a predefined CAD dataset for any type of converter stored in a database is used. It includes information on the possible feature positions, dimensions to be inspected, and coordinate system alignments specific for each converter.

The features that have to be inspected are: the center points of the inlet and outlet and the positions of boreholes, fastening bolts and plane areas in flanges and brackets. Boreholes and threaded bolts are marked with measuring adapters for more accurate and robust determination. These adapters are spherically or cylindrically shaped and represent the center of the holes and the axis of the bolts, respectively. In an automated search, all defined features must be identified and aligned in the reference coordinate system. Afterward, the acquired geometry data is compared to the nominal values.

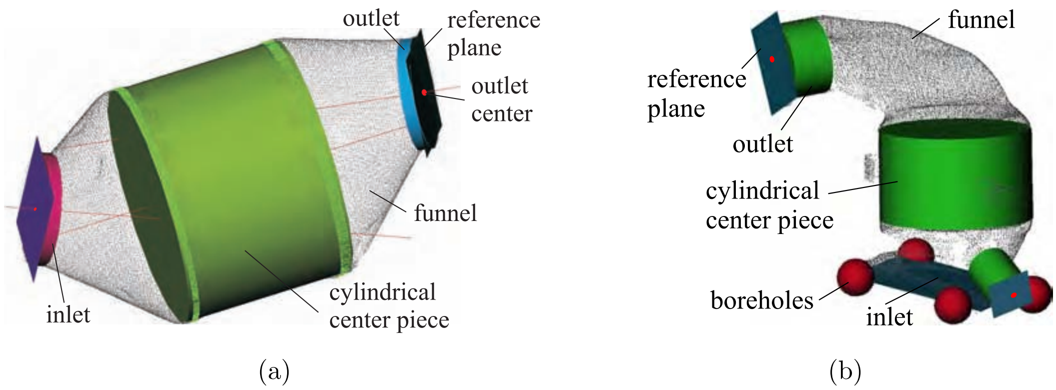


Fig. 7.3: Extraction of the geometric properties to be verified (a) and (b). The open funnel outlets are sealed with endpieces to capture their reference plane. Boreholes are sealed with hemispheres, which are measured and evaluated as spheres (b) whose center represents the borehole center.

A simple example briefly explains the automated procedure. Using the database information, all relevant inspection features are separated from the digitized 3D point cloud of the converter. Taking Figure 7.3 as an example, the circular bordered planes are identified on the inlet and outlet, and a cylindrical centerpiece. The positions of the planes are corrected by the thickness of the cap. Afterwards, the sections are measured using the cylindrical segments of the inlet and outlet. The center points of the circles projected on each of the planes are the reference points for this converter. In this case, the measured data is aligned by the axis of the cylindrical centerpiece that constitutes the z-axis of the reference coordinate system. The remaining two degrees of freedom around this axis are defined by the reference point on the inlet. The measurement report finally provides the distance of the reference points in z-direction and their positions in x and y.

Automated feature detection is an extremely important aspect of this approach. To increase the level of automation, geometries should largely be identified automatically. The following section describes a method for the efficient detection of simple geometries using system parameters and characteristics specific to the measuring principle.

7.1.3 Automated Feature Detection

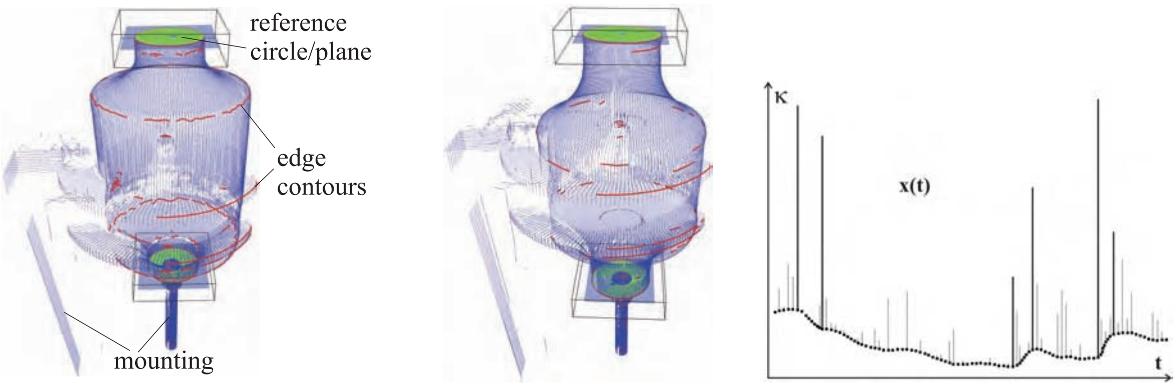
Different converter types, and thus different shapes and surfaces limit the automation of the process of data acquisition described. Algorithms are needed that automatically detect the geometric features mentioned in Section 7.1.1. The measuring principle and the system parameters can be used to efficiently detect such features within the generated 3D dataset.

The three-dimensional points generated by each sensor are sorted based on the measuring principle. This approach uses a laser that projects a thin line. As a result, all captured surface points lie on the line distorted to a 3D curve in relation to the topology of the surface. Thus, every position of the locomotor system has a curve consisting of 3D points. Additionally, all points lie within a 3D plane known from the calibration (recall Sect. 3.4). Each scanned line (scanline) is approximated by B-spline curves as proposed in Section 4.1.1. These curves are used as analytical description for further analysis. Due to the irregular point distributions, an adapted knot vector with values based on the chordal distances was used.

Geometric features of a surface usually have sharp edges. Thus, the next step is to determine these edges. Edge points are characterized by high curvatures κ. The strongest κ values denote edge points that are further evaluated. In Section 4.3.1 it was found out that $\kappa = 0.2$ reliably indicates these edge positions. Therefore, all edge points per scanline are determined. The known sorting of the scanlines is then used to successively connect neighboring edge points from scanline to scanline. Edge points are automatically connected if they are neighbors within the radius $r_E = 0.5\,\mathrm{mm}$, which was empirically ascertained. A set of neighboring, connected edge points is marked as an edge contour for further processing (see Fig. 7.4(a) and 7.4(b)).

The measuring task requires to determine the positions of the circular areas of the inlet and outlet. By CAD definition, all points within these areas lie on a plane. Both features are used to automatically detect inlet/outlet locations within the point cloud. Therefore, each of the identified edge contours is evaluated if a good fit of circular curve can be achieved. This is done by fitting a circle in $\mathbb{R}^3$ [ARR99] to those points and evaluating the standard deviation and the expected radius stored in the CAD file. In the next step, the resulting set of circular objects is projected back to the original point cloud, and points within the plane and circular area extracted. The fitted plane (recall Sect. 6.2.3) yields the spatial orientation. Additionally, the sought funnel position is obtained from the center of the circle (see Fig. 7.4). Based on this information, the object coordinate system of the measurements is aligned with that of the CAD data. In the following steps, the positions and dimensions stored in the CAD file are employed to easily identify the other areas to be inspected.

This approach can automatically and efficiently determine the correct positions in more than 95% of the converter types. For the other cases, expectation ranges and complex heuristics must be manually stored in the database.



(a) (b) (c)

Fig. 7.4: Principle of an automated curvature-based feature detection (see red contours) using the examples of two catalytic converters (a) and (b). The circular regions marked in green indicate the sought reference planes. Points with a curvature $\kappa > 0.2$ (thick lines) denote significant edges (c).

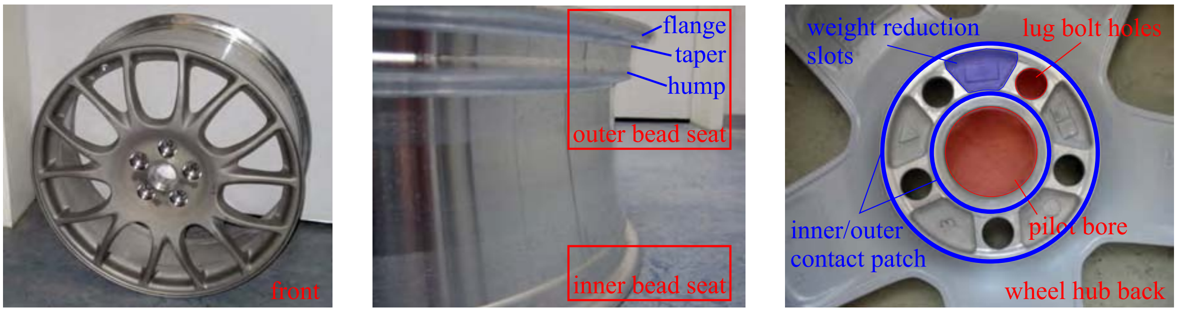
The identification of the geometric primitives requires a preprocessing step. The usually very large point sets have to be reduced without appreciable loss of information. Additionally, artifacts and noise influence processing with the proposed methods. The B-spline description of a scanline can be used to drastically reduce the point cloud. Usually, the scanner produces point sets with a density of 0.02 mm within each scanline. During the development it was found out that a density of 0.1 mm is sufficient. Furthermore, high frequent noise was generated (recall Sect. 3.3.2). Therefore, the data had to be preprocessed using the B-spline methods introduced in Chapter 4.

