

Die approbierte Originalversion dieser Diplom-/Masterarbeit ist an der Hauptbibliothek der Technischen Universität Wien aufgestellt (<http://www.ub.tuwien.ac.at>).

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology (<http://www.ub.tuwien.ac.at/englweb/>).

DIPLOMA THESIS

3D Surface-Scanning and -Reconstruction Methods and Experiments

Institute of Solid State Physics
Vienna University of Technology
Wiedner Hauptstrasse 8-10
A-1040 Wien

Supervisor:

Ao. Univ. Prof. Dipl.Ing. Dr.techn. Norbert Gurker

by

Christoph Bloch
Franz-Josefs Kai 43/16
1010 Wien

Vienna, March 2004

DIPLOMARBEIT

3D Oberflächen-Scanning und -Rekonstruktion Methoden und Experimente

Institut für Festkörperphysik
Technische Universität Wien
Wiedner Hauptstrasse 8-10
A-1040 Wien

unter Anleitung von:

Ao. Univ. Prof. Dipl.Ing. Dr.techn. Norbert Gurker

von

Christoph Bloch
Franz-Josefs Kai 43/16
1010 Wien

Wien, März 2004

Abstract

3D surface-scanning -the capture of an object's shape, or more formal the acquisition of a (dense) set of 3D coordinates on the object's surface- is in the focus of this diploma thesis. 3D Surface Scanning is becoming more and more important, and it has a wide variety of applications ranging from prototyping and in-process inspection, via medicine and biometrics to the entertainment industry. In this thesis an overview of the hardware to obtain surface samples (3D scanners) and the algorithms used for the reconstruction process are given. These reconstruction algorithms compute meshes from the measured data (usually a set of points, rarely color information and estimated surface normals), which are intended to give a good representation of the scanned object's shape.

In Part I we will first describe non-contact range measurement methods. Based on these distance measuring methods we will demonstrate a wide variety of usual 3D scanning principles. Furthermore we will give a more detailed description of specially selected scanning systems.

Part II is concerned with surface reconstruction algorithms. We will discuss selected algorithms in sufficient detail that a basic understanding of their procedures is given, and that an implementation is straightforward.

In Part III we will finally present the 3D laser scanner experiment in our laboratory that has been developed during this thesis to serve as a test environment and our self-developed scanning software we have used for data-handling. Furthermore, the stages of the surface-reconstruction process are illustrated, and an impression of our datasets is given.

Kurzfassung

3D Oberflächenscanning -das Erfassen der Form eines Gegenstandes, oder mehr formell das Erfassen eines (dichten) Datensatzes von Oberflächenpunkten- ist Thema dieser Diplomarbeit. 3D Scannen gewinnt immer mehr an Bedeutung und findet ein breites Anwendungsspektrum, das vom Prototypenbau und der Qualitätskontrolle über die Medizin und die Biometrie bis zu Anwendungen in der Unterhaltungsindustrie reicht.

In dieser Arbeit wird sowohl auf die Hardware, die für den Scan-Prozess erforderlich ist, als auch auf die Rekonstruktions-Algorithmen eingegangen. Die Aufgabe der Rekonstruktions-Algorithmen besteht darin, aus den durch den Scanvorgang gewonnen Daten (meist nur die Koordinaten der Oberflächenpunkte, gegebenenfalls Farbinformation, und Näherungen für die Oberflächennormalen) das abgetastete Objekt möglichst gut zu rekonstruieren.

Im ersten Abschnitt werden zunächst Methoden der berührungslosen Entfernungsmessung beschrieben. Auf diesen aufbauend wird dann ein Überblick über verschiedene 3D Scanprinzipien gegeben. Anschließend werden ausgewählte Systeme ausführlicher abgehandelt.

Der zweite Abschnitt befasst sich mit Rekonstruktionsalgorithmen, dabei werden die einzelnen Algorithmen so detailliert behandelt, dass ein grundlegendes Verständnis über deren Arbeitsweise gegeben wird und eine direkte Implementierung möglich ist.

Im letzten Abschnitt beschreiben wir den 3D Laser-Scanner in unserem Labor, der im Rahmen dieser Diplomarbeit entwickelt wurde, um eine Testumgebung zu schaffen und unsere selbst entwickelte Software, die für den Scan- und Rekonstruktionsprozess eingesetzt wird. Weiters werden die verschiedenen Stufen des Rekonstruktionsprozesses dargestellt und ein rekonstruiertes Objekt exemplarisch präsentiert.

Contents

I	Methods of 3D-Surface-Scanning:	1
1	Introduction	2
2	Range Measurement	4
2.1	Triangulation [13]	4
2.1.1	Passive Triangulation	5
2.1.2	Active Triangulation	6
2.2	Interferometry	8
2.2.1	Multiple Wavelength Interferometry (MWI) [6]	9
2.2.2	White Light Interferometry / Coherence Radar [13]	11
2.2.3	Conoscopic Holography [57, 58]	12
2.3	TOF (Time-of-Flight) Ranging [13]	14
2.3.1	Pulsed Modulation [13]	15
2.3.2	Continuous Wave (CW) Modulation [13]	17
2.3.3	Pseudo-Noise (PN) Modulation [26]	17
3	Overview of Non-Contact 3D Scanning Systems	20
3.1	3D Scanning Systems based on Triangulation [15]	20
3.1.1	Photogrammetry (Passive Triangulation)	21
3.1.2	Depth from Focus and Defocus [27]	21
3.1.3	Structured Light (Active Triangulation) [54, 34]	21
3.2	3D Scanning Systems based on Interferometry	26
3.2.1	Multiple Wavelength Interferometry (MWI) [7]	26
3.2.2	Conoscopic Holography [58, 59]	27
3.2.3	Coherence Radar [9, 12]	28
3.3	3D TOF-Scanning-Systems	29
3.3.1	Acoustic Position Tracker	29
3.3.2	“Scanning beam” 3D Scanners (optical TOF) [22]	30
4	Selected 3D-Scanning Systems	33
4.1	Terrestrial or Close-Range Photogrammetry [31, 32]	33
4.1.1	Correspondence Analysis [69]	36
4.1.2	Bundle Block Adjustment [31]	38
4.2	Structured-Light Range Scanning for Moving Scenes [34]	39
4.2.1	Objectives of the Scan-System-Design	40
4.2.2	Concept of the Scanning System	40
4.2.3	Alignment of Multiple Scans	40

4.2.4	Illumination Pattern	42
4.2.5	Layout and Results	43
4.2.6	Improved Illumination Pattern [37]	43
4.3	Non-Scanning 3D TOF-Camera [13]	45
4.3.1	Demodulation and Sampling	46
4.3.2	Demodulation Pixels	49
4.3.3	System	50
 II Methods of 3D-Surface Reconstruction		52
 5 General Statement of the Problem		53
 6 Hoppe's Algorithm		54
6.1	Basic Principle [39]	54
6.2	The Signed Distance Function [39]	54
6.2.1	Tangent Plane Estimation	55
6.2.2	Consistent Tangent Plane Orientation	56
6.2.3	Implementation	57
6.3	Contour Tracing	57
6.4	Algorithm Complexity and Conclusion [41]	57
 7 Three Dimensional Alpha-Shapes		59
7.1	Intuitive Description [2]	60
7.2	Terms and Definitions	60
7.3	Observations on Alpha-Shapes	62
7.4	Implementation [2]	63
7.4.1	Implicit Representation	64
7.5	Conclusion	65
 8 Power-Crust Algorithm		66
8.1	Sketch of the Algorithm	66
8.2	Geometric Definitions [42]	67
8.3	Further Definitions [42]	68
8.4	Basic Algorithm [42]	71
8.5	Extensions [42]	73
8.6	Conclusion	74
 9 Cocone Algorithm		75
9.1	Definitions [48]	75
9.2	Theoretical Background	76
9.3	Basic Algorithm	78
9.3.1	Approximation of the Restricted Voronoi Diagram [48]	78
9.3.2	Manifold extraction [48, 49]	79
9.4	Conclusion and Extensions	82

III Experiments	83
10 Experimental Setup	84
10.1 Hardware	84
10.1.1 Touch-Probe Scanner	85
10.1.2 Laser Triangulation Sensor	85
10.1.3 Mounting the Laser Triangulation Sensor	86
10.1.4 Addressing the timing problem	87
10.1.5 Scanning Procedure	87
10.1.6 Accuracy and Performance	88
10.2 Software	89
10.2.1 Sampling	89
10.2.2 Decimation	89
10.2.3 Reconstruction	90
10.2.4 Smoothing	90
10.3 Stages of the Surface Reconstruction Process	91
10.3.1 Example of Scanned Data	93
10.4 Conclusion and Outlook	94
IV Appendix	95
A Geometric Definitions and Proofs	96
A.1 Glossary	96
A.1.1 Convex Hull [52]	96
A.1.2 General Position	96
A.1.3 Simplex	96
A.1.4 Voronoi Diagram [1]	96
A.1.5 Delaunay Triangulation [1]	97
A.1.6 Medial Axis [43]	98
A.2 Proofs	98
A.2.1 Definitions	98
A.2.2 Lemmas	99
B SCANDESK Manual	102
B.1 Installation and Components	102
B.2 User's Guide	102
B.2.1 Data-Objects and File-Handling	104
B.2.2 The View Toolbar	105
B.2.3 The Scanner Toolbar	105
B.2.4 The Data Manipulation Toolbar	106

Part I

Methods of 3D-Surface-Scanning:

Chapter 1

Introduction

3D surface scanning, the capture of an objects shape, is designated as the acquisition of a (dense) set of 3D coordinates on the objects surface and has widespread applications. Just to name a few of the disciplines that consider surface scanning as important: In medicine 3D scanning systems are used by dentists, in plastic surgery, or to align 3D volumetric data (obtained by NMR,CT) of the patient to his current position (determined by a surface scan) on the operation table. Also in-process inspection and reverse engineering make use of 3D scanners. In combination with fast prototyping procedures 3D copiers can be realized. Archaeology benefits from 3D scanning as well, since the conventional measurement of the artefacts is often time consuming- because of their complex shapes. The obtained 3D models can be used for later restoration or in virtual museums. Of course, 3D models have become important in the entertainment industry, where they are used for computer games and for special effects in motion pictures. Here 3D scanners offer a way to digitize models that are made of clay, for example, a method which is often faster than to the use 3D modelling software.

Each of these various applications needs special requirements. For many applications measuring precision is most important. Other applications necessitate large measurement ranges, portability, ease of use and reasonable short scanning times.

Therefore many different scanning systems are available. In a first overview, provided by Figure 1.1, the systems are classified by their range measurement methods.

In chapters 3 and 4 some examples of 3D scanning systems are described in more detail. In the following chapter we will start with a description of the various range measurement methods.

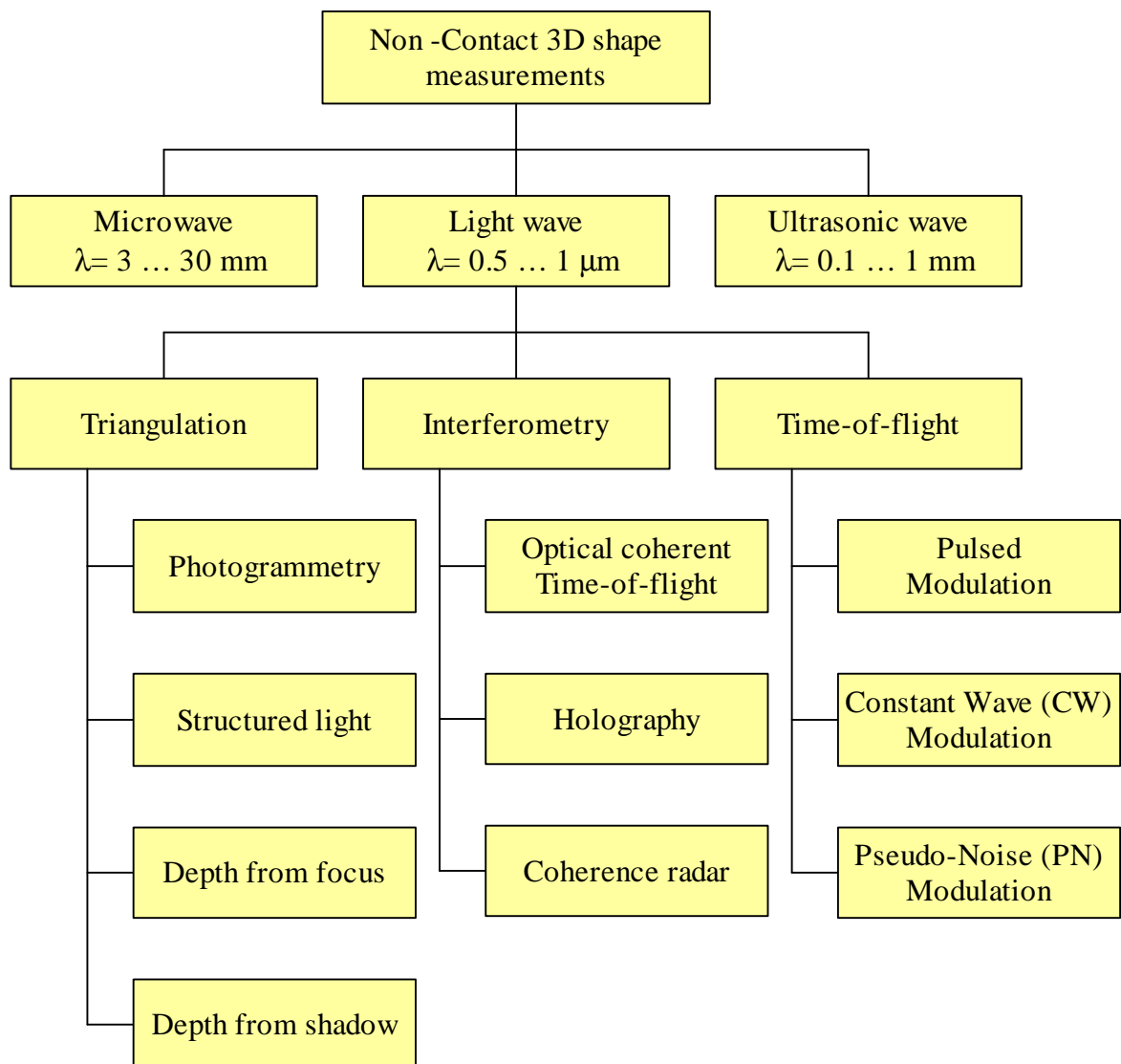


Figure 1.1: Non-Contact 3D shape measurements according to [19].