Discussion. This system has been working reliably in an industrial environment for more than two years. The measuring system enables the offline inspection of samples on the production line and is used by machine operators. Currently, the geometry of 30 different types of catalytic converters can be inspected. Mechanical gauges are no longer used for that purpose. The results are directly used to control and correct the manufacturing equipment (e. g., welding machines), and thus to maintain manufacturing tolerances and quality.

Using this approach, a testing station for inspecting the geometry of wheel rims is presented in the next section.

7.2 Geometry Measurement of Wheel Rims

Quality assurance in the manufacturing is of vital importance. The high degree of automation requires automated control and quality assurance as well. In this scope, another 3D measuring device is presented that benefits from the scan data procedures previously proposed in this work. The measuring machine was developed to support the quality assurance in the manufacturing of wheel rims. Therefore, each wheel is measured, instead of taking single samples as for the converters in the section before. This section gives a brief overview on the system configuration, the used sensor technology, and the basic measuring procedure.



(a) (b) (c)

Fig. 7.5: Illustration of the wheel construction. The wheel front showing the spoke design, the wheel hub and the lug bolt holes is given in (a). The inspection areas at the bead seats are shown in (b) and at the backside of the wheel hub in (c).

For high driving comfort and a maximum safety, the exact geometry of a wheel is very important. If the metal parts are manufactured exactly, the rubber tire runs round. Any deviation from the ideal shape leads to vibrations which are noticeable for the driver. Especially the wheel hub, the pilot bore, and the lug bolt holes have a strong influence, since they represent the interface between the axes of the vehicle and the road (see Fig. 7.5). Important parameters are, for example, the radial and lateral run-out, the absolute and relative positions of several bores as well as the distances between inner/outer bead seat to each other and to the contact surface on the pilot bore.

Actually, the quality of a wheel is tested by the manufacturers with mechanical probes moving along the wheel surface. This process enables the determination of dints or deformations, which move the probe in radial direction. The application of mechanical probes is a widely-used and approved method, but still has several disadvantages: the probe wears out and the parameters cannot be acquired correctly. Additionally, the surface must be digitized pointwise to capture other or more complex geometry parameters, which becomes very time-consuming. Tactile machines are often placed in air-conditioned rooms, and thus are not suitable for online inspections in many cases. On the other hand, a temperature compensation is needed for optical (online) systems, too.

7.2.1 Functional Description

The measuring system mainly consists of three different subsystems: a 2D part identification unit for "chaotic" part supply, a 3D measuring unit for capturing and evaluating the wheel geometry and an additional elevator/side shift unit for NOK/Rework classifications. The single subsystems are connected by conveyers (see Fig. 7.6). In this work, we only focus on the 3D measuring unit.

(a) (b) (c)

Fig. 7.6: Schematic illustration of the wheel measuring machine (a), the real machine (b) and a closer view on the sensor technology (c).

In contrast to the converter measuring machine, this system uses the principle of a fixed wheel and moving sensors. Actually, this unit consists of four triangulation-based sensors. Three line-based sensors are used to capture the inner and outer bead seats and the center bore area with the contact patch and the lug bolt holes. The fourth sensor measures ranges to the bolt holes pointwise from the outside. Therefore, the sensors are mounted on two rotation axes (inner and outer sensors) with additional lateral axes for the optimal horizontal and vertical orientation. Within a continuous movement, the sensors are rotated around the wheel and 360 three-dimensional profiles are computed and combined to one point cloud which is finally evaluated and compared against the given nominal geometry from CAD.

The measuring system is designed to support all types of passenger car, SUV, and truck wheels without mechanical re-setting. Therefore, all necessary information (geometry parameters, parameters for custom-specific data evaluation) derived from given CAD data sets are stored in a database, which additionally enables an easy parameter modification and the insertion of new wheel types.

The measuring concept requires a triggered data acquisition for establishing angular relationships between successively captured scanlines. The basis for an equidistant (related to rotation angles) data capturing are the rotary encoders. They generate signals for a trigger unit, which again generates impulses for all connected devices. This principle assures the real-time and synchronous generation of trigger signals. The basic idea behind the hardware concept is the decoupled arrangement of data measuring and hardware controlling functions. This allows the minimization of the digital i/o signals between these systems.

Finally, the measured data is transfered to a standard PC via Firewire (IEEE 1394 standard) bus, which then performs the 3D data reconstruction, feature extraction and evaluation.

7.2.2 Measuring Procedure

The wheel is automatically transported to the measuring system on a conveyer, which is connected to the manufacturing process. Once the wheel enters the machine, it is pre-centered and arrested. The measuring arms and axes, on which the sensors are mounted, are automatically moved to the optimal measuring position. Within 3.6 seconds, the sensors rotate around the wheel and capture data of all relevant parts. After one complete rotation, 360 profiles (scanlines) were generated. Each profile initially contains 1024 2D coordinates, which are merged to more than 1.1 mio. 3D coordinates for the measurements. On the basis of the given CAD data of the wheels, a geometric matching compares the actual against the nominal shape. The entire process takes about 12 seconds and 8 seconds remain for the 3D evaluation. Thus, as many evaluations as possible must be applied to the scanlines as soon as they are acquired.

7.2.3 3D Calibration Procedure

The computation of 3D coordinates from the raw 2D profiles of the laser sensors requires a mathematical model of the machine axes and movements and a calibration procedure to calculate its parameters. Therefore, a setting master is used, whose global shape is similar to a wheel rim but with several revolving reference areas forming cylinders and planes (see Fig. 7.7(a)). Their dimensions are precisely determined using a tactile coordinate measuring machine. The nominal parameters of the setting master are stored in a database. For the adjustment of the sensors it was specified that the plane spanned by the laser has to coincide with its rotation axis. Thereby, the most attention is turned to the orthogonality between the plane normal and the direction of the axis. With the center axis of the setting master running nearly parallel to the rotation axes the complexity is reduces to a 2D problem.

The automated calibration procedure starts a regular measuring cycle for the setting master. At each angular position φ and for each sensor, a multi-stage alignment process starts. At first, a variant of the Iterative-End-Point-Fit algorithm (IEPF) [DH73, MQ94] is used to cluster the measured points into sets of collinear points. Starting with a line from the first to the last point of the ordered dataset, all distances to points in between are computed, and the point with the maximum distance specifies the intersection of two new lines. This is done recursively until a specified minimum distance is reached and the clusters are returned. After fitting lines to the point clusters according to [ARR99], the complexity is reduced from 1024 points to up to 10 line segments (see Fig. 7.7(b)), which makes the following calculations more efficient.

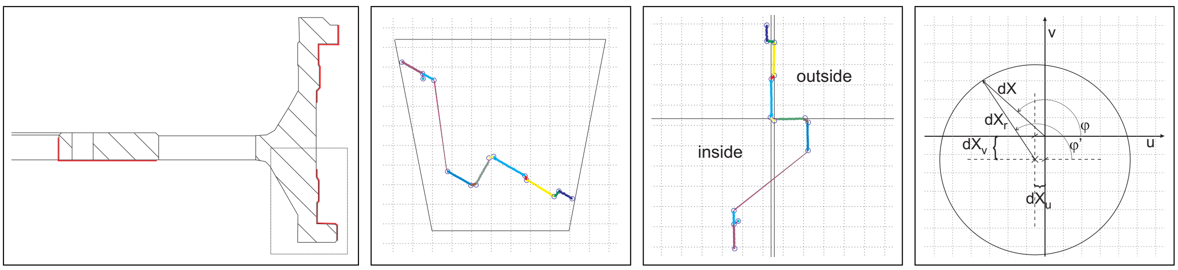
The setting master with its measured faces only defines two or three reference lines in the two-dimensional space for one sensor. All detected line segments are matched to the reference structure, a quality coefficient is computed and, at last, there are up to four possibilities to align the measured data to its reference. With the a priori knowledge that the sensors for the bead seats have to be outside of the rim and the sensor for the pilot bore has to be inside, the correct transformation $[\alpha, dX, dY]$ is chosen (see Fig. 7.7(c)).