Chapter 2

Range Measurement

In this chapter we will describe distance-measurement techniques based on triangulation, interferometry and time-of-flight. A graph of the performance of these various methods is given in Figure 2.1.

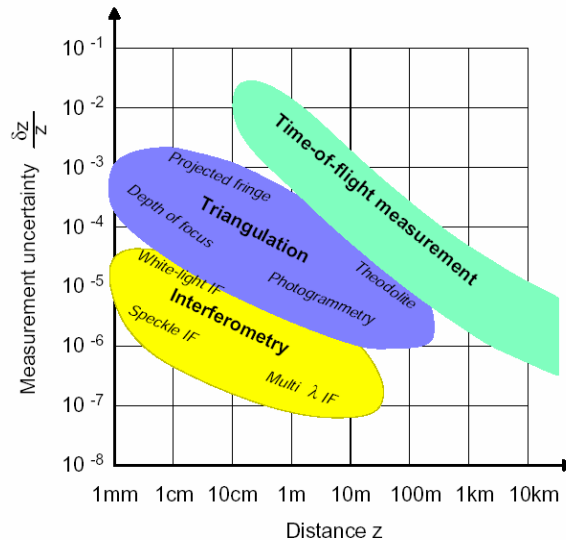


Figure 2.1: Performance map of optical 3D measurement systems [18].

2.1 Triangulation [13]

This ranging technique has been known and used by nature for millions of years. It is, in the form of stereo-vision, together with the depth-of-focus system (which can also be considered to belong to triangulation systems), the basis for human depth perception. Triangulation is a geometrical approach, where the target is one point of a triangle of which the two remaining points are known in the measurement system. The distance of the target can then be determined by measuring the triangle's angles or the triangulation base. In literature passive and active triangulation techniques are distinguished.

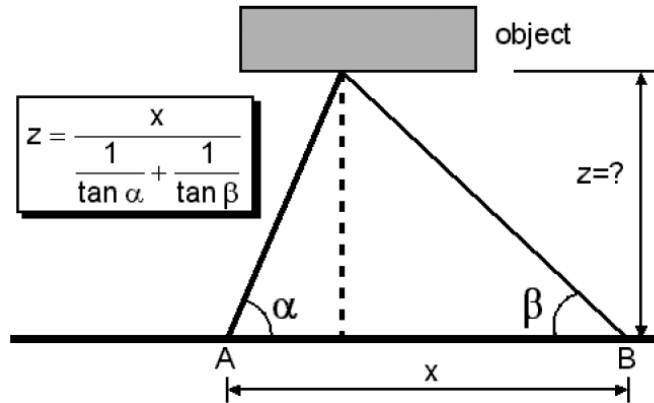


Figure 2.2: Passive Triangulation [13].

2.1.1 Passive Triangulation

Passive techniques rely on observing the same point from two different sites A and B of a known distance x and measuring the viewing angles α and β with respect to the base AB , as illustrated in Figure 2.2. The observed point's distance z can then be calculated using the following equation:

$$z = \frac{x}{\frac{1}{\tan \alpha} + \frac{1}{\tan \beta}} \quad (2.1)$$

Since each point to be measured must be identified from both viewing positions unambiguously, passive triangulation techniques require a scene with high contrast. Stereovision is one famous 3D realization of passive triangulation. This technique uses at least two cameras to observe the scene from different angles. Using 2D-correlation, typical object features are found and compared in both images. From the position of each feature's centroid in both separate images, the angles α and β can be deduced and the distance can be calculated with Equation 2.1, assuming that the distance of one camera from the other, as well as their orientations are known. The efforts of such computation must not be underestimated. Shadowing effects are also typical problems, which all triangulation systems have to cope with. Stereovision works pretty well for certain defined scenes, preferably chosen with rich contrast and relatively flat objects. For typical industrial scenes, however, it is often not suitable. Though the shadowing problem can be minimized by enlarging the number of cameras and realizing "multiple viewpoint triangulation systems", this improvement has to be paid for by an enormous increase in computation complexity. Nevertheless, shadowing and the need for significant contrast in the targeted scene remain a problem. Additionally, cost and the overall system size -one major drawback of triangulation systems anyhow- increase with the number of cameras.

Another famous member of passive triangulation systems is the theodolite.

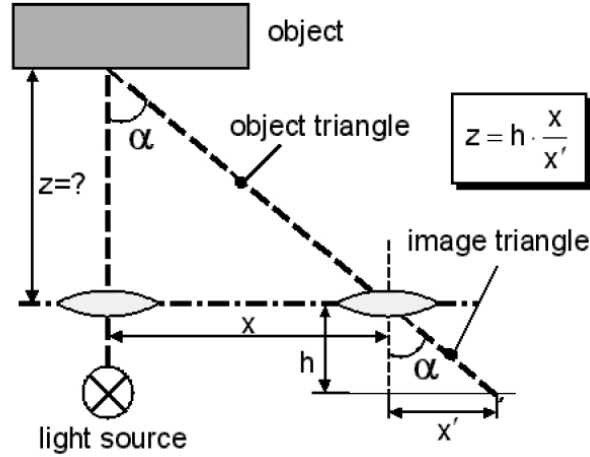


Figure 2.3: Active Triangulation [13].

2.1.2 Active Triangulation

Active triangulation, as illustrated in Figure 2.3, uses a light source to project a point (in the simplest case) to the scene, which is observed by a position sensitive detector. Rather than measuring angles directly, active triangulation is based on the similarity of triangles, the object triangle and the image triangle, which is fully defined by the optical axis of the imaging device, the focal length h of the system, and the position of the point projection x' on the detector. Knowing the displacement x of the light source from the imaging device, the distance z of the target can be determined:

$$z = h \cdot \frac{x}{x'} \quad (2.2)$$

the distance resolution δz estimates to:

$$\delta z = \frac{1}{h} \cdot \frac{z^2}{x} \cdot \delta x' \quad (2.3)$$

Hence for a good distance resolution δz small absolute distances z , a large triangulation base x and a good local detector resolution $\delta x'$ are required.

A large triangulation base means that the sensors cannot be built arbitrarily compact and are sensitive to shadowing effects. Shadowing effects can be limited by the use of multiple sensors. An interesting approach to this issue is the circular triangulation sensor developed by Wolf and Beck [20, 21]:

The focused laser beam emitted from the laser diode is projected onto the object surface and a spot is formed on the surface to be measured, as shown in Figure 2.4, 2.5. The scattering light from the spot is partly condensed by a rotation-symmetric condenser lens onto a ring shaped PSD (Position Sensitive Detector) array. The measuring distance in the direction of the axis of a projected beam is detected by calculating the average radius of a ring-shaped detector according to the triangulation principle. The main advantage of this laser probe is that the measuring distance can be detected using only parts of the scattering light. This helps to overcome the problems typically found with conventional triangulation laser probes, where shading-effects at steps or other shape discontinuities can cause problems by loss of

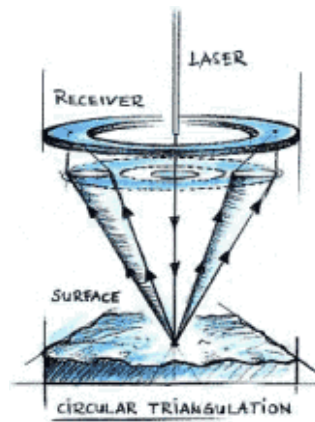


Figure 2.4: Circular Triangulation [20].

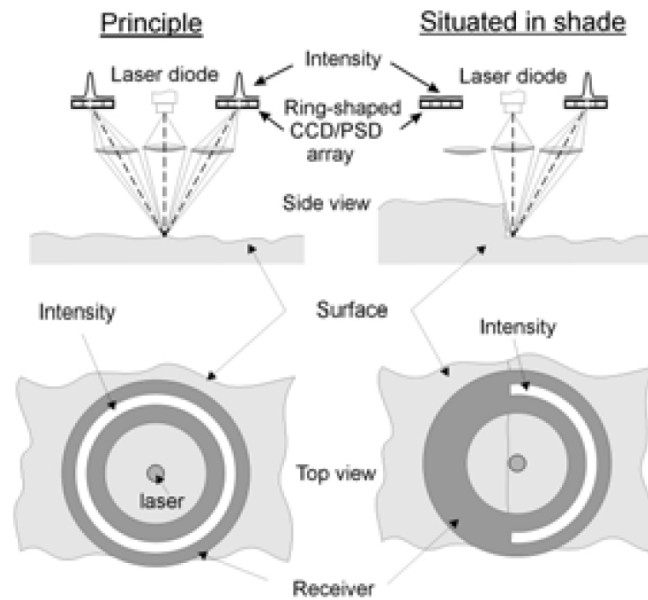


Figure 2.5: Working principle of the circular triangulation laser probe [21].

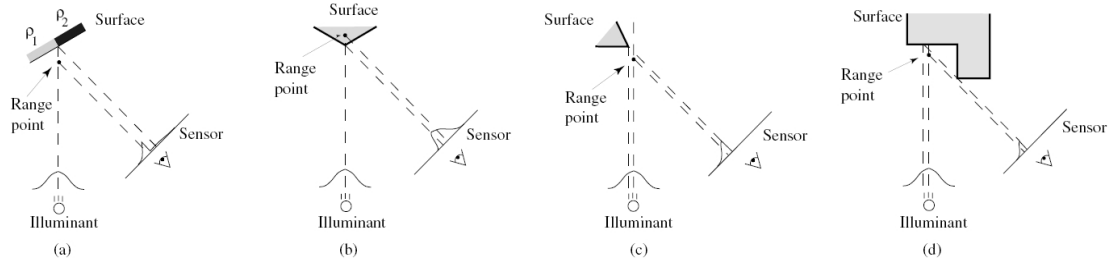


Figure 2.6: Range errors using traditional triangulation methods. (a) Reflectance discontinuity. (b) Corner. (c) Shape discontinuity with respect to the illumination. (d) Sensor occlusion [14].

signal. Also the measuring uncertainty can be significantly improved. The working distance of this laser probe type is 85mm and the measuring range is 20mm. A resolution of $2\mu\text{m}$ can be attained with this probe.

Generally active triangulation systems are suitable for a widespread scope of range measurements. Drawbacks are shadowing effects, low resolution (compared to interferometry) and improper detection of the illumination pattern due to textures, corners or partial occlusion introduces extra errors, as illustrated in Figure 2.6.

2.2 Interferometry

Interferometric techniques use collinear beams, so no shadowing effects appear and in addition they are not sensitive to textures.

Interferometry is described by the superposition of two (or more) monochromatic waves resulting in another monochromatic wave of the same frequency ν , but with different phase and different amplitude. For interference the waves need to have coincident frequency, polarization and a fixed phase-difference. This phase difference can be estimated by observing the evolving interference pattern.

In the best known interferometer setup, the Michelson interferometer, illustrated in Figure 2.7, a laser beam (monochromatic and coherent) is split into two beams by a beam splitter. One beam is directed to a mirror of constant displacement x_1 (reference path) whereas the other beam is targeted onto the object at a variable distance x_2 (measurement path). Both beams are reflected back to the beam splitter, which superimposes them onto an integrating detector.

Considering the fact that each beam has to travel the distance forth and back to the splitter we have a path difference of $(2x_2 - 2x_1)$, thus we obtain the interference equation

$$I = I_1 + I_2 + 2\sqrt{I_1 \cdot I_2} \cdot \cos \left\{ \frac{2\pi \cdot (2x_2 - 2x_1)}{\lambda} \right\} \quad (2.4)$$

where I_1 and I_2 are the optical intensities of the two beams U_1 and U_2 .

This method allows very precise distance-measurements, but it is restricted to a unambiguous measurement range of $\frac{\lambda}{2}$. Interferometry can also provide incremental values:

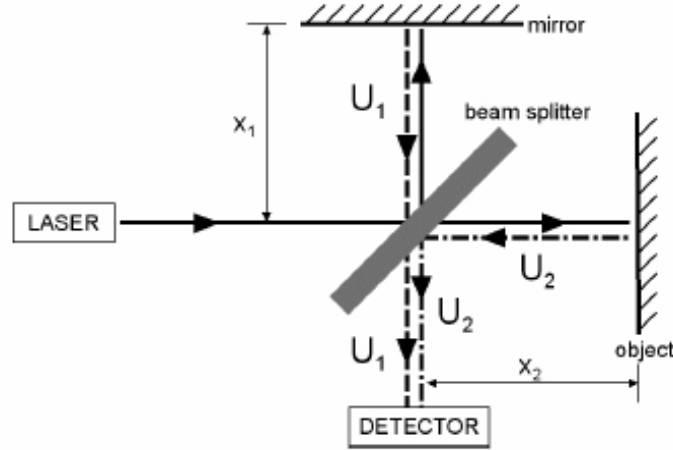


Figure 2.7: Michelson Interferometer [13].