The parameter α is the inclination of the sensor and $[dX, dY]$ its projection center referring to the coordinate system of the setting master. A subsequent fitting process [JH02, RL01] ensures the computation of transformation parameters for minimal deviation. The raw sensor data $[x_s, y_s]$ is transformed with the following equation.

$$\begin{aligned} x_t &= x_s \cdot \cos\alpha - y_s \cdot \sin\alpha + dX \\ y_t &= x_s \cdot \sin\alpha - y_s \cdot \cos\alpha + dY \end{aligned} \tag{7.1}$$

The last step determines the transformation parameters for the rotary movement separately for each sensor. Because the setting masters center axis is not identical to the rotation axis, the parameters change during the movement. These changes follow a circular path using the rotation angle φ and the corresponding α, dX and dY, respectively. For example, fitting a circle for dX results in three new parameters dX_u, dX_v, dX_r (see Fig. 7.7(d)). For small displacements between the axes the current rotation angle φ can also be used for the back transformation instead of using φ':

$$dX = \sqrt{dX_r^2 + 2dX_r(dX_u \cos\varphi + dX_v \sin\varphi) + dX_u^2 + dX_v^2}\,. \tag{7.2}$$



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

Fig. 7.7: The reference shape of the setting master (a), its 2D representation in the laser sensor (b), the transformed shape after fitting to the reference (c) and the coordinate system for the calibration with its transformation parameters (d).

The back transformation of all 2D profiles results in a 3D point cloud in the world coordinate system. For the 3D data processing, an object coordinate system is defined by the pilot bore with its theoretical rotation axis and the contact patch.

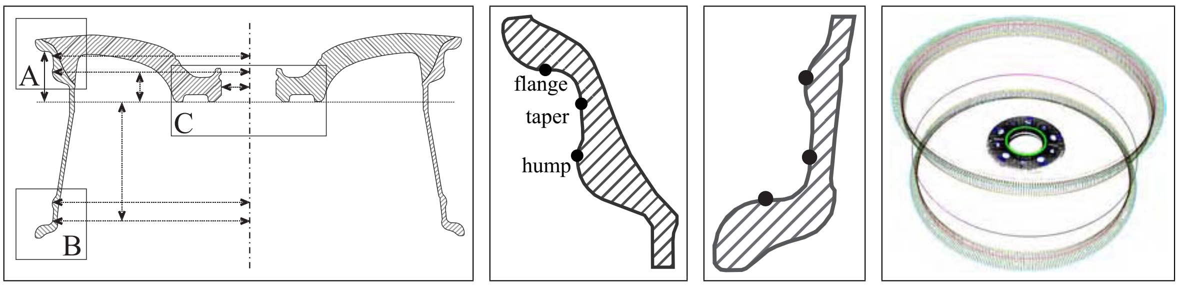
7.2.4 3D Data Processing

Usually, the acquired 3D data contains some high-frequent noise within a certain tolerance. This is due to the discrete sampling of the optical sensor on the one hand, and the different reflection properties of the surfaces on the other hand (recall Sect. 3.3.2). Therefore, a scanline-based B-spline smoothing is performed. Since the point cloud consists of single scanlines, we use this data structure to process the profiles independently from each other. Interrelationships between multiple scans are considered afterwards in the feature extraction and geometry evaluation steps.

To analyze a set of points on scanlines, B-splines are used as proposed in Chapter 4. They are used to obtain an analytical description from which features can be easily extracted. In addition, they are used to close small gaps between neighboring, disconnected sublines. Interpolating these gaps keeps the precision of measurements, if the distance between the corresponding sublines is less than a 0.5 mm.

3D Feature Extraction

The lateral and radial run-out determine the quality of the manufactured wheel as well as the position of the bolt holes. Therefore, the set of single radii over all measured features at pre-defined positions are considered. There are three important areas, which must be analyzed: the inner and the outer bead seat and the area at and around the pilot bore (see Fig. 7.8). The bead seat is the edge of the rim that creates a seal between the tire bead and the wheel. At the bead seat, there are three features: the rim flange, the rim taper, and the hump. The rim flange is the edge of the wheel and the shoulder is the outer edge of the tire tread where it meets the sidewall. The small hump is applied for safety reasons and ensures the optimal seat of the tire to the inner wheel. The area at the pilot bore consists of a contact surface, a cylindrical area at the hub, and the lug bolt holes for mounting the wheel at the vehicle axes.



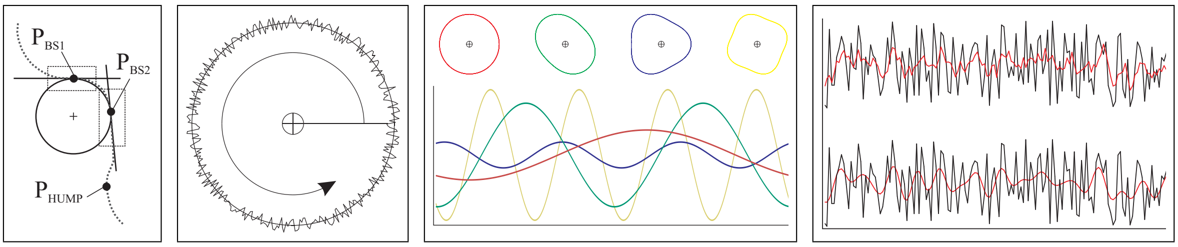
(a) (b) (c) (d)

Fig. 7.8: Schematic wheel construction (a) with marked expectation ranges: outer and inner bead seat [A], [B] and the area around the pilot bore [A]. Enlarged regions at the bead seats (b), (c) and the reconstructed 3D point cloud (d).

For the feature extraction from the point clouds, we use a priori CAD geometry information, since the wheel type is always known. Thus, expectation ranges are defined, which limit the possibilities, where a single feature can be found. In case of the bead seat features, the lateral run-out (planarity) is determined. The radial run-out represents the roundness. The planarity or conicity of the contact surface at the pilot bore is a direct quality measure, which also applies to the spatial position and diameters of the pilot bore and the lug bolt holes.

Measures at the Bead Seat

The shape of the bead seat and its measures with respect to the pilot bore are important parameters that significantly determine the runnability. If all radii to the rotation axis are the same, the wheel was manufactured optimally. This also applies to the distances of the bead seats to the outer contact plane at the pilot bore. The corresponding measures are found at the slope heading up to the rim flange and at the shoulder (see Fig. 7.9(a)). Therefore, a circle with a fixed radius of 8 mm must be placed at the bead seat, touching, but not intersecting the rim flange and the taper. This is achieved by fitting lines to the slopes in these regions. A parallel translation of these lines (the mandatory norm requires 8 mm) yields an intersection point, which is the center of the sought circle. The distances of the circle to the outer contact patch of the pilot bore and to the rotation axis are the measures for the run out. This procedure is applied to all scanlines of the inner and outer bead seat. As a result, a curve showing the development over the perimeter is obtained (see Fig. 7.9(b)). The contour of the resulting curves



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

Fig. 7.9: Measures at the sampled bead seat (a),(b) and result of the 1.–4. harmonic analysis (c). Figure (d) compares the moving average method (top) and the Fourier smoothing (bottom) on the radial distances. The noise in the signal is additionally amplified for a better visualization.

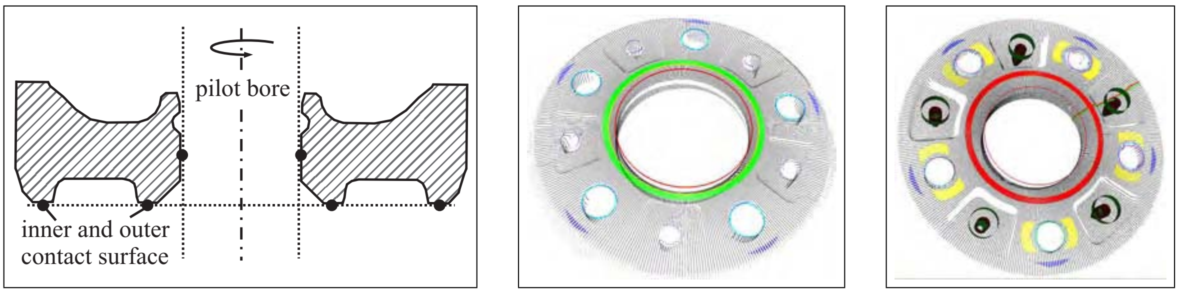
is evaluated by computing its Fourier series (Eq. 7.3). After decomposing the complex signal into its frequencies, the contained shapes are rated (see Fig. 7.9(c)). For example, the maximum of the first harmonic yields a direct measure of the deviation from a circle, the second considers ellipses. In the manufacturing, the first four harmonics must be analyzed.

$$F(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos nx + \sum_{n=1}^{\infty} b_n \sin nx \tag{7.3}$$

Usually, the resulting curve is smoothed in a preliminary step based on a very fast moving average approach. But in most cases this method is not able to smooth high frequencies sufficiently. A more sophisticated but slightly slower method is the Fourier interpolation. High frequencies, which are contained in the last harmonics, are removed by accumulating only the lower ones. The curve consists of 360 values, and with respect to the sampling theorem, only 180 harmonics can be used to interpolate this curve without oscillations. A smooth curve is obtained by accumulating only the first 90 harmonics (see Fig. 7.9(d)).

Measures at the Pilot Bore

The spatial position and the orientation of the pilot bore define the object coordinate system on the one hand, and the shape and the dimensions at the contact patch are geometric measures on the other hand. Each wheel has a cylindrical center bore, which must be detected. Due to the arrangement of the hardware, the rotation axes of the sensors and the wheel are nearly the same. This observation helps to detect an area around these axes. The resulting point cloud represents a cylinder, which is limited in its length by the inner contact patch (see Fig. 7.10(a)). Because of the varying wheel geometries, optimal data for robust cylinder fittings cannot be provided in every case. Therefore, the direction of the inner contact plane is employed. It is computed by fitting a plane. All the data from the assumed cylinder are now projected to this plane and constitute a circle. The center and radius of this circle and the direction of the plane finally define the position and orientation of the pilot bore. The bolt holes are found at locations parallel to the inner plane. Their shape is analyzed by approximating spheres and cones into their bore. The center of the shape is the reference for further measurements, such as other distances and radii.



(a) (b) (c)

Fig. 7.10: Scheme of the area at the pilot bore (a) and resulting point clouds with automatically detected colored inner/outer contact patches and the detected lug bolt holes (b),(c).

Fitting standard geometries to incomplete data is a recurrent problem [FWSW03]. Even if the curve is quite simple, such as a circle, it is still hard to reconstruct it from noisy data sampled along short arcs. Therefore, Chernov et al. study the least-squares, geometric and algebraic fit of circular arcs to incomplete scattered data [CL05]. For the wheel measurements the geometric fittings of Ahn were employed [ARR99].

Due to the fact that the machine is used in industrial environments, the influence of temperature changes must also be considered. Thus, all measures V (distances, radii) that have been directly computed, are corrected by a simple temperature compensation (Eq. 7.4), which includes a temperature coefficient T_c (e. g., aluminum $23.8 \cdot 10^{-6}$ per degree) and the difference ΔT between the actual temperature and the nominal 20 ℃:

$$V_c = V \cdot (1.0 + T_c \cdot \Delta T)\,. \tag{7.4}$$

Since this procedure does not consider temperature influences for the locomotor system and other mechanical parts, it is not yet complete.

Actually, the system is still a prototype to be introduced into manufacturing this year. The system has already passed a test series with 700 wheels for the most important measures, such as diameter of the pilot bore, lateral and planar run out and other measures at the bead seats. The results have been analyzed by the customers using the statistics software qs-Stat. They found the measures comparable to those of their tactile machines. Furthermore, the repeatability was excellent, only showing variances of 2–5μm. However, there are still some systematic deviations that must be explored and corrected. The chief cause may be found in an incomplete calibration concept, which does not cover all necessary mechanical uncertainties, such as elliptical movements of the locomotor systems instead of the expected rotations. A more detailed error analysis is in progress, but cannot be given due to secrecy agreements.

7.3 Summary

This section presented two industrial applications employing the scan data algorithms presented in this work. At the beginning, an automated 3D geometry inspection system for catalytic converters was introduced. This system is based on methods for automated geometric feature detection derived from scanline curvature information. Furthermore, local B-spline approximations are employed to revise single measurements in one step and to scan geometric features in another step (e. g., curvatures, sharp edges). Important geometric segments of a converter model (e. g., planes, circles, etc.) are scanned automatically and compared to the nominal values. The result of a measurement is a set of parameters describing the spatial location and the dimensions of parts of the model geometry. Afterwards, the measured geometry is compared to given values from drawings or CAD models, and deviations are identified. Deviations and errors are classified using given tolerance values provided by the manufacturer.

In the second part of this chapter a wheel measuring machine was presented, which is also based on the evaluation of 3D scanlines. A calibration procedure was proposed to find the transformations between the coordinate systems of the machine and the sensors. This approach is based on an alignment of the captured data to the nominal data of a calibrated setting master. In the next steps, the 3D data evaluation was discussed. Therefore, the acquired set of 3D data is corrected by approximating single scanlines with B-spline curves, which have a smoothing character and are able to fill holes. In the next steps, geometry extraction approaches were presented that use the given CAD data to adaptively define expectation ranges for the feature detection. The extracted surface parts were analyzed and compared to the given ones by approximating the same geometric primitives, such as lines, circles and planes.

Finally, a set of more than eighty measures and deviations are computed and stored in a database for process control and documentation. Actually, the system is a prototype, but this kind of 3D measurement is unique in the quality assurance for wheel production.

Chapter 8

Concluding Remarks

Modern quality assurance systems in the industry apply optical 3D scanners to capture and evaluate the geometry of automatically manufactured parts. However, 3D model acquisition is not limited to the industrial domain. Even 3D computer graphic applications employ these techniques to obtain and process model information from real-world objects, e. g., for visualization purposes.

This dissertation discussed the 3D scan data acquisition and its adaptive preparation and processing for the model based evaluation in practical applications. Thus, the focus was on automation, efficiency and robustness against environmental influences. Starting from the data acquisition, the entire process pipeline was considered. Therefore, the work is divided into three basic parts. The first part introduced the technical principles of 3D point cloud generation from optical (laser) scanners. In the second part, evaluation methods were presented employing the underlying scan data structures derived from the measuring principle. On this basis, procedures for the approximation of parametric curves and surfaces for data analyses were devised. The algorithms presented adaptively determine their parameters by exploiting the scan data structure and the known additional system information (e. g., sensor positions and movements). Finally, the third part discussed the application of the proposed techniques using the example of two industrial 3D measuring tasks.

The large amount of 3D measures, affected by noise and artifacts, usually complicates the automation in practical applications. Typically, polygonal meshing algorithms are employed for visualization and reconstruction of topological neighborhoods. Thus, the entire point cloud is considered, which usually contains too much information for local feature analyses. Therefore, the point cloud techniques in this work are based on the logical subdivision into its scan data structure. This approach enables an adaptive, directed and selective information reduction, which results in less complex evaluations and thus in a higher robustness and efficiency.

In summary, by taking the entire process pipeline into account (see Fig. 8.1), the degree of automation could be significantly increased. Therefore, modular tools were developed including simplification, smoothing or curvature evaluations based on adaptive scanline approximations by B-spline curves. Further evaluations were based on combining successively acquired scanlines to regular row/column grids for B-spline patch approximations and their evaluation. The remainder of this chapter summarizes the main contributions, discusses points of criticism, and future directions.

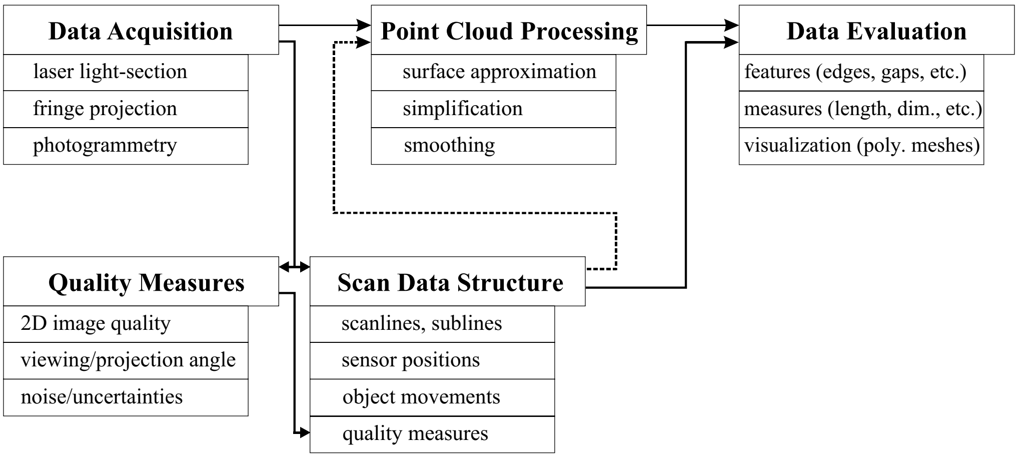


Fig. 8.1: The data acquisition pipeline and post-processing sequence. The scan data structure and system specific information enriches the measured data for an optimized point cloud processing and evaluation.

8.1 Summary of Contributions

When processing raw (unstructured) point clouds, a lot of information is thrown between data acquisition and data evaluation. Thus, the general strategy in this work is to analyze and exploit as much information as possible from the acquisition for a better evaluation. This procedure enables fewer errors from (information) approximations and a process- and sensor-adjusted data interpretation with higher processing speeds. Although this concept may be obvious, it has not been implemented with this complexity and completeness in practical applications before. On the one hand, the reason was found in many different and proprietary 3D sensors, and inconsistent data formats. On the other hand, point clouds without relation to acquisition hardware, and thus without additional system information, are typically processed. In this thesis, the entire procedure was considered. By employing 3D scanning technologies and data evaluations for real industrial applications, all necessary steps were brought together.