If the object is moved in relation to the interferometer, the arising intensity-extrema at the detector -on a minimum follows a maximum after a further displacement of $\frac{\lambda}{4}$ - can be counted easily.

However, incremental measurement is a severe restriction in many cases.

It also has to be mentioned that interferometry runs into problems when applied to rough surfaces. If the surface(-height) varies in the order of λ in the area of the measurement spot, no meaningful interference patterns can be obtained due to speckle effects.

2.2.1 Multiple Wavelength Interferometry (MWI) [6]

Both limitations can be overcome by Multiple-Wavelength Interferometry, which is also a coherent method, but it offers great flexibility in sensitivity by an appropriate choice of the different wavelengths, leading to synthetic wavelengths.

For instance, let us consider two-wavelength interferometry using the optical wavelengths λ_1 and λ_2 . For an interferometric path difference z , the phases ϕ_1 and ϕ_2 corresponding to the wavelengths λ_1 and λ_2 are given by

$$\phi_1 = 2\pi \frac{2z}{\lambda_1}, \phi_2 = 2\pi \frac{2z}{\lambda_2} \quad (2.5)$$

The phase difference between ϕ_1 and ϕ_2 is given by

$$\Delta\phi = \phi_1 - \phi_2 = 2\pi \left(\frac{2z}{\lambda_1} - \frac{2z}{\lambda_2} \right) = 2\pi \frac{2z}{\Lambda} \quad (2.6)$$

Thus the phase difference can be considered to be sensitive to a new synthetic wavelength Λ , which can be expressed as

$$\Lambda = \frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2} \quad (2.7)$$

Therefore, the use of two slightly different wavelengths permits the generation of a new synthetic wavelength much longer than the individual optical wavelengths. The range of non-ambiguity of the phase difference $\Delta\phi$, which is also known as the synthetic phase, is therefore

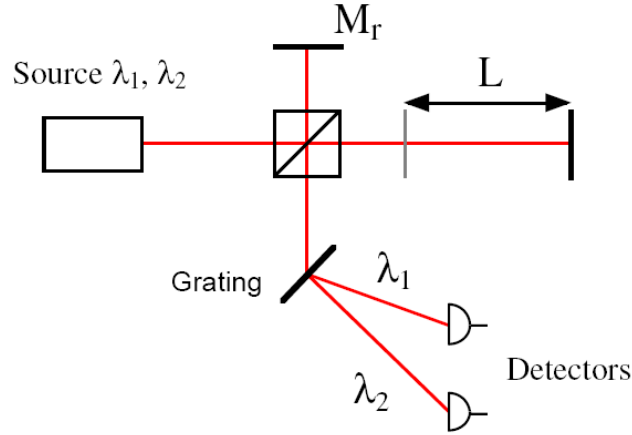


Figure 2.8: Multiple Wavelength Interferometry: separation of the wavelengths [6].

increased compared to the range of non-ambiguity of classical interferometry. Moreover, the sensitivity of the measurement is reduced and Multiple-wavelength-interferometry can thus be operated on rough surfaces.

Two-wavelength interferometry can be accomplished by injecting two wavelengths simultaneously into the interferometer, and by optically separating them at the output using a prism or a grating, as shown in Fig. 2.8.

Both interference signals are then detected individually. The synthetic phase can be determined by measuring the interferometric phases at both wavelengths, and by computing the difference. However, this method only works for relatively large wavelength differences and thus small synthetic wavelengths ($< 1\text{mm}$), since both wavelengths have to be separated by means of a prism or a grating. Moreover, this method requires interferometric stability at the optical wavelength, which is difficult to achieve in applications where reduced sensitivity is desired and sufficient. Alternative methods can be used to solve this problem by detecting the total interference signal without any optical separation of the two wavelengths.

Superheterodyne detection, introduced by Dändliker et al. [8], enables high resolution measurements at arbitrary synthetic wavelengths Λ without the need for interferometric stability at the optical wavelengths λ_1 and λ_2 or separation of these wavelengths optically. This is of great importance for range-finding and industrial distance measuring of large distances with sub-millimeter resolution.

In Superheterodyne Interferometry we apply a frequency shift f_i to each wavelength (the superposition of the original and the shifted signal is called heterodyne modulation)

$$\begin{aligned} U_1 &= \sqrt{I_1} \exp(i\phi_1) \exp(i2\pi\nu t) \\ U_2 &= \sqrt{I_2} \exp(i\phi_2) \exp[i2\pi(\nu + f)t] \end{aligned} \quad (2.8)$$

The interference signal, which is given by $I = |U_1 + U_2|^2$ then becomes

$$I(t) = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos(2\pi f t + \Delta\phi) \quad (2.9)$$

where $\Delta\phi = \phi_2 - \phi_1$.

Thus for every wavelength we have a signal of the form

$$I_{\lambda_1}(t) = B_0 + B_1 \cos(2\pi f_1 t + \Delta\phi_1) ; I_{\lambda_2}(t) = C_0 + C_1 \cos(2\pi f_2 t + \Delta\phi_2) \quad (2.10)$$

Assuming incoherent superposition the total intensity equals to

$$I(t) = I_{\lambda_1}(t) + I_{\lambda_2}(t) = A_0 + A_1 \cos(2\pi f_1 t + \Delta\phi_1) + A_2 \cos(2\pi f_2 t + \Delta\phi_2) \quad (2.11)$$

Further assuming $A_1 = A_2 = A_{12}$ the time dependent part can be written as

$$I(t)' = A_{12} \cos[2\pi(f_1 - f_2)t + (\Delta\phi_1 - \Delta\phi_2)] \cdot \sin[2\pi(f_1 + f_2)t + (\Delta\phi_1 + \Delta\phi_2)] \quad (2.12)$$

Because $f_1 - f_2$ is chosen to be small compared with f_1 and f_2 , the detector output has the form of a carrier-suppressed amplitude-modulated signal with carrier $(f_1 + f_2)/2$ and modulation frequency $(f_1 - f_2)/2$. After amplitude demodulation, one therefore gets

$$I_{dem}(t) = A_{12} \cos[2\pi(f_1 - f_2)t + (\Delta\phi_1 - \Delta\phi_2)] \quad (2.13)$$

This signal at $f = f_1 - f_2$ makes it possible to directly measure the phase difference $\Delta\phi_1 - \Delta\phi_2 = 4\pi z/\Lambda$ which is now only sensitive to the synthetic wavelength Λ .

The phase-measurement is performed electronically by a comparison with an optically generated reference signal having $\Delta\phi_1 - \Delta\phi_2 = 0$. A typical setup is shown in Figure 2.9.

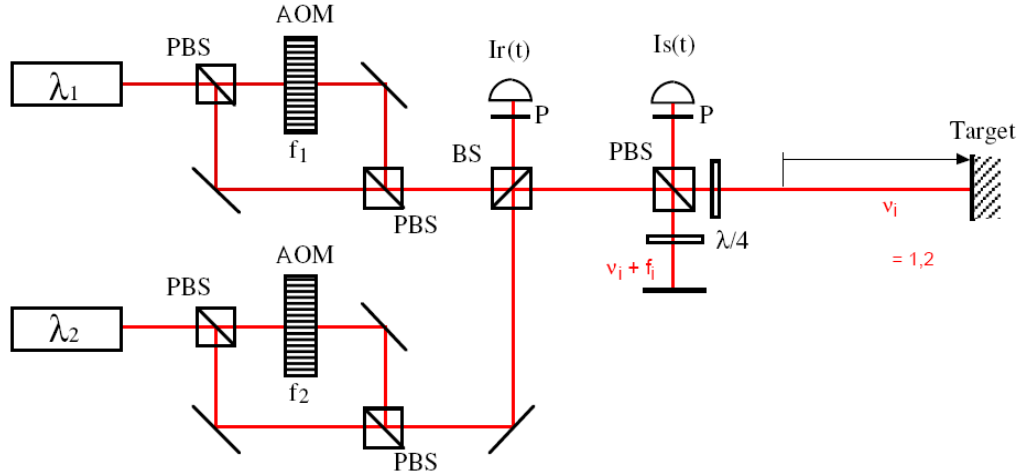


Figure 2.9: Two-wavelengths superheterodyne interferometer set-up. P: polarizers; BS: beam splitters; PBS: polarizing beam splitters; $\lambda/4$: quarter-wave plate; AOM: acousto-optical modulators [6].

2.2.2 White Light Interferometry / Coherence Radar [13]

Another way to enlarge the measurement range is to use light sources of low coherence length. Such interferometers (white-light interferometers or low-coherence interferometers) make use of the fact that only coherent light shows interference effects. If the optical path difference between the measurement and reference paths is higher than the coherence length of the light, no interference effects appear. For a path difference in the order of magnitude

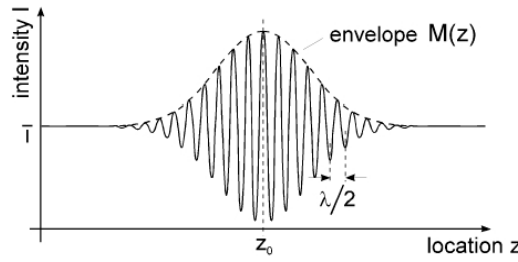


Figure 2.10: White light correlogram [11].

of the coherence length, however, interference takes place. The strength of interference is depending on the path difference between the reference and object beams as shown in Figure 2.10. Maximum intensity is achieved if the pathlengths of the reference- and the object beam are equal, thus absolute distances can be measured.

White-light interferometry with a Michelson-interferometer can be considered as the observation of the autocorrelation-function of the light signal.

2.2.3 Conoscopic Holography [57, 58]

In classical holography an interference pattern is formed from the object and the reference beams using a coherent light source. These beams propagate with the same velocity, but follow different geometrical paths.

In conoscopic holography the object and reference beams of the coherent holography are replaced by the ordinary and the extraordinary components of a single beam propagating in birefringent media (Figure 2.11).

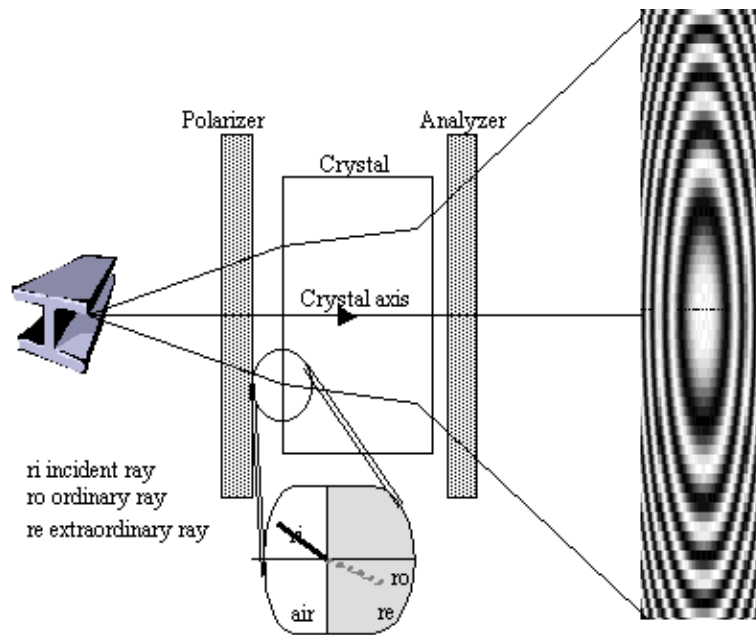


Figure 2.11: Principle of Conoscopic Holography [58].

An uniaxial (birefringent) crystal has different indices of refraction (different velocities of propagation) along practically equal geometrical paths depending on the polarization of the incident light.

The ordinary refraction index, n_O , is isotropic; the extra-ordinary refraction index, $n_E(\Theta)$, is a function of Θ , which is the angle between the optical axis of the crystal and the direction of propagation, and is given by

$$n_E(\Theta) = n_O + \Delta n \sin^2 \Theta \quad (2.14)$$

where $\Delta n = n_O - n_E$.

Thus from a single incident ray, two superposed rays emerge from the crystal with a phase difference $\Delta\varphi$ at orthogonal polarizations.

The two beams are naturally coherent and therefore this technique allows to generate holograms, even with non-coherent light. This is a key advantage of conoscopic holography.

For small angles Θ this phase difference is estimated by

$$\Delta\varphi = (2\pi/\lambda) (L/\cos \Theta) \Delta n \sin^2 \Theta \approx (2\pi L/\lambda) \Delta n \Theta^2 \quad (2.15)$$

where L is the length of the crystal and λ is the optical wavelength.

In order for both rays to interfere, an analyzer (polarizer) aligns the directions of the electrical fields.

In the system sketched in Figure 2.12, $P(x, y, z)$ is a monochromatic non-coherent point source. The system is composed of an uniaxial crystal, with its optical axis parallel to the axis of the system, between two circular polarizers, a polarizer and analyzer.