In practice, there is no automatic (optical) 3D measuring system which generates single point measurements independently from each other. Thus, there is always a structure that can be employed for a more efficient processing, requiring less approximation techniques. This results in an increasing robustness and degree of automation. At the example of line based scanners, points on a line are stored as connected neighbors. If the sensor is moved, successive lines are neighbors. The same assumption applies to area scanners, such as fringe projection and phase-shifting, where the multiple neighboring lines are captured within one step. Even photogrammetric measurements contain these structure, since the image matrix is a natural grid parameterization. Besides proprietary sensors, commercial devices were also used in the wheel measurement application. However, these devices also produce scanlines that can be processed with the proposed methods.

The data evaluation and analysis was discussed at three different consecutive abstraction levels. At first, the analysis of single measured contour lines (scanlines) was discussed, which enables a very selective and fast processing. The combination of successively captured scanlines to regular row/column grids represented the second level and analyzed interrelationships between scanlines. In the third level, the entire point cloud was finally processed.

It shall be mentioned that although this thesis has mainly focused on scan data evaluation from laser scanners, the contributions are not restricted to this domain. In fact, the techniques are suited to all optical measuring systems presented in the review of Chapter 2, which may only require some minor adjustments.

8.1.1 Data Acquisition

The analysis of the measuring principles indicated that there is already ancillary information about the generation of 3D point clouds. This includes the geometry of projected patterns, sensor positions, and defined movements during the acquisition.

At the example of two employed laser scanning systems, the data acquisition step and the 3D point cloud generation were introduced. Therefore, the system and measuring principle specific characteristics were extracted. Furthermore, a general scan data structure was derived, basically consisting of scanlines and sublines on the one hand, and additional system information about laser and camera positions on the other. In this scope, a scanline represents an ordered set of sublines, which themselves consist of an ordered sequence of points. An unambiguous point ordering is obtained from the unique (laser) projection angle. The scanlines order itself is defined by the order of the sensor movements. Furthermore, the influence from different error sources was discussed, which enables to perform error specific optimizations in the following processing steps.

The introduced scan data structure can also be derived from the other optical measuring systems, that generate point clouds line by line (e. g., fringe projection) or pointwise (e. g., stereo-photogrammetry).

8.1.2 Data Evaluation

In the data evaluation stage, point clouds were approximated by a sorted set of B-spline curves to obtain an analytical description. Based on this representation, edge information was derived from these curves with respect to curvature values and the viewing/projection quality of each single point. Furthermore, the curve approximation reduces high-frequent noise and allows to interpolate small gaps. Since the proposed methods are based on the scanline approach discussed in Chapter 4, they may easily be adapted to other scanning systems.

In addition to processing scanlines separately, an efficient procedure for the automated preview generation from several scan operations was presented. Therefore, scanline points were sorted according to their projection angle in the same way as proposed for the curve processing. The movements of the sensors were additionally exploited to establish an order between single scanlines. Based on the projection angles, an approach to construct regular row/column grids was presented. Furthermore, a more global data analysis, in contrast to the curve processing was employed in order to check for surface features, geometric discontinuities, gaps and holes. Taking the grids as a basis, a method for the rational B-spline surface approximation was presented. These parametric surfaces were employed to evaluate the surface topology and compute different surface curvatures, e. g., to check for surface fairness and existence of surface features such as edges.

The next abstraction level discussed the processing of multiple grids and cases of unordered scanlines sequences, e. g., as derived from the flexible scanner. Therefore, approved techniques to manage 3D point sets, such as kd-trees, were used. These methods were adjusted by adding the a priori known system information and quality measures. Furthermore, it was found out that the contained noise is normally distributed, which enables an average based smoothing and simplification or fitting of geometric primitives as outlined in Section 6.2.3.

Due to different application requirements (e. g., measuring task, object complexity, computation speed, etc.), one or a combination of the three approaches may be used to analyze, correct and evaluate the data. Due to the different abstractions it is hard to compare the different results, because more or less (additional) information is available. Thus, while the scanline evaluation is very fast, it is very local compared to the point cloud approach. The grid approach exists between them.

Finally, two industrial applications were presented employing the scan data algorithms presented in this work. At the beginning, an automated 3D geometry inspection system for catalytic converters was introduced. In the second part, a wheel measuring machine was presented. Both measuring systems employ the laser light section principle. Therefore, the proposed methods could be used for automated geometric feature detection, for example based on the scanline curvature information. The resulting measures are finally compared to given values from drawings or CAD models and deviations are identified.

8.2 Criticism

Although most of the proposed methods have been successfully applied to industrial measuring tasks, there are still some limitations. For example, the scanline-based curvature evaluation and edge detection require the scanline to "intersect" the object. This requirement may not be fulfilled for some system/object configurations. But in practice, there were mainly free form surfaces and this problem did not occur.

In case of grid construction from neighboring scanlines, the scanlines are not allowed to overlap, which may result in degenerated shapes. This requirement is, for example, not fulfilled by the presented flexible measuring device. However, even in this case, a separate scanline processing can be applied for local analyses anyway. Global relations must be derived from a general 3D point sets procedure as proposed in Chapter 6.

8.3 The Horizon

Throughout this thesis, extensions and improvements for processing optical 3D scan data were discussed. In addition to the proposals for solving the problems discussed above, there are also some major directions to continue the work presented in this thesis.

8.3.1 A Unifying Scan Data Description

In this work, the possibilities of a formal description for optical 3D scan data have been explored at the example of laser scanners. The implicitly given data structures and system information (see Fig. 8.1) enable a significantly more efficient data processing and robust data interpretation. However, as already mentioned, the derived structuring is not limited to the employed laser scanners. Thus, a further goal is the development of a formalism and the definition of a unifying data format for scan data from optical 3D scanners in general. This allows for the measuring system independent application of the methods proposed. Furthermore, this approach enables the exchange of data sets between several software packages from different developers comparable to simple ASCII coordinate files. Furthermore, the variety of existing proprietary algorithms for different scan data can be combined based on a standardized description.

8.3.2 Future Worker Support Systems

In addition to the quality assurance tasks presented, interdisciplinary approaches combining optical 3D metrology and 3D computer graphics have a variety of further application domains. The ideal case is the implementation of a worker support system. The following scenario with respect to the converter application illustrates this idea: Within the production step, the worker takes a sample item and puts it into the optical measuring device. After data acquisition, the converter type is automatically detected. In the following steps, the sought geometric measures are computed and the systems detect the deviations. The derived 3D model is prepared to illustrate the problematic location and automatically aligned for an optimal view on it. Based on the given nominal data, a modification proposal is automatically made and visualized or animated to the worker.

This scenario includes loads of subtasks that must be processed. The scan data preparation and automated feature detection may be performed by applying the methods presented in this thesis. Methods for database-related type queries may be performed by using either the features already obtained or the shape-based skeletons as described in [Ise04].

Bibliography

[AB99] N. Amenta and M. Bern. Surface reconstruction by voronoi filtering. *Discr. Comput. Geom.*, 22:481–504, 1999.

[ABCO+01] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point set surfaces. In *IEEE Visualization 2001*, pages 21–28. IEEE Computer Society, October 2001.

[ABCO+03] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, January 2003.

[AC98] F. Allamandri and P. Cignoni. Adaptively adjusting Marching Cubes output to fit a trilinear reconstruction filter. In *Eurographics Workshop on Scientific Visualization '98*, pages 25–34, 1998.

[ACK01] N. Amenta, S. Choi, and R. K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Comput. Geom., Theory and Applications*, 19:201–210, 2001.

[Ahn04] S. J. Ahn. *Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space*. Lecture Notes in Computer Science. Springer, Berlin, 2004.

[ARR99] S. J. Ahn, W. Rauh, and M. Recknagel. Geometric fitting of line, plane, circle, sphere, and ellipse. In *Proc. of 6. ABW-Workshop Optische 3D-Formerfassung (Esslingen, Germany, 25–26 January, 1999)*, pages 1–8. Technical Academy Esslingen, 1999.

[AS00] M. Attene and M. Spagnuolo. Automatic surface reconstruction from point sets in space. *Computer Graphics Forum*, 19(3):457–465, 2000.

[Aur91] F. Aurenhammer. Voronoi Diagrams—A Survey of a Fundamental Geometric Data Structure. *ACM Computing Surveys*, 23(3):345–405, September 1991.

[BBX97] C. L. Bajaj, F. Bernardini, and G. Xu. Reconstructing surfaces and functions on surfaces from unorganized three-dimensional data. *Algorithmica*, 19(1):243–261, 1997.

[BFTT05a] D. Berndt, A. Fix, E. Trostmann, and C. Teutsch. 3-D image processing as the key for a flexible manufacturing cell for the automated welding of large steel structures. In A. Gruen and H. Kahmen, editors, *Proc. of Optical 3-D Measurement Techniques VII (Vienna, Austria, October 3–5, 2005)*, volume 1, pages 317–326, Vienna, 2005. TU Vienna, ETH Zurich.

[BFTT05b] D. Berndt, A. Fix, T. M. E. Trostmann, and C. Teutsch. Optical 3-d metrology for aiding the autonomous welding of steel constructions. In *EOS Conf. on Industrial Imaging and Machine Vision (Munich, Germany, 13–15 June, 2005)*, pages 12–16. European Optical Society, 2005.

[BHP06] R. Bade, J. Haase, and B. Preim. Comparison of fundamental mesh smoothing algorithms for medical surface models. In *Simulation und Visualisierung*, pages 289–304. SCS-Verlag, 2006.

[BKP07] R. Bade, O. Konrad, and B. Preim. Reducing artifacts in surface meshes extracted from binary volumes. *Journal of WSCG*, 15(1–3):67–74, 2007.

[BMM+02] F. Bernardini, I. Martin, J. Mittleman, H. Rushmeier, and G. Taubin. Building a Digital Model of Michelangelo's Florentine Pietà. *IEEE Computer Graphics & Applications*, 22(1):59–67, January 2002.

[BMR+99] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, Oct-Dec 1999.

[BOW+04] N. Blanc, T. Oggier, G. G. Weingarten, A. Codourey, and P. Seitz. Miniaturized smart cameras for 3D-imaging in real-time. In *Proc. IEEE Sensors '04*, volume 1, pages 471–474, 2004.

[BR04] B. J. Brown and S. Rusinkiewicz. Non-rigid range-scan alignment using thin-plate splines. In *Proc. 3D Data Processing, Visualization, and Transmission, 2nd Int.l Symp. on (3DPVT'04)*, pages 759–765, Washington, DC, USA, 2004. IEEE Computer Society.

[Bre93] B. Breuckmann. *Bildverarbeitung und optische Messtechnik in der industriellen Praxis.* Franzis Verlag, München, 1993.

[BSG+03] I. N. Bronstein, K. A. Semendjajew, G. Grosche, V. Ziegler, and D. Ziegler. *Taschenbuch der Mathematik*, volume 1. B.G. Teubner Verlagsgesellschaft, Stuttgart, 2nd edition, November 2003.

[CC06] Z. Chen and H.-L. Chou. New efficient octree construction from multiple object silhouettes with construction quality control. In *18th Int. Conf. on Pattern Recognition (ICPR 2006)*, volume 1, pages 127–130, 2006.

[CL95] B. Curless and M. Levoy. Better optical triangulation through spacetime analysis. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 987, Washington, DC, USA, 1995. IEEE Computer Society.

[CL05] N. Chernov and C. Lesort. Least squares fitting of circles. *Journal of Mathematical Imaging and Vision*, 23(3):239–252, November 2005.

[dB72] C. de Boor. On calculation with B-splines. *Jour. Approx. Theory*, 6:50–62, 1972.

[dBvKOS00] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, 2nd edition, 2000.

[Del24] B. N. Delaunay. Sur la sphere vide. In *Proc. Int. Congress of Mathematicians, (Toronto, Canada, August 11–16)*, pages 695–700. University of Toronto Press (1928), 1924.

[DG04] T. K. Dey and S. Goswami. Provable surface reconstruction from noisy samples. In *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*, pages 330–339, New York, NY, USA, 2004. ACM Press.

[DH73] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.

[DLS05] T. K. Dey, G. Li, and J. Sun. Normal estimation for point clouds : A comparison study for a voronoi based method. In *Eurographics Sympos. on Point-Based Graphics (2005)*, pages 39–46, 2005.

[Dol97] J. Dold. *Ein hybrides photogrammetrisches Industriemeßsystem höchster Genauigkeit und seine Überprüfung*. PhD thesis, Universität der Bundeswehr, München, 1997.

[Doo78] D. Doo. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In *Proceedings of the International Conference on Interactive Techniques in Computer Aided Design (Bologna, Italy, September, 1978)*, Conference Proceedings, pages 157–65. IEEE Computer Society, ACM, 1978.

[EEAHM05] S. El-Etriby, A. Al-Hamadi, and B. Michaelis. Improvement 3-d reconstruction accuracy considering distortion in stereovision using a set of linear spatial filters. In *Computational Intelligence for Modelling, Control and Automation '05*, volume 2, pages 654–659, 2005.

[EH95] M. Eck and J. Hadenfeld. Knot removal for B-spline curves. *Computer Aided Geometric Design*, 12(3):259–282, 1995.

[EH96] M. Eck and H. Hoppe. Automatic Reconstruction of B-spline Surfaces of Arbitrary Topological Type. *SIGGRAPH 96*, 30:325–334, August 1996.

[FH05] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 157–186. Springer, Berlin, Heidelberg, 2005.

[FS07] K. Fritzsche and G. Schubert. Das LIDAR-Projekt Leipzig. Website. `http://www.imn.htwk-leipzig.de/~p1fri/lidar/lidar.html`, June 2007.

[FvDFH00] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. The System Programming Series. Addison-Wesley, Boston, MA, 2nd edition, 2000.

[FWSW03] R. Frühwirth, W. Waltenberger, A. Strandlie, and J. Wroldsen. A review of fast circle and helix fitting. *Nuclear Instruments and Methods in Physics Research*, A502(2):705–707, April 2003.

[Gib98] S. F. F. Gibson. Constrained elastic surface nets: Generating smooth surfaces from binary segmented data. In *MICCAI'98*, pages 888–898. Springer, 1998.

[GL92] R. Godding and T. Luhmann. Calibration and accuracy of a multi-sensor on-line-photogrammetric system. *Int. Archives of Phot. & Remote Sensing*, 29(B5):24–34, 1992.

[Had98] J. Hadenfeld. *Iteratives Glätten von B-Spline Kurven und B-Spline Flächen*. PhD thesis, TU Darmstadt, 1998.

[HBL97] W. Heidrich, R. Bartels, and G. Labahn. Fitting Uncertain Data with NURBS. In *Proc. Curves and Surfaces in Geometric Design*, pages 1–8, Nashville, 1997. Vanderbilt Univ. Press.

[HDD+93] H. Hoppe, T. DeRose, T. Duchamp, J. McDondald, and W. Stuetzle. Mesh Optimization. In *Proc. SIGGRAPH 93*, pages 19–26, New York, 1993. ACM Press.

[HDD+94] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise Smooth Surface Reconstruction. *SIGGRAPH 94*, 28:295–302, 1994.

[HG00] K. Hormann and G. Greiner. Quadrilateral remeshing. In *5th Vision Modeling and Visualization '00*, pages 153–162, 2000.

[Hoa61] C. A. R. Hoare. Partition: Algorithm 63, quicksort: Algorithm 64, and find: Algorithm 65. *Comm. ACM*, 4:321–322, 1961.

[Hoa62] C. A. R. Hoare. Quicksort. *Computer J.*, 5:10–15, 1962.

[Hop94] H. Hoppe. *Surface Reconstruction from Unorganized Points*. PhD thesis, University of Washington, 1994.

[Hor03] K. Hormann. From scattered samples to smooth surfaces. In D. Cohen-Or, N. Dyn, G. Elber, and A. Shamir, editors, *Proc. Fourth Israel-Korea Bi-National Conf. on Geometric Modeling and Computer Graphics*, pages 1–5, Tel Aviv, Israel, February 2003.

[Ise04] T. Isenberg. *Capturing the Essence of Shape of Polygonal Meshes*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, Germany, 2004.

[JH02] T. Jost and H. Hugli. Fast ICP algorithms for shape registration. In *Pattern Recognition – 24th DAGM Symposium*, pages 91–99, Zurich, Switzerland, September 2002.

[KBSS01] L. P. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel. Feature sensitive surface extraction from volume data. In *Proc. SIGGRAPH '01*, pages 57–66, New York, NY, USA, 2001. ACM Press.

[KKŽ05] J. Kohout, I. Kolingerová, and J. Žára. Parallel delaunay triangulation in E^2 and E^3 for computers with shared memory. *Parallel Comput.*, 31(5):491–522, 2005.

[Kru56] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Problem. Am. Math. Soc.*, 7:48–50, 1956.

[KvD92] J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8):557–565, October 1992.

[KVPL04] O. Konrad-Verse, B. Preim, and A. Littmann. Virtual resection with a deformable cutting plane. In T. Schulze, S. Schlechtweg, and V. Hinz, editors, *Proc. of 15. Conference on Simulation and Visualization (Magdeburg, Germany, 4–5 March, 2004)*, pages 203–214, Erlangen, San Diego, 2004. University of Magdeburg, SCS European Publishing House.

[LC87] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH 87*, 21(3):163–169, July 1987.

[LC00] R. Lohner and J. Cebral. Generation of non-isotropic unstructured grids via directional enrichment. *International Journal for Numerical Methods in Engineering*, 49(1):219–232, September 2000.