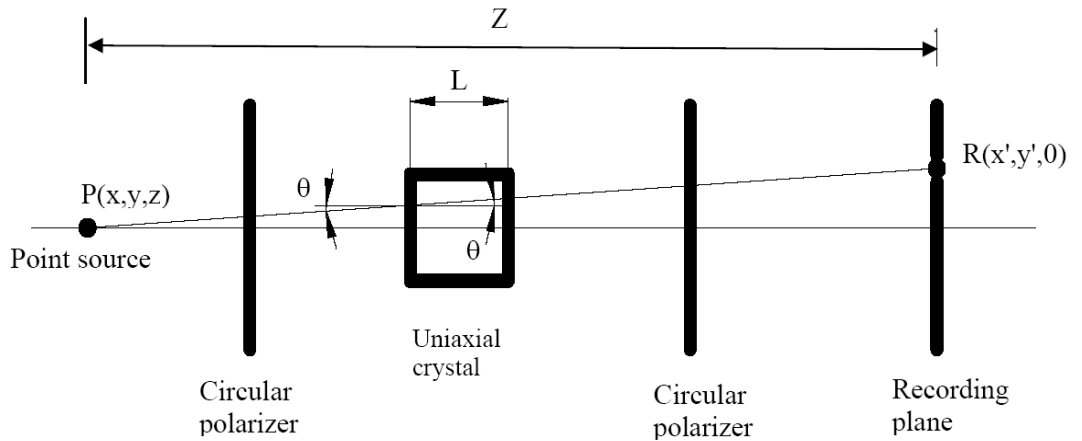


Figure 2.12: Light propagation in uniaxial crystal [57].

The light intensity is detected by a camera at the recording plane.

$$I(R, P) = I(P) \left[\frac{1}{2} + \frac{1}{2} \cos \left(2\pi L \Delta n \frac{(x - x')^2 + (y - y')^2}{z^2 \lambda} \right) \right] \quad (2.16)$$

where the distance z (the depth coordinate) enters as a parameter of the fringe spacing.

The fringe periods can be precisely measured to determine the exact distance to the point measured.

The important physical parameter is the difference of the optical path length (geometrical path length multiplied by index of refraction). In a coherent holographic system the variations of the geometrical path length are interferometrically recorded, whereas in a conoscopic holographic system the refraction index changes are recorded; the two beams have the same path. This reduces the problems of stability of the system and coherence requirements and produces holograms, even with non-coherent light. The output of the system is a hologram since it contains all the data about point P , its intensity, its lateral position and its longitudinal coordinate.

2.3 TOF (Time-of-Flight) Ranging [13]

Measurements of distances by TOF methods are indeed straight forward. If we are able to measure the time a signal has to travel from the emitter to the object and back to the receiver, the travelled distance can be computed easily (assuming the velocity of propagation of the signal is known and constant).

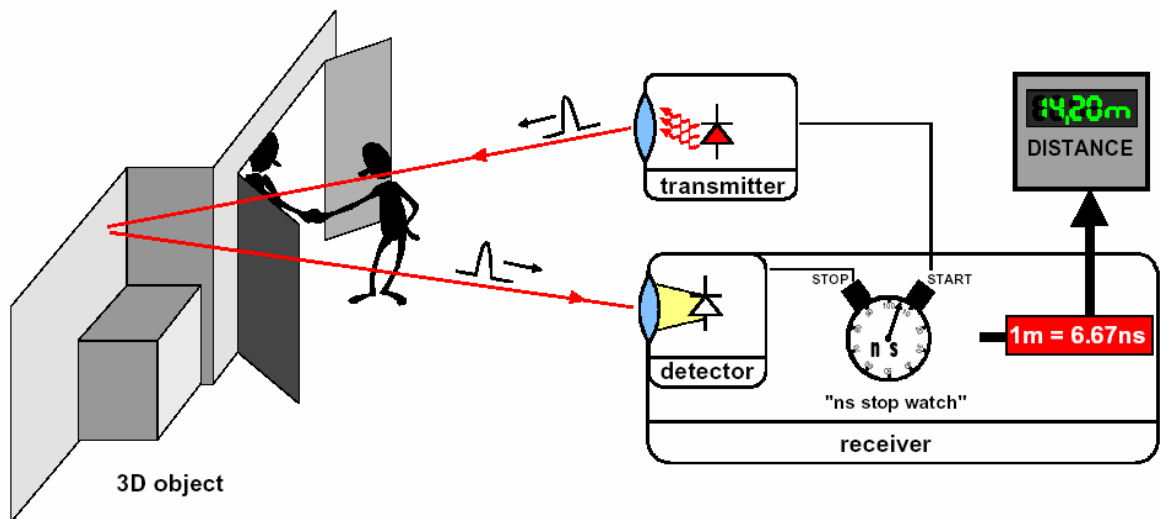


Figure 2.13: Basic principle of an (optical) TOF ranging system [13].

One can find time-of-flight measurement systems in nature's navigation solutions for dolphins and bats, which use this sensor system for both navigation and object tracking (hunting). Moreover, humans have also used TOF techniques for a long time, for example by dropping rocks down deep holes and waiting for the echo or by trying to estimate the distance of a lightning discharge by evaluating the time delay between the lightning and the associated thunder.

All these examples of TOF methods are based on the propagation of sound.

Nature favors acoustic over optical TOF measurements, due to the relative low propagation velocity ($\sim 343\text{m/s}$ at 20°C) compared to light.

Thus for technical application travel times (even for distances in sub mm ranges) can be measured directly by nowadays state of the art.

However, due to diffraction limitations ($\lambda = 0.1 \dots 1\text{mm}$), they are not suited for range

measurements with high angular resolution, at an acceptable size of the measurement system, so we will focus on optical TOF methods in the following.

A typical setup is illustrated in Figure 2.13.

An advantage of TOF compared to triangulation is the possibility of a compact sensor with the emitter located next to the receiver. This way emitted and detected beams are collinear and shadowing effects will not appear.

But there is a price to pay, and in this case it is obvious: since the propagation velocity of light is very high ($c=2.99792458 \cdot 10^8 \text{m/s}$) the travel times of the signal are very short, thus the basic problem of establishing a TOF ranging system is the realization of a sufficiently accurate time measurement. Considering the fact that the light pulse travels the path twice (forth and back) a measured time of 6.67ns corresponds to a distance of 1m, thus a time accuracy of better than seven picoseconds is required for a distance resolution of 1mm.

TOF methods can be classified by the way of modulation of the measurement beam.

In the following we will discuss Pulsed -, Continuous Wave- (AM, FM ...) or Pseudo Noise Modulation (a combination of both) methods.

2.3.1 Pulsed Modulation [13]

Pulsed light operation, as illustrated in Figure 2.13, is the most obvious method of operating a TOF system, because the time of flight is measured directly. The actual time measurement is performed by correlation of a start and stop signal with a parallel running counter. The advantage of using pulsed light is the possibility of transmitting a high amount of energy in a very short time. Thus the influence of background illumination can be reduced and a high short-term optical signal-to noise (and signal-to-background) ratio is attained while maintaining a low mean value of optical power. This is an important factor for eye-safety, which is the limiting criterion for many measurement applications. Furthermore, it reduces the demand on a very high sensitivity and signal-to-noise ratio of the detector, thus enabling long distance measurements. However, at the same time the receiver must offer high dynamics and a large bandwidth. The basic problem for the receiving path is to exactly detect the arrival time of the back-scattered light pulse. This is because (1) the optical threshold is not a fixed value but changes with background and distance of the object, and (2) atmospheric attenuation leads to dispersion of the light pulse and flattens the slope of the received pulse. It is also tricky to produce very short light pulses with fast rise and fall times, which are necessary to assure an accurate detection of the incoming light pulse. Current lasers or laser diodes, the only optical elements offering the required short pulse widths at sufficiently high optical power, still suffer from relatively low repetition rates for the pulses, which are typically in the range of some 10 kHz. Such low repetition rates drastically restrict the frame rate for TOF scanners.

Nevertheless, due to the advantages gained concerning signal-to-background ratio, most of today's TOF rangefinders are operated with pulsed modulation.

High Precision Time Measurement [25]

In the following we will give an impression how the time measurement is realized:

For resolutions down to millimeters it would be necessary to count the ticks of a clock running at about 200 GHz which is difficult to do.

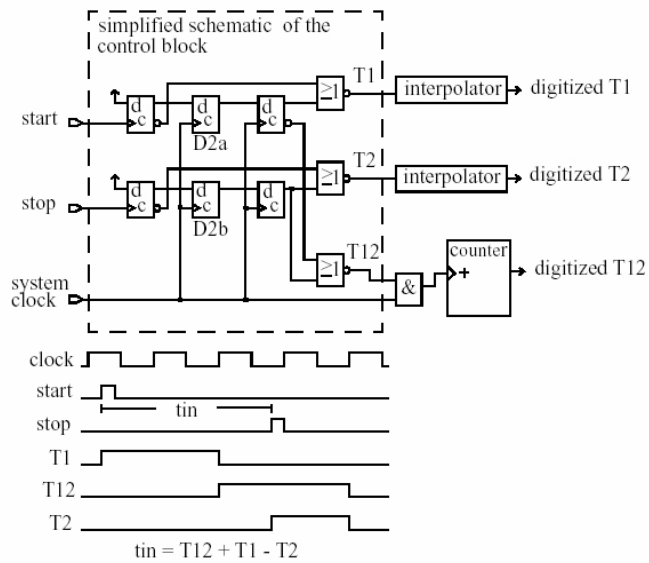


Figure 2.14: Block diagram and operating principle of a TDC [25].

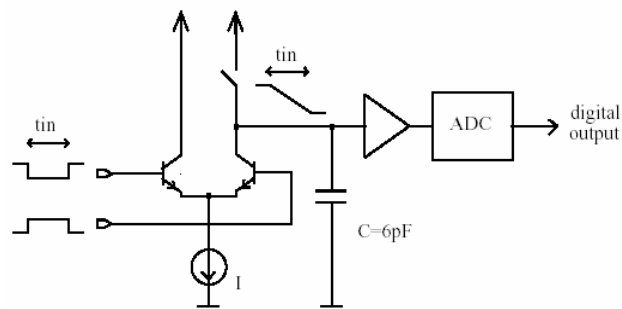


Figure 2.15: Principle of the time interpolator in the TDC described in [25].

Instead this task is usually performed by a TDC (Time to Digital Converter). A schematic diagram of a TDC is given in Figure 2.14.

The core of the TDC is a clock counter system running at about 100 MHz (higher frequencies would be hard to handle). To get a better resolution than $1/f$ the delay between clock signal and start impulse (T1) and also between clock signal and stop impulse (T2) is interpolated by a time to voltage converter (Figure 2.15), where a capacitor is discharged with constant current during the input time interval.

Since the start and stop signal is not synchronous to the clock, the error of the interpolators can be assumed as random, therefore averaging leads to higher resolutions.

In case of the special setup shown in Figure 2.14 the end mark of T1 (T2) is not taken from the first but from the second clock pulse following the start (stop).

This is done to avoid errors when the start or stop pulse occurs near the rising clock edge and the flip-flop would cause an additional propagation delay, the so called “Excess Delay”.

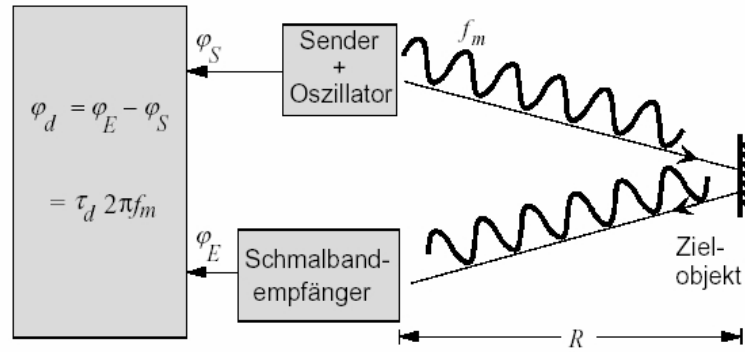


Figure 2.16: Principle of optical phase-difference time of flight method using CW [26].

2.3.2 Continuous Wave (CW) Modulation [13]

Instead of directly measuring a light pulse's turn-around time, the emitted wave is RF (radio-frequency) modulated and the phase difference between emitted and received signal is used for the calculation of the distance.

In Figure 2.16 the principle of a CW method based device is shown which uses only one modulation frequency (homodyne modulation).

CW-modulation offers the possibility of using alternative modulation-, demodulation- and detection-mechanisms. Compared to pulsed modulation a larger variety of light sources is available for this mode of operation because extremely fast rise and fall times are not required. Different shapes of signals are possible; sinusoidal waves or square waves are only two examples. For CW-modulation the phase difference between sent and received signals is measured, rather than directly measuring a light pulse's turn-around time. As the modulation frequency is known, this measured phase directly corresponds to the time of flight, the quantity of interest. The use of several modulation frequencies is known as heterodyne operation or frequency shifting. Especially heterodyne mixing offers the powerful possibility of synthetically generating beat frequencies. Thus the unambiguous distance range is increased while maintaining absolute accuracy. However, this requires relatively high bandwidth and linearity for both transmitting and receiving path.

The homodyne operation works with one single frequency and does not necessarily require a large bandwidth.

Additionally, a large variety of intelligent CW modulation techniques is available, but similar to interferometry the measurement range is limited by the unambiguous range to $\lambda/2$.

2.3.3 Pseudo-Noise (PN) Modulation [26]

Pseudo noise modulation generally is a band-spreading technique, originally developed for communication. Band-spreading means that a larger bandwidth is used than it would be necessary for transmitting the data alone, but it offers a better signal to noise ratio and the possibility of encrypting and hiding transmission of data.

A pseudo noise sequence is a random-like but periodic sequence of “+1” and “−1” and has to be formed following strict rules. In [28] a more detailed information on PN generators,

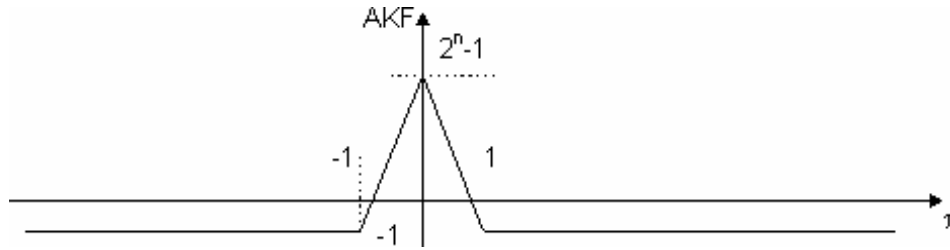


Figure 2.17: Autocorrelation-function of a PN sequence with $2n - 1$ bits [28].

properties and applications of PN modulation is given. The origin of the name pseudo-noise is that the digital signal has an autocorrelation function (Figure 2.17) which is very similar to that of a white noise signal.

The autocorrelation function is demonstrated with the following PN Sequence ($n=3$)

$$\begin{array}{r}
 Ra(\tau = 0) \\
 pn(0) = \quad +1 + 1 + 1 - 1 + 1 - 1 - 1 \\
 pn(0) = \quad +1 + 1 + 1 - 1 + 1 - 1 - 1 \\
 \hline
 Ra(\tau = 0) \quad +1 + 1 + 1 + 1 + 1 + 1 + 1; \Sigma = 7
 \end{array}$$

$$\begin{array}{r}
 Ra(\tau = 1) \\
 pn(0) = \quad +1 + 1 + 1 - 1 + 1 - 1 - 1 \\
 pn(1) = \quad +1 + 1 - 1 + 1 - 1 - 1 + 1 \\
 \hline
 Ra(\tau = 1) \quad +1 + 1 - 1 - 1 - 1 + 1 - 1; \Sigma = -1
 \end{array}$$

PN modulation is very interesting to TOF applications, because this sharp autocorrelation function allows detecting even small shifts between synchronous signals.

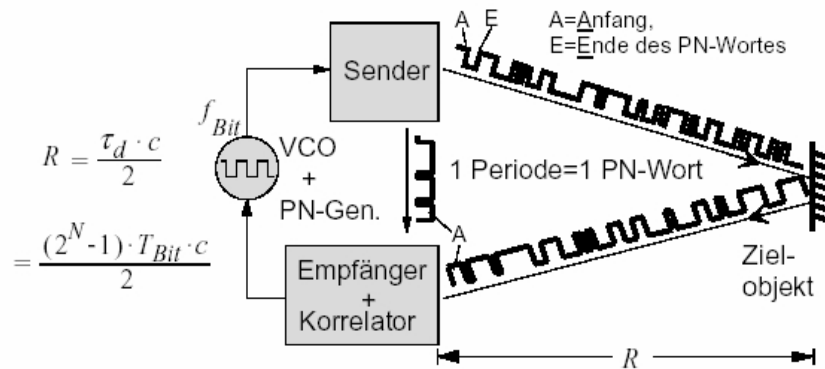


Figure 2.18: Principle of optical time of flight method using PN [26].

The principle of a PN rangefinder system is shown in Figure 2.18.

A transmitter emits the PN code which is detected by a receiver. Due to travel time the signals are shifted relative to each other. The received signal is matched with the reference signal of the PN generator, by using delay lines (DLL) or changing the frequency of the generator. These adaptations and the measurement of the autocorrelations allow the computation of the travelled distance. An advantage over the CW modulation is the possibility of having a large unambiguous range Z_{max} , that is given by the clock period of the generator $T_{Bit} = 1/f_{Bit}$ and the number of bits in the PN-sequence $(2^N - 1)$:

$$Z_{\max} = \frac{(2^N - 1) T_{Bit} c}{2} \quad (2.17)$$

With an increasing number of bits not only the unambiguous range becomes larger, the sensitivity of the correlator and so the precision of the rangefinder increases as well.

However, the number of bits should be chosen reasonably because the maximum measuring time rises in order of $((2N - 1) T_{Bit})^2$. Similar to pulsed modulation PN offers a good signal to noise relation but also needs transmitters and receivers with a high bandwidth, therefore restricting the choice of light sources and detectors.

Chapter 3

Overview of Non-Contact 3D Scanning Systems

In the previous chapter we described different methods of range-measurements, in this chapter we will discuss how we can apply these methods to 3D data acquisition.

In general range-measurement sensors allow only a 1D measurement, i.e the distance measurement of one point in the 3D scene. To capture the whole scene, we would have to use many of such sensors in parallel, but this appears to be impractical due to the large size and the enormous demand on additional electronics. Therefore in most 3D scanning systems, the object is moved relatively to the 1D detector, and every point has to be measured serially. This movement is either performed by a calibrated motion stage or the beam is deflected by a motor driven mirror system. Most of these scanning beam systems are bulky and sensitive to vibrations [13].

In the following the different systems are summarized according to their range measurement method.

3.1 3D Scanning Systems based on Triangulation [15]

Triangulation is one of the most common methods for acquiring range data. Although this technology has been in use for over two decades, its speed and accuracy has increased dramatically in recent years with the development of geometrically stable imaging sensors such as CCD's and position sensitive (lateral effect) photodiodes.

Researchers and manufacturers have used optical triangulation scanning in a variety of applications. In medicine, optical triangulation has provided range data for plastic surgery simulation, offering safer, cheaper, and faster shape acquisition than conventional volumetric scanning technologies. In industry, engineers have used triangulation scanners for applications that include -as an example- postal package processing and printed circuit board inspection. Triangulation scanners also provide data to assist computer graphic applications, such as digital film production.

Triangulation systems are available for distance measurement ranges from mm-range (depth of focus) to 100km range (photogrammetry).

3.1.1 Photogrammetry (Passive Triangulation)

Photogrammetry is one of the oldest methods for 3D data acquisition. In the photogrammetric process an object is reconstructed by a set of photos taken from different points of view. Photogrammetry is based on passive triangulation and will be discussed in more detail in the next chapter.

3.1.2 Depth from Focus and Defocus [27]

This technique obtains depth information by actively controlling camera parameters. It is a passive monocular triangulation method, therefore in contrast to stereo methods (like photogrammetry) no correspondence between two or more images has to be established. There are two distinct scenarios for using depth focus information for depth recovery:

Depth from Focus

The distance to one point is determined by having a sequence of images in progressively better focus.

The main issue is to find a measure for the quality of the focus. An ideal measure of the quality of the focus should be expressed by a function that is unimodal, monotonic and should reach its maximum only if the image is focused. With such a criterion for the quality of the focus, the optimal focus-interval is found by a quick search algorithm, like the Fibonacci-search [65].

Depth from Defocus:

By taking a small number of images under different lens parameters, the depth at all points of the scene can be determined. This method uses direct relationships among the depth, the camera parameters and the amount of blurring in the images. Since the camera parameters can be calibrated, the depth can be expressed by the amount of blurring change correspondingly. However, as with passive stereo vision the scene must have enough texture for the blurring to be detected.

The system demonstrated in [27] can determine the depth within the measurement range ($\sim 2,5\text{m}$) with a relative accuracy of $1/200$.

3.1.3 Structured Light (Active Triangulation) [54, 34]

Structured light scanners project a known light pattern onto the 3D surface of the object to be modelled. In its simplest variant (apart from spot illumination) this consists of illuminating the object (being scanned) with a stripe of light and observing it with a detector (typically a CCD camera) placed at a known angle with respect to the light source. The side-to-side wiggles of the observed stripe correspond to the shape of a contour on the object; mathematically, we may determine the 3D locations of points on this contour by computing the intersections between the camera rays and the plane of the projected light stripe (refer to Figure 3.1).

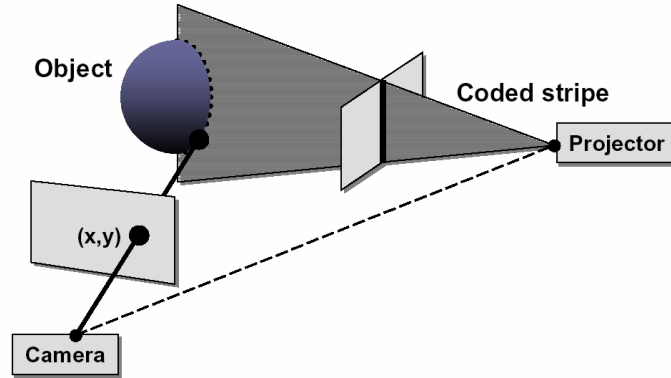


Figure 3.1: Layout of a single-camera, single-source triangulation system. The 3D positions of points on the object are determined by the intersection between the camera ray and the plane of light produced by the illumination source [34].

Structure of Illumination [15]

The structure of illumination can take a variety of forms. A beam of light forms a spot on a surface and provides a single range value. By passing the beam through a cylindrical lens, a light stripe can be projected onto an object to collect a range profile (Figure 3.1).

It has also been tried to project multiple spots or multiple stripes onto an object for more parallelized shape acquisition, though multiple steps are usually required to disambiguate the detected light patterns.

Typically LCD (Liquid Crystal Display) projectors are used for the projection of these patterns.

When the reflected light is imaged onto a sensor by lenses, the single point and stripe illuminations offer the advantage that at any instant all intersections of the light with the object must lie in a plane. Since lenses image points in a plane to points in another plane, the sensor can be oriented to keep the beam or sheet of light in focus, thus reducing depth of field problems. When the focal plane is tilted, the image plane must also be tilted so as to satisfy the Scheimpflug-condition [17]:

$$\tan \alpha = M \tan \Theta \quad (3.1)$$

where α and Θ are the tilt angles of the focal and image planes, respectively, as shown in Figure 3.2, and M is the magnification on the optical axis.

The resulting triangulation geometry has the property that the focal-image and lens-planes all intersect in a single line.

Multiple-point and multiple-stripe systems generally cannot take advantage of this optimal configuration, because the illumination usually does not lie in a single plane.

Type of Illumination [15]

The type of illumination can be either coherent or incoherent. Coherent light sources such as lasers offer several distinct advantages over their incoherent counterparts. First, lasers can be held in tight focus over a long range. Second, since laser light is tuned to a single

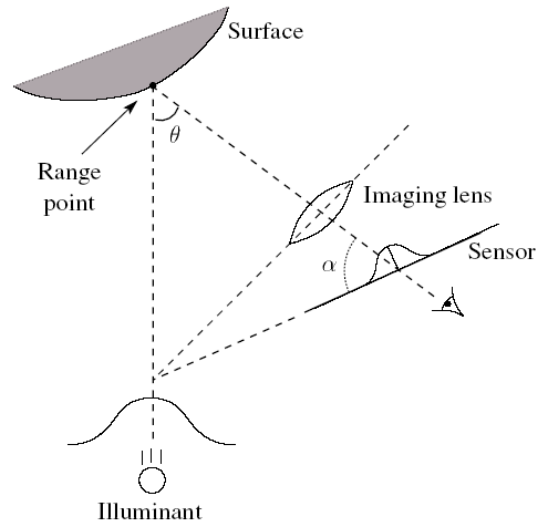


Figure 3.2: In order for all points along the center of the laser sheet to be in focus, the angles Θ and α are related to one another by Eqn 3.1 [15].

wavelength, the sensor can be coated with a bandpass wavelength filter, decreasing the sensitivity to ambient light. In addition, the optical system does not have to be corrected for chromatic aberrations, and elements such as prisms can be simply used. Usually lasers used in triangulation also do not have problems with heat dissipation, whereas incoherent illumination sources of sufficient power frequently do. The disadvantages of using lasers are laser speckle (random coherent interference due to surface roughness) and the need for special safety precautions for lasers operating at visible and invisible (ultraviolet) wavelengths.

Object-Capturing [34]

Having a spot or a stripe as an illumination pattern the object must be moved with reference to the scanner, to obtain dense 3D information. Illumination patterns like "multiple stripes" allow to capture a scene with a single shot. For 3D model acquisition (complete models of rigid objects), the object must be moved relative to the scanner (or the scanner moved relative to the object) in order to obtain and integrate views of the object from all sides.

In both cases this movement is usually performed by a calibrated motion stage. There are also hand-guided scanning systems where the position (all 6 degrees of freedom: 3 linear coordinates and 3 angles of orientation) of the triangulation sensor is captured by a tracking system. This tracking system can be acoustical or optical, which involves that the operator must not obscure the sensor, or magnetically, like in the Polhemus-Fastscan system.

Polhemus Fastscan [23]

The Polhemus Fastscan is a hand-guided laser triangulation scanner.

The device consists of a laser stripe projector and a CCD camera to obtain depth information of the illuminated area. The scanner is moved over the object of interest by hand. The position of the scanning head is acquired by a magnetic tracking system collecting all 6 degrees of freedom. The magnetic tracking system has the advantage that it needs no line of sight between the scan head and the detector (usually fixed to the object). However, the

magnetic tracking system can be disturbed by metallic objects in the surrounding area, thus the scanner is restricted to non-metallic, opaque materials.