[LCK06] H. Lee, H. Cho, and M. Kim. A new 3d sensor system for mobile robots based on moire and stereo vision technique. In *Int. Conf. on IEEE/RSJ Intell. Robots and Systems '06*, pages 1384–1389, 2006.

[Lee00] I.-K. Lee. Curve reconstruction from unorganized points. *Comput. Aided Geom. Design*, 17(2):161–177, 2000.

[LP05] C. Lange and K. Polthier. Anisotropic smoothing of point sets. *Comput. Aided Geom. Des.*, 22(7):680–692, 2005.

[LPC+00] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project: 3D Scanning of Large Statues. In K. Akeley, editor, *Proc. SIGGRAPH 2000*, pages 131–144. ACM SIGGRAPH, Addison Wesley, 2000.

[LRKH07] T. Luhmann, S. Robson, S. Kyle, and I. Harley. *Close-Range Photogrammetry*. Wiley, 2007.

[Maa97] H.-G. Maas. Mehrbildtechniken in der digitalen Photogrammetrie, Habilitationsschrift, ETH Zurich, 1997.

[MAB06] D. K. Mackinnon, V. Aitken, and F. Blais. A comparison of precision and accuracy in triangulation laser range scanners. In *Canadian Conf. on Electrical and Comp. Engineering*, pages 832–837, 2006.

[MAM04] J. Marchadier, D. Arques, and S. Michelin. Thinning grayscale well-composed images. *Pattern Recognition Letters*, 25(5):581–590, April 2004.

[MH03] X. Q. Meng and Z. Y. Hu. A new easy camera calibration technique based on circular points. *Pattern Recognition*, 36(5):1155–1164, 2003.

[Min61] M. Minsky. Microscopy apparatus. U.S. Patent 301467, Dec. 19 1961.

[MK05] M. Marinov and L. Kobbelt. Optimization methods for scattered data approximation with subdivision surfaces. *Graph. Models*, 67(5):452–473, 2005.

[MNG04] N. J. Mitra, A. Nguyen, and L. Guibas. Estimating surface normals in noisy point cloud data. *Special issue of Int. J. Computational Geometry and its Applications*, 14(4–5):261–276, 2004.

[MQ94] T. Michael and T. Quint. Sphere of influence graphs in general metric spaces. *Mathematical and Computer Modelling*, 29:45–53, 1994.

[Nie04a] G. M. Nielson. Dual Marching Cubes. In *IEEE Visualization'04*, pages 489–496, 2004.

[Nie04b] G. M. Nielson. Radial hermite operators for scattered point cloud data with normal vectors and applications to implicitizing polygon mesh surfaces for generalized csg operations and smoothing. In *VIS '04: Proc. of the Conf. on Visualization '04*, pages 203–210, Washington, DC, USA, 2004. IEEE Computer Society.

[NISA06] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Laplacian mesh optimization. In *ACM GRAPHITE 2006*, pages 381–389, 2006.

[NW99] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 1999.

[OBA+03] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 463–470, New York, NY, USA, 2003. ACM Press.

[Owe98] S. J. Owen. A survey of unstructured mesh generation technology. In *Proc. 7th Int. Meshing Roundtable (Sandia National Lab)*, pages 239–267, 1998.

[Par97] J. R. Parker. *Algorithms for image processing and computer vision.* John Wiley & Sons, Inc., New York, NY, USA, 1997.

[PGK02] M. Pauly, M. Gross, and L. P. Kobbelt. Efficient simplification of point-sampled surfaces. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 163–170, Washington, DC, USA, 2002. IEEE Computer Society.

[PKKG03] M. Pauly, R. Keiser, L. Kobbelt, and M. Gross. Shape modeling with point-sampled geometry. In *Proc. SIGGRAPH 2003*, pages 641–650. ACM Press, 2003.

[Pri57] R. C. Prim. Shortest connection networks and some generalizations. *Bell System. Tech. J.*, 36:1389–1401, 1957.

[PS97] Y. P. Presnyakov and V. P. Shchepinov. Use of the speckle effect to analyze vibrations of a rough surface. *Technical Physics*, 42(8):919–922, August 1997.

[PT95] L. Piegl and W. Tiller. *The NURBS Book.* Monographs in Visual Communication. Springer, Berlin, 2nd edition, 1995.

[PTVF02] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing.* Cambridge University Press, Cambridge, UK, 2nd edition, 2002.

[PYL99] I. K. Park, I. D. Yun, and S. U. Lee. Constructing NURBS Surface Model from Scattered and Unorganized Range Data. In *Proc. 3-D Imaging and Modeling*, Annual Conference Proceedings, pages 312–340. National Research Council of Canada, 1999.

[Ray74] L. Rayleigh. On the manufacture and theory of diffraction gratings. *Philosophical Magazine*, 47:81–93, 193–204, 1874.

[Rin66] T. Rindfleisch. Photometric method for lunar topography. *Photogrammetric Engineering*, 32(2):262–277, March 1966.

[RL01] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pages 145–152, 2001.

[Rog01] D. F. Rogers. *An Introduction to NURBS.* Academic Press, San Diego, 2001.

[RTvVV02] B. Rieger, F. Timmermans, L. van Vliet, and P. Verbeek. Curvature estimation of surfaces in 3d grey-value images. In C. S. R. K. D. Laurendeau, editor, *Proc. 16th Int. Conf. on Pattern Recognition (ICPR'02)*, volume 1. IEEE Computer Society, August 2002.

[Rus04] S. Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 486–493, 6-9 Sept. 2004.

[Sch00] S. Schröder. Kombination von hierarchischem Blockmatching und dynamischer Programmierung zur automatischen Zuordnung korrespondierender Bildbereiche bei der Stereofotogrammetrie. Master's thesis, Otto-von-Guericke-Universität Magdeburg, 2000.

[Sch05] C. Schulz. Approximation von kruemmungsinformationen zur umsetzung von techniken zur illustrativen medizinischen visualisierung. Master's thesis, Otto-von-Guericke-University Magdeburg, Dept. of Computer Science, 2005.

[SGS05] D. Stoppa, L. Gonzo, and A. Simoni. Scannerless 3d imaging sensors. In *IEEE IST '05*, pages 58–61, 2005.

[SKP+06] S. Surti, S. Karp, L. M. Popescu, E. Daube-Witherspoon, and M. Werner. Investigation of time-of-flight benefit for fully 3-dpet. In *IEEE Trans. on Medical Imaging*, volume 25, pages 529–538, May 2006.

[SLS+06] A. Sharf, T. Lewiner, A. Shamir, L. Kobbelt, and D. Cohen-Or. Competing fronts for coarse-to-fine surface reconstruction. *Computer Graphics Forum*, 25(3):389–398, 2006.

[SLS07] P. Stelldinger, L. J. Latecki, and M. Siqueira. Topological equivalence between a 3d object and the reconstruction of its digital image. In *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, volume 29, January 2007.

[SM83] F. Stentiford and R. Mortimer. Some new heuristics for thinning binary handprinted characters for OCR. *IEEE Trans. Systems, Man and Cybernetics*, 13:81–84, 1983.

[SN06] P. Sarder and A. Nehorai. Deconvolution methods for 3-d fluorescence microscopy images. In *IEEE Signal Processing Magazine*, volume 23, pages 32–45, 2006.

[SPSP02] D. Selle, B. Preim, A. Schenk, and H.-O. Peitgen. Analysis of vasculature for liver surgical planning. *IEEE Trans. Med. Imaging*, 21(11):1344–1357, 2002.

[SS71] Y. Shirai and M. Suwa. Recognition of polyhedrons with a range finder. In D. C. Cooper, editor, *Proc. of the 2nd Int. Joint Conf. on Artificial Intelligence, (London, UK, September, 1971)*, pages 80–87, 1971.

[SS96] L. L. Schumaker and S. S. Stanley. Shape-preserving knot removal. *Computer Aided Geometric Design*, 13(9):851–872, 1996.

[SSFS06] J. Schreiner, C. Scheidegger, S. Fleishman, and C. Silva. Direct (re)meshing for efficient surface processing. *Comp. Graph. Forum*, 25(3):527–536, 2006.

[Str93] T. Strutz. *Ein genaues aktives Triangulationsverfahren zur Oberflächenvermessung*. PhD thesis, Faculty of Electrical Engineering and Information Technology (FEIT), Otto-von-Guericke Universität Magdeburg, 1993.

[SWPG05] F. Sadlo, T. Weyrich, R. Peikert, and M. Gross. A practical structured light acquisition system for point-based geometry and texture. In *Proc. Point-Based Graphics, Eurographics '05*, pages 89–145, 2005.