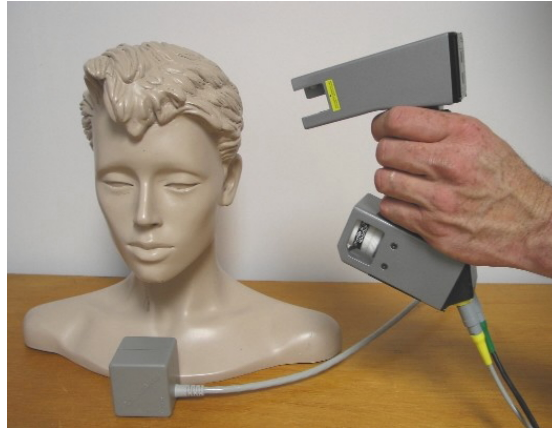


Figure 3.3: Polhemus Fastscan: Handguided scanhead with laser stripe projector, CCD-camera, magnetic tracking system (transmitter); the tracking system-receiver is placed near the object [23].

A similar system is available from Steinbichler [66], which uses an optical instead of the magnetic tracking system.

Line Triangulation 3D Scanner in our Laboratory [64]

In his Diploma Thesis S. Pelech developed a 3D Laser-Scanning System based on Line Triangulation. He built a scanner with high precision and low system costs. The scanner uses a cylindrical lens to widen up the laser beam, and projects it onto the object of interest, that is mounted on a rotation- and translation stage. The stripe of light is orientated perpendicular to the axis of rotation as shown in Figure 3.4. The costs of the scanner were kept low by the use of off-the-shelf-components.

The imaging-sensor is a standard CCD video-camera. By the use of a subtile calibration process the influence of the lens distortion was eliminated and thus an excellent resolution of $\sim 0.1\text{mm}$ was achieved. A further contribution to attain this high resolution is the rather large triangulation angle of 45° . This however makes the scanner vulnerable to shadowing effects. In order to minimize those, the laser-line is observed from two directions. For a consistent calibration and low overall costs this is done with the help of a rotating (central) mirror, instead of two CCD-cameras.

The scanner has a cylindrical working volume of 400mm in length with a diameter of 200mm, and achieves a precision of $\sim 0.1\text{mm}$; a slice is scanned in ~ 1.5 min. The scanner thus competes commercial systems but can be inexpensively built using standard components.

We will later use scan data-sets produced with this scanner to verify our software and to compare them with our scanning system.

Cyberware Desktop 3D Scanner Model 15 [67]

This scanner is also based on laser light-sheet triangulation, and is closely related to the system described above. The main difference is that here the laser-line and the axis of

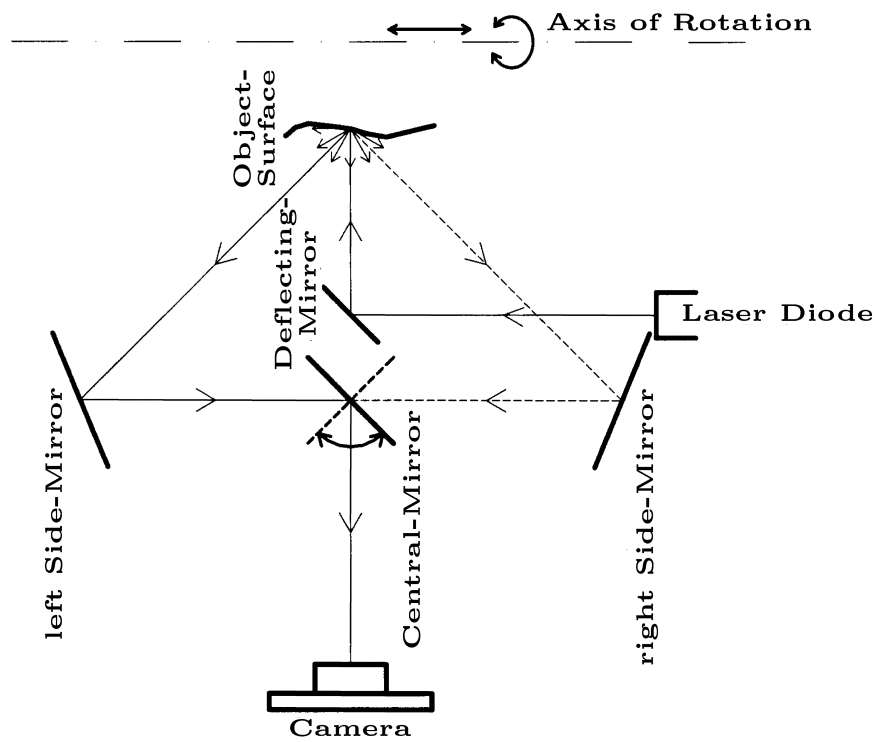


Figure 3.4: Optical path and directions of motion [64].

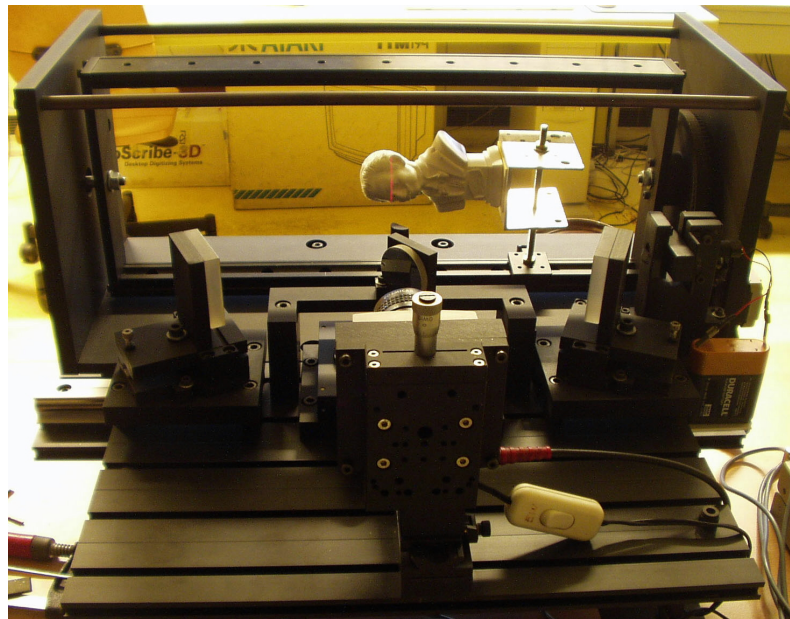


Figure 3.5: Photograph of the prototype system.

rotation are parallel. The scan-head is moved by a translation stage to obtain a range image. In order to acquire a range image from another point of view, the turntable with the object is rotated. The scanner samples ~ 15000 points per second, and has a resolution of $\sim 0.3\text{mm}$, but it is limited to rather small objects ($250\text{mm} \times 150\text{mm} \times 75\text{mm}$). A complete object scan lasts for about 1-2h.



Figure 3.6: Cyberware Desktop 3D Scanner Model 15 [16].

3.2 3D Scanning Systems based on Interferometry

As already mentioned above in the chapter concerning range measurement techniques, single-wavelength interferometry is not very suitable for 3D-scanning applications, since it suffers from a small unambiguous range and the measurement can be disturbed by rough surfaces (speckle effect), whereas other interferometric methods are quite common.

With multiple-wavelength interferometry scanning-beam-systems can be realized that offer a precision of a few $10\mu\text{m}$ over a measurement range of $\sim 200\text{mm}$.

White light interferometric methods like the Coherence-Radar allow highly accurate absolute distance measurements and find their applications for example by measuring circuit boards in quality control or in dermatology to determine the roughness of human skin [12].

Sensors based on Conoscopic Holography provide high precision measurements and are robust against temperature drifts, vibrations, surface texture variations and shadowing effects [59].

3.2.1 Multiple Wavelength Interferometry (MWI) [7]

Distance sensors based on TOF methods allow for large measurement ranges, but their resolution is limited to a few tenths of a mm. Since the precision of triangulation sensors is determined by their triangulation-angle, they may not be suitable for surfaces with holes or steep edges, whereas MWI systems benefit from a comparably high distance resolution and their collinear measuring beams. The measuring range and precision of MWI systems can be adapted to specific applications by a proper choice of the virtual wavelength. Theoretically the distance-resolution of MWI is only limited by the roughness of the sample surface. A system based on heterodyne modulation as described in section 2.2.1 is presented by Trautner

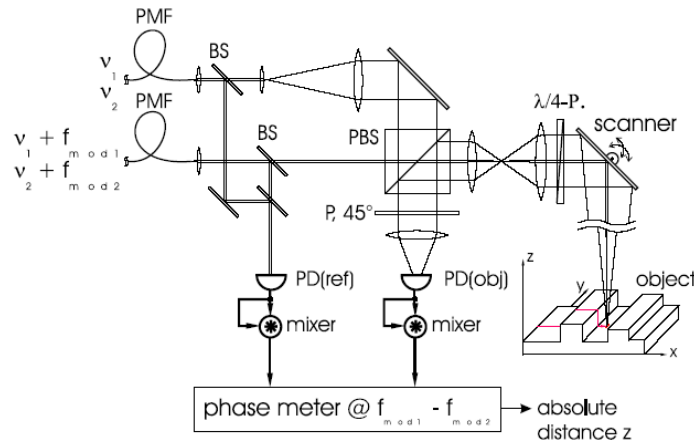


Figure 3.7: Measuring head of a multiple-wavelength range-scanner [7].

et al in [7]. There two different configurations are examined, one with a virtual wavelength of $\Lambda = 2\text{mm}$ and a second with $\Lambda = 347\text{mm}$. For the larger wavelength they were able to achieve a measurement resolution of 0.7mm and $60\mu\text{m}$ for the smaller one. They proposed that by increasing the intensity of the light source and an improved setup a resolution down to $10\mu\text{m}$ should be possible. Furthermore they are planning to extend their system by using two virtual (three optical) wavelengths. The larger wavelength would be used for a coarse measurement of the sample and detailed information would be gathered by the smaller one. Thus in combination of both a measurement range of $\sim 200\text{mm}$ and a resolution of $10\mu\text{m}$ should be achieved.

3.2.2 Conoscopic Holography [58, 59]

As stated earlier, Conoscopic Holography is a holographic technique based on light propagation in uniaxial crystals. Since its discovery in 1985 at the California Institute of Technology, it was targeted as a general-purpose, three-dimensional, optical sensor. Optimet produces sensors which can be adapted to various working ranges by using different lenses. Working areas are from $\sim 0.5\text{mm}$ to $\sim 250\text{mm}$ and a resolution of about $1/8000$ of the working range can be attained. An illustration of such a system is given in Figure 3.8.

The probe emits a laser beam that is reflected by a beam splitter and hits the object being measured. Scattered light returns from the object through the beam splitter and birefringent crystal and is then detected by the sensor's CCD camera. The birefringent crystal modifies the speed of each light ray differently in accordance with its angle with respect to the crystal axis. This creates a high contrast fringe pattern on the CCD camera. The angle of the light ray is a function of the distance between the reference plane and the laser spot projected on the object. The distance can therefore be determined by analyzing the characteristics of the created pattern.

A typical application for these high-precision single-point measurement sensors are non-contact CMMs (Coordinate Measurement Machine), where the probe is rastered over the sample by motion-stages.

These non-contact CMMs have the advantage over contact CMMs that they are also suitable

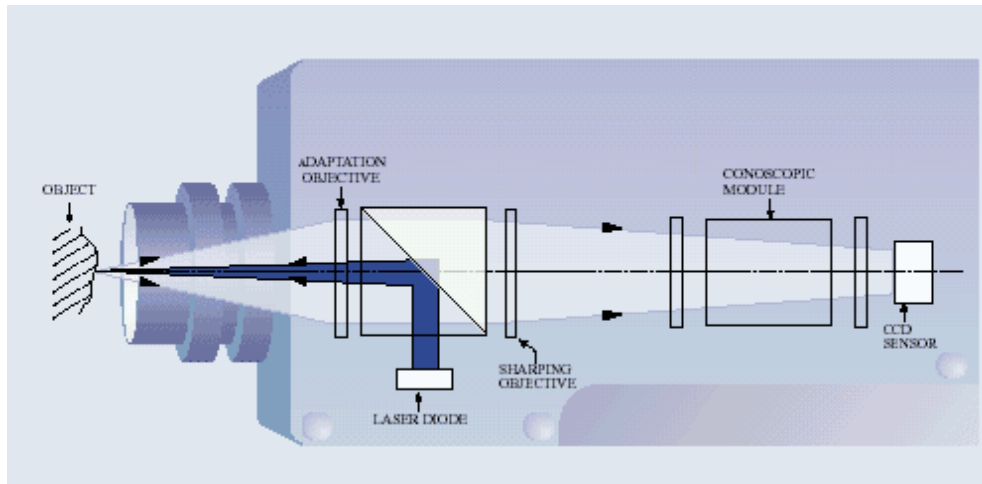


Figure 3.8: Schematic of the Conoprobe (a sensor based on Conoscopic Holography) [59].

for soft materials and higher scanning speeds are possible, since one axis is measured in motion, and there is no need for a slow approach of the measurement tip to the object. There are also sensors available that provide a 2D profile in a single run, thus a full 3D surface area profile is obtained in a single axis scan. They are used, for example, in in-process inspection systems, that observe objects on a conveyor belt.

3.2.3 Coherence Radar [9, 12]

A coherence radar is an optical 3D range sensor based on white light interferometry. It is basically a Michelson interferometer (Figure 3.9).