[TBSS06] C. Teutsch, D. Berndt, A. Sobotta, and S. Sperling. A flexible photogrammetric stereo vision system for capturing the 3d shape of extruded profiles. In P. S. Huang, editor, *Proc. Two- and Three-Dimensional Methods for Inspection and Metrology IV (Optics East 2006, October 1-4, 2006, Boston, MA, USA)*, volume 6382, page 63820M. SPIE, 2006.

[TBST06] C. Teutsch, D. Berndt, N. Schmidt, and E. Trostmann. Automated geometry measurement of wheel rims based on optical 3d metrology. In P. S. Huang, editor, *Proc. Two- and Three-Dimensional Methods for Inspection and Metrology IV (Optics East 2006, October 1-4, 2006, Boston, MA, USA)*, volume 6382, page 63820I. SPIE/IS&T, SPIE, 2006.

[TBTM05] C. Teutsch, D. Berndt, E. Trostmann, and R. Mecke. A hand-guided flexible laser-scanner for generating photorealistically textured 3D data. In *Proceedings of 3D-NordOst 2005*, 7. Anwendungsbezogener Workshop zur Erfassung, Verarbeitung, Modellierung und Auswertung von 3D-Daten, pages 37–44. Virtual Working Group 3D, 2005.

[TBTP07] C. Teutsch, D. Berndt, E. Trostmann, and B. Preim. Adaptive realtime grid generation from 3d line scans for fast visualization and data evaluation. In *Proc. 11th Int. Conf. Information Visualization*, pages 177–184, Washington, DC, USA, 2007. IEEE Computer Society.

[TBTW04] C. Teutsch, D. Berndt, E. Trostmann, and M. Weber. 3D geometry verification in the automotive industry. In *Proceedings of 3D-NordOst 2004*, 7. Anwendungsbezogener Workshop zur Erfassung, Verarbeitung, Modellierung und Auswertung von 3D-Daten, pages 51–58. Virtual Working Group 3D, 2004.

[TBTW05] C. Teutsch, D. Berndt, E. Trostmann, and M. Weber. Efficient reconstruction of nurbs surfaces for shape analysis and surface inspection. In A. Gruen and H. Kahmen, editors, *Proc. of Optical 3-D Measurement Techniques VII (Vienna, Austria, October 3–5, 2005)*, volume 2, pages 144–153, Vienna, 2005. TU Vienna, ETH Zurich.

[Teu03] C. Teutsch. Optimierte Rekonstruktion von 3D-Geometrien aus Laserscan-Daten unter Ausnutzung von Information über den Scan-Prozess. Master's thesis, Otto-von-Guericke-University Magdeburg, Germany, Department of Simulation and Graphics, september 2003.

[Thu97] W. P. Thurston. *Three-dimensional Geometry and Topology.* Princeton Univ. Press New Jersey, 1997.

[TIT+05] C. Teutsch, T. Isenberg, E. Trostmann, M. Weber, T. Strothotte, and D. Berndt. Evaluation and correction of laser-scanned point clouds. In J.-A. Beraldin, S. F. El-Hakim, A. Gruen, and J. S. Walton, editors, *Proc. of Videometrics VIII (Electronic Imaging 2005, San Jose, California, USA, 16–20 January, 2005)*, volume 5665, pages 172–183, Bellingham, Washington, January 2005. SPIE/IS&T.

[TITW04] C. Teutsch, T. Isenberg, E. Trostmann, and M. Weber. Evaluation and Opimization of Laser Scan Data. In T. Schulze, S. Schlechtweg, and V. Hinz, editors, *Proc. of 15. Conference on Simulation and Visualization (Magdeburg, Germany, 4–5 March, 2004)*, pages 311–322, Erlangen, San Diego, 2004. University of Magdeburg, SCS European Publishing House.

[TM06] C. Teutsch and M. Maasland. Kombinierte optische Vermessung und Oberflächenprüfung von 3-D-Objekten. *Leitfaden zur Inspektion von Oberflächen mit Bildverarbeitung*, 9(1):28–32, May 2006.

[TMHF00] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[TO04] T. Tu and D. R. O'Hallaron. Extracting hexahedral mesh structures from balanced linear octrees. In *Proc. 13th Int. Meshing Roundtable (Williamsburg, VA, Sandia Nat. Labs., September 19-22)*, SAND 2004-3765C, pages 191–200, 2004.

[Ver75] K. J. Versprille. *Computer-aided Design Applications of the Rational B-spline Approximation Form.* PhD thesis, Syracus University, Syracuse, NY, 1975.

[WH06] Y. Wu and Z. Hu. Camera calibration and direct reconstruction from plane with brackets. *Journal of Mathematical Imaging and Vision*, 24(3):279–293, May 2006.

[WMW86] G. Wyvill, C. McPheeters, and B. Wyvill. Data structure for soft objects. *The Visual Computer*, 2(4):227–234, February 1986.

[WS05] D. Wobschall and M. Z. B. Srinivasaraghavan. An ultrasonic/optical pulse sensor for precise distance measurements. In *Sensors for Industry '05*, pages 31–34, 2005.

[XPB06] G. Xu, Q. Pan, and C. L. Bajaj. Discrete surface modelling using partial differential equations. *Comput. Aided Geom. Des.*, 23(2):125–145, 2006.

[Yao75] A. C. Yao. An O(E loglog V) algorithm for finding minimum spanning trees. *Inf. Process. Lett.*, 4:21–23, 1975.

[Zha99] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proc. Int. Conf. on Computer Vision (ICCV'99)*, pages 666–673, 1999.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[ZS84] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27(3):236–239, 1984.

Appendix A

List of Implementations

The following table contains a list of tools that were implemented in connection with this dissertation. In all implementations the programming language C++ was used.

tool	author	description
QTviewer	Michael Schiller, Nico Schmidt, Christian Teutsch	scandata visualization for optical sensors
GLviewer	Christian Teutsch	openGL scandata evaluation and visualization tools
optoInspect Converter	Erik Trostmann, Michael Schiller, Christian Teutsch	evaluation procedures for the converter measuring machine
optoInspect Rim	Nico Schmidt, Christian Teutsch, Erik Trostmann	evaluation procedures and tool for visualizing the results of the rim measuring machine
optoInspect Stereo	Christian Teutsch	photogrammetric stereo application for measuring and visualizing the front of moving plastic profiles
math library	Christian Teutsch, Erik Trostmann	approximation of NURBS curves, NURBS surfaces and geometric primitives
sensor library	Michael Schiller, Christian Teutsch	methods for integrating the scandata algorithms into optical sensors functions
pointCloud library	Christian Teutsch	kd-tree-based point cloud evaluation

Tab. A.1: List of implementations created in connection with this dissertation.

Appendix B

Supplementary Details

In addition to Eqs. 5.6 to 5.9, the forward and backward differences for regular grids of arbitrary order k are given by the following relations:

$$\Delta_n^k = \Delta^k a_n = \sum_{i=0}^{k} (-1)^i \binom{k}{i} a_{n+k-1} \tag{B.1}$$

$$\nabla_p^k = \nabla^k f_p = \sum_{m=0}^{k} (-1)^m \binom{k}{m} f_{p-m} \; . \tag{B.2}$$

Furthermore, computing the derivatives of NURBS surface is much more complicated than in the non-rational case. This is due the requirement, that influence of the weights must be included into the computation of the rational basis functions. In addition to Eqs. 5.18 to 5.22, the corresponding terms for for the (weighted) partial derivatives are given here.

The nominators N and denominators D of Eqs. 5.18 to 5.22 are given by the following expressions (also see [Rog01, PT95]):

$$\bar{N}_u = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} B_{i,j} N'_{i,k}(u) M_{j,l}(v) \tag{B.3}$$

$$\bar{N}_v = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} B_{i,j} N_{i,k}(u) M'_{j,l}(v) \tag{B.4}$$

$$\bar{N}_{uv} = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} B_{i,j} N'_{i,k}(u) M'_{j,l}(v) \tag{B.5}$$

$$\bar{N}_{uu} = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} B_{i,j} N''_{i,k}(u) M_{j,l}(v) \tag{B.6}$$

$$\bar{N}_{vv} = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} B_{i,j} N_{i,k}(u) M''_{j,l}(v) \tag{B.7}$$

$$\bar{D}_u = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j} N'_{i,k}(u) M_{j,l}(v) \tag{B.8}$$

$$\bar{D}_v = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j} N_{i,k}(u) M'_{j,l}(v) \tag{B.9}$$

$$\bar{D}_{uv} = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j} N'_{i,k}(u) M'_{j,l}(v) \tag{B.10}$$

$$\bar{D}_{uu} = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j} N''_{i,k}(u) M_{j,l}(v) \tag{B.11}$$

$$\bar{D}_{vv} = \sum_{i=1}^{n+1}\sum_{j=1}^{m+1} h_{i,j} N_{i,k}(u) M''_{j,l}(v)\,. \tag{B.12}$$

List of Figures

List of Tables