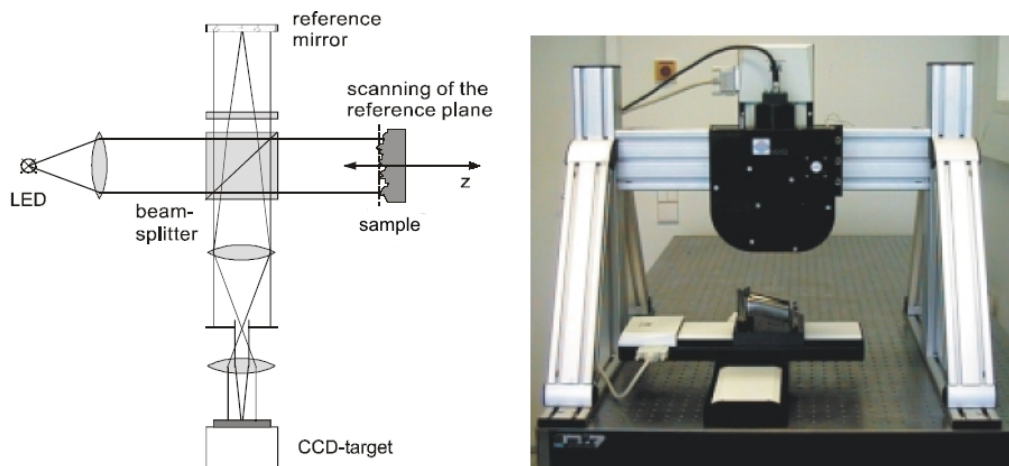


Figure 3.9: Experimental setup of the Coherence Radar (left) [12], Sensor in a portal setup (right) [10].

The object of interest and the reference mirror are illuminated by a plane wave. The illumination source is chosen to have a large spectral bandwidth, thus the coherence length of the source is short and good contrast fringes will be obtained only when the two paths of the interferometer are closely matched in length.

The in-depth scan is done by moving the surface of the sample along the optical axis. During this movement the interference contrast is observed for each camera pixel, and the vertical peak position is detected.

Thus the range measurement is done in parallel for each camera pixel and the scanning time only depends on the measurement-range.

In contrast to triangulation systems the measurement accuracy is independent of the distance between object and sensor, also no shadowing effects arise, since the measurement uses collinear beams. An advantage over coherent interferometry is the ability to scan also rough surfaces.

The system described in [10] achieves a resolution of $1\mu\text{m}$, features vertical scan velocities $4\text{--}70\mu\text{m/s}$, and the measuring field is scalable from 1mm^2 to 100mm^2 .

Note that by replacing the reference mirror with a master object the direct measurement of deviations between object under test and master object is possible with a Coherence Radar as well.

3.3 3D TOF-Scanning-Systems

3.3.1 Acoustic Position Tracker

Acoustic positioning systems benefit from the relatively low propagation velocity of sound. Considering a sound velocity of 340m/s , we need to measure times of $\sim 0,3\mu\text{s}$ to scan details as small as 0.1mm , which can be performed easily by counting pulses from a MHz oscillator. On the other hand, due to the large wavelength, "scanning sonic-beam systems" are not available, because of their low spatial resolution. Thus the only acoustic scanning systems are position trackers. A typical arrangement is shown in Figure 3.10.

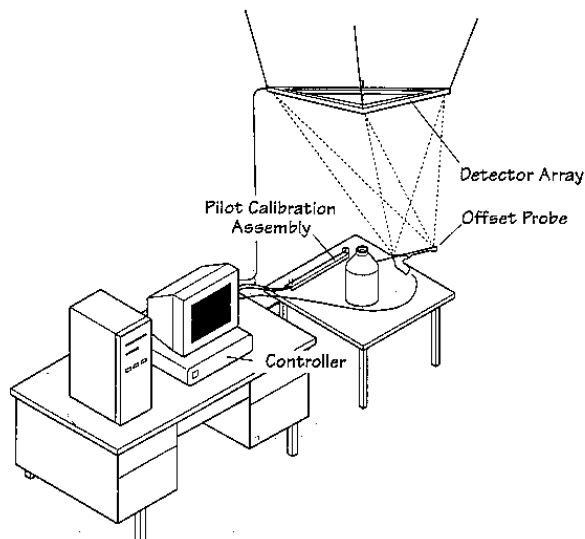


Figure 3.10: Acoustic Position Tracker Setup [55].

It consists of a probe that emits ultrasonic acoustic waves which are received by a detector array. To compute the position and the direction of the probe, it uses at least two emitters, working at different frequencies or emitting their signals in a special pattern.



Figure 3.11: Freepoint 3D Scanner [56].

The position of the probe is computed by the travel-time of the signals. Travel time measurements are made within a defined time-window to minimize the effect of reflections from nearby surfaces.

It is possible to build scanners with a scanning-volume of several m^3 achieving a resolution down to 0.1mm.

This is quite a good performance, but there are a few limitations to ultrasonic digitizers: One of the problems associated with ultrasonic devices is that the digitizing probe needs to be in the line of sight of the detector array. If the operator or model obscures one of the two beam-paths the system it cannot operate accurately.

Another problem is that the sound-propagation velocity is affected by atmospheric conditions, like humidity, temperature and air pressure.

As a solution the Freepoint 3D Scanner from GTCO [68] features an automatic calibration procedure to determine the velocity of sound:

As shown in Figure 3.10 there is a calibration assembly nearby the scanner. Each time the position of the probe is tracked, the time of flight is compared to a reference measurement made by a separate emitter and detector placed at a well known distance.

The Freepoint 3D Scanner (Figure 3.11) is capable of scanning a volume up to 2,4m x 2,4m x 4,8m with a resolution of 0.1mm.

3.3.2 “Scanning beam” 3D Scanners (optical TOF) [22]

A laser range measurement system can be used to gather entire 3D scenes by sequentially scanning a beam across the observed object.

A high-end product based on this principle is manufactured by Riegl [22], a small Austrian company that has a long experience in optical rangefinding technology.

The components of the scanning system (refer to Figure 3.12) and the principle of operation according to [22]:

The rangefinding electronics (1) of the 3D scanner is optimized in order to meet the requirements of high-speed scanning (fast laser repetition rate, fast signal processing, and high-speed data interface).

The vertical deflection (“line scan”) of the laser beam (2) is realized by a polygon (3) with a number of reflective surfaces. For high scanning rates and/or a vertical scan angle ϑ up to 90° , the polygonal mirror rotates continuously at adjustable speed. For slow scanning rates

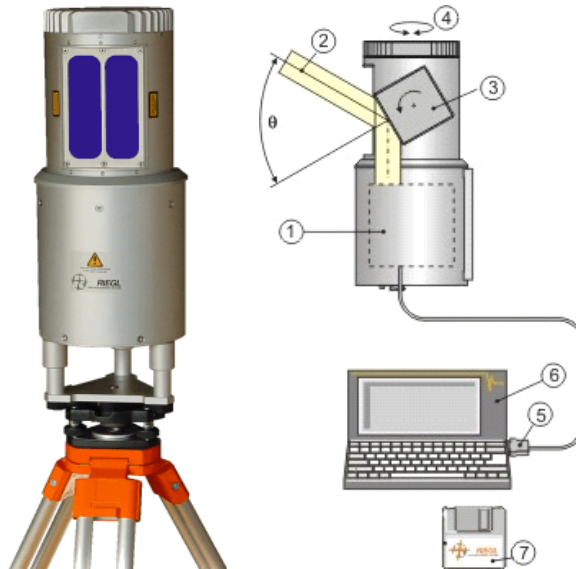


Figure 3.12: Scanning head and schematic of the Riegl 3D Scanner [22].

and/or small scanning angles, it is oscillating linearly up and down. The horizontal scan ("frame scan") is provided by rotating the complete optical head (4) up to 360° . The gained information -range, angle, signal amplitude, and target color- is provided via a parallel data output.

The scanner has a measurement range from 2 to $\sim 400\text{m}$ (depending on the surface reflectivity) an averaged distance resolution of 5mm, an angular resolution of 25 mrad and is capable of scanning ~ 10000 pts/s.

As a special feature the scanner is also available with a true color channel, allowing to determine the color of each laser measurement. In Figures 3.13, 3.14 one can get a few impressions of the performance of this scanner. The Colosseum in Rome was scanned in field-of-view of $80^\circ \times 80^\circ$ yielding a total number of 1.2×10^6 Points. The entire scan was accomplished in 3min 25s.

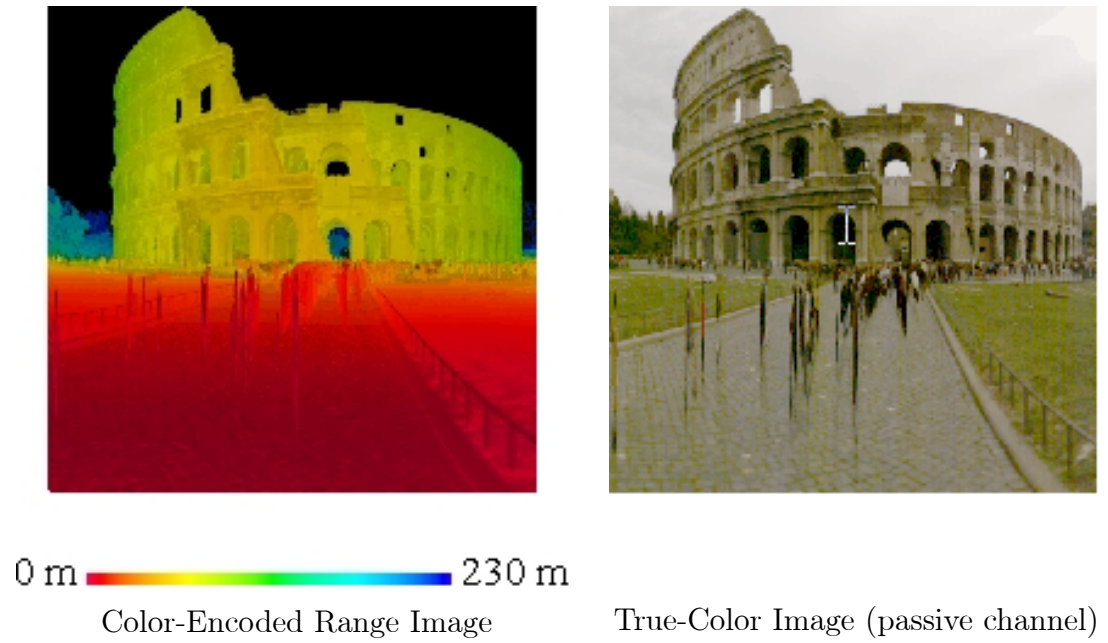


Figure 3.13: Colosseum scanned with the Riegl 3DScanner [22].

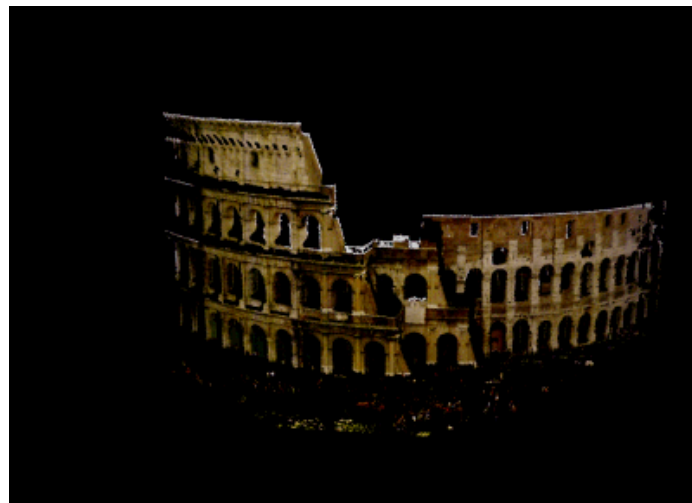


Figure 3.14: Rendered 3D-scene with true-color-texture [22].

Chapter 4

Selected 3D-Scanning Systems

In the following we will present three scanning systems in more detail.

- Terrestrial or Close-Range Photogrammetry
- a Real-time Structured-Light range-scanner
- and a 3D Time-of-Flight-camera

Photogrammetry was selected because of its importance and widespread applications. The two latter were chosen, because of they offer a new approach to 3D-scanning and may influence future scanner designs.

4.1 Terrestrial or Close-Range Photogrammetry [31, 32]

Photogrammetry is a technique of making reliable measurements of objects (2D or 3D) by the use of (usually multiple) images. According to [31] close range- or terrestrial photogrammetry is the application of photogrammetry where the measured distances are small ($\sim < 300\text{m}$) compared to those of air or satellite picture processing. There is a further classification based on the number of images to process: singleview- (not suitable for 3D measurements) stereo- and multiperview-photogrammetry. The geometrical background of photogrammetry is central perspective mapping. The following section will provide a short overview of the photogrammetric process (refer to Figure 4.1).

To reconstruct the object of interest from the photographs, it is necessary to have a network of reference points. If the object lacks significant points, which are easily identified, it has to be signalled, which is usually done by retroreflective stick-on targets. The position of these control points have to be measured in order to obtain an “object-coordinate system”, a reference system between the photographs and the real world, which will be used later to determine the camera positions, view angles, and for scaling.

The main thing of data collection is to take pictures of the object from two or more different points of view.

Pre-processing involves film processing, enlargement and digitizing (obviously only necessary if analogue cameras were used), storage and the calculation of the object coordinate system from the reference points.

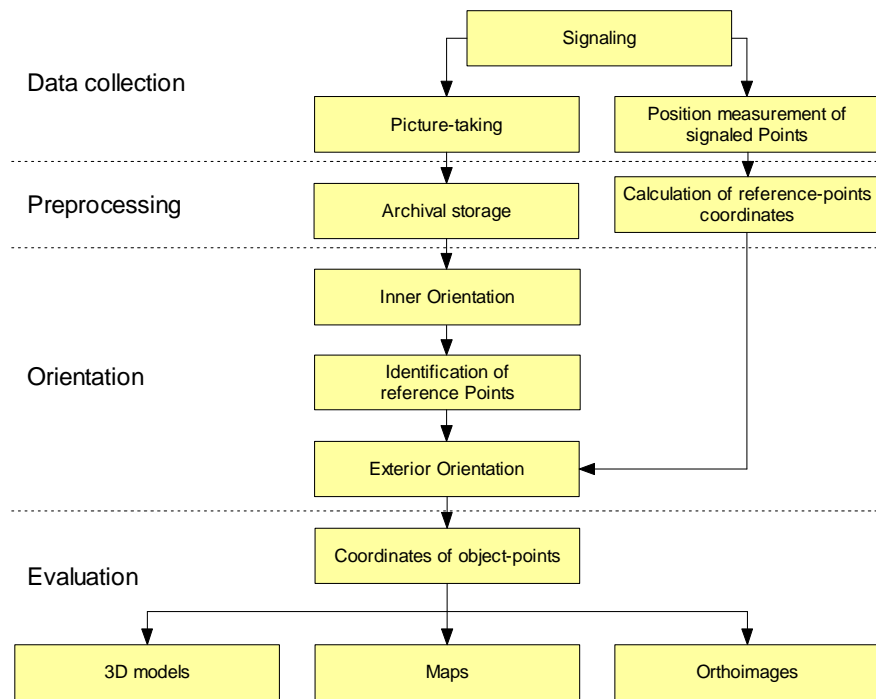


Figure 4.1: The photogrammetrical procedure according to Luhmann [31].

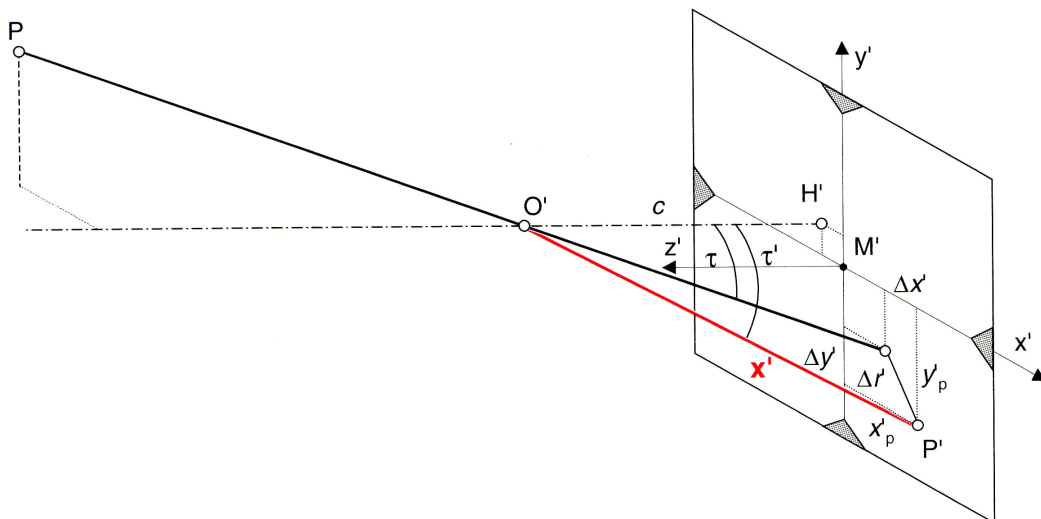


Figure 4.2: Parameters of inner orientation [31].

Calculation of the inner-orientation (refer to Figure 4.2) means, that the position of the image with respect to the centre of projection is determined ($H(x_0, y_0), c$) and lens-distortion ($\Delta x', \Delta y'$) is compensated.

The main part of the lens distortion is radial-symmetric; a smaller radial-asymmetric and tangential part is due to a misalignment of the lenses in the objective.

Camera calibration must be associated with each image. This identifies which camera/lens combination was used when actually the image was taken, and it provides the analytical software with critical information such as the focal length of the camera, the size of the image, and distortion parameters of the lens. Camera calibration is accomplished by using a number of test photographs which are taken of a specially designed calibration field.

To compute the orientation and position of the cameras relative to the object (exterior orientation) at least three control points have to be identified in each picture.

The parameters that have to be calculated are the translation vector \mathbf{X}_0 (X_0, Y_0, Z_0) and the angles of the rotation matrix $\mathbf{R}(\omega, \varphi, \kappa)$.

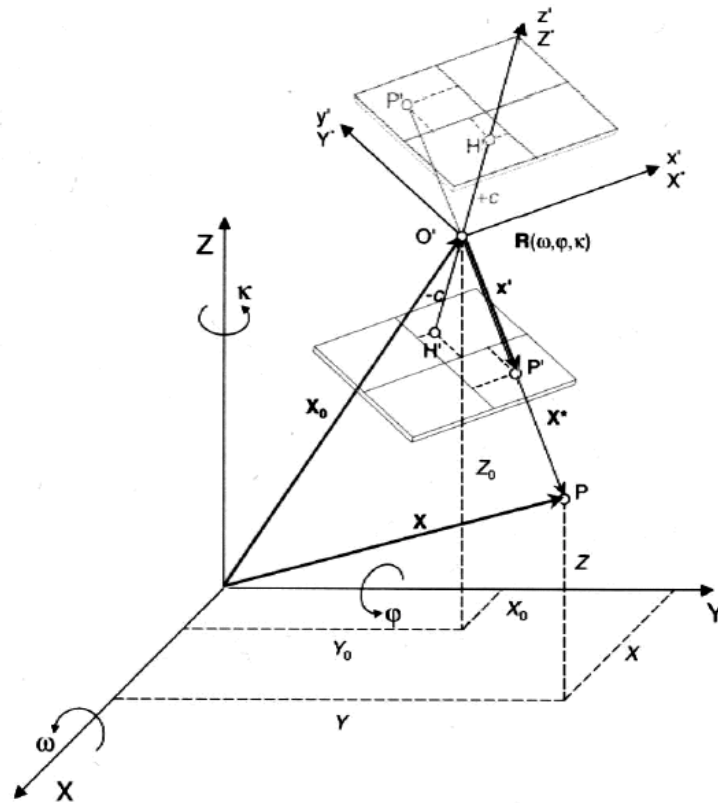


Figure 4.3: Central projection and exterior orientation [31].

Equation 4.1 explains how an image-point $\mathbf{P}'(x', y', -c)$ is mapped to an object point \mathbf{X} :

$$\mathbf{X} = \mathbf{X}_0 \frac{1}{m} \mathbf{R}(\omega, \varphi, \kappa) \begin{pmatrix} x' - x'_0 \\ y' - y'_0 \\ -c \end{pmatrix} \quad (4.1)$$

where $\mathbf{R}(\omega, \varphi, \kappa)$ is the Rotation-matrix, and m is the unknown scale parameter.

It is for this scale parameter, that it is not possible to reconstruct the object coordinates from a single picture. The scale parameter m can be expressed by Z leading to the following “collinear- equations”, where r_{ij} are the components of the Rotation-matrix $\mathbf{R}(\omega, \varphi, \kappa)$.

$$X = X_0 + (Z - Z_0) \frac{r_{11}(x' - x'_0) + r_{12}(y' - y'_0) - r_{13}c}{r_{31}(x' - x'_0) + r_{32}(y' - y'_0) - r_{33}c} \quad (4.2)$$

$$Y = Y_0 + (Z - Z_0) \frac{r_{21}(x' - x'_0) + r_{22}(y' - y'_0) - r_{23}c}{r_{31}(x' - x'_0) + r_{32}(y' - y'_0) - r_{33}c} \quad (4.3)$$

A known exterior orientation allows the various photographs to be mathematically tied together and tied to the control point coordinate system.

If the orientation parameters of all photographs are known and homologous (identical) points are found in two or more images, this point can be determined in the object coordinate system by spatial intersection from two or more images.

In practice the most difficult step is to identify homologous points in the images. (This is usually denoted Correspondence Analysis.)

4.1.1 Correspondence Analysis [69]

In [31] Correspondence Analysis is stated to be an “ill-posed problem” because due to occlusion points need not be visible in all pictures, ambiguous object structures lead to several (possibly matching) points, and object texture with low contrast causes unstable and noise sensitive solutions.

A method to avoid this problem is structured-light-scanning; it will be treated in the next chapter.

The strategies of Correspondence Analysis can be listed as follows:

- In area-based matching the correspondence between two image areas is determined by the similarity of their grey values. The cross correlation and the least square correlation are the well-known methods for area based matching.
- The feature-based matching determines the correspondence between two image features (points, edge elements, short edges or lines and small regions).
- In structural-matching/relational-matching not only image features but also their relation (is above is parallel, ...) are used for identifying corresponding points.

Hierarchical methods are used in many matching algorithms in order to reduce the ambiguity problem. They are employed (like sifters) from coarse to fine. The results achieved on one resolution are considered as approximations for the next finer level. For this task images are represented in a variety of resolutions, leading to “image pyramids”.

Due to central perspective mapping there is a powerful constraint that limits the search area for homologous points: for the special case of stereo-photogrammetry (two identical cameras with a view direction normal to their base) the Correspondence Analysis has to be carried out only in the x-parallaxes of the point of interest from one picture to the other. Thus if the pictures were taken from arbitrary but known positions, the search region can be limited to a small rectangle as shown in Figure 4.4. Also the number of ambiguous points can be reduced by epiplanar-geometry if there are multiple images covering the same region of interest (refer to Figure 4.5).

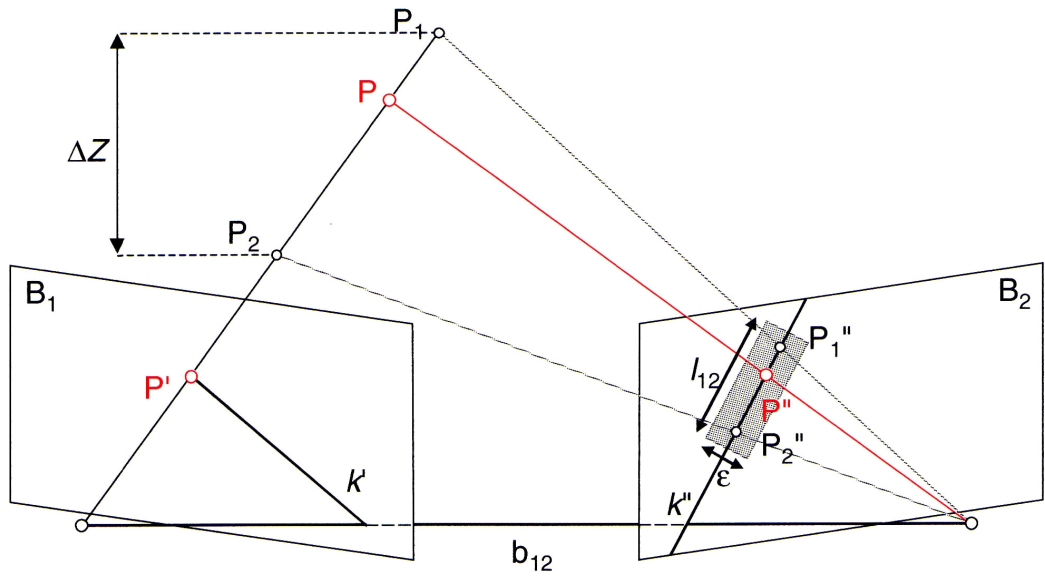


Figure 4.4: Epiplanar-geometry [31].

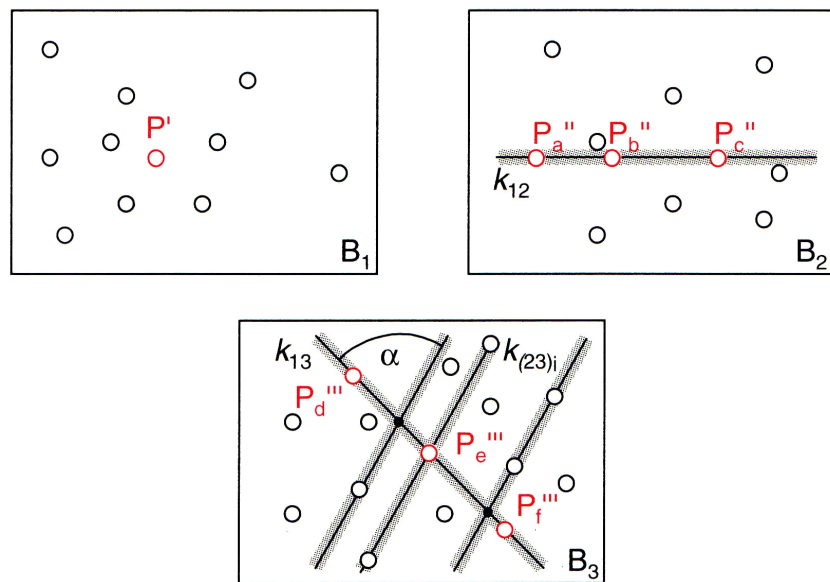


Figure 4.5: Correspondence-Analysis in image-triple using "Kernlinien" [31].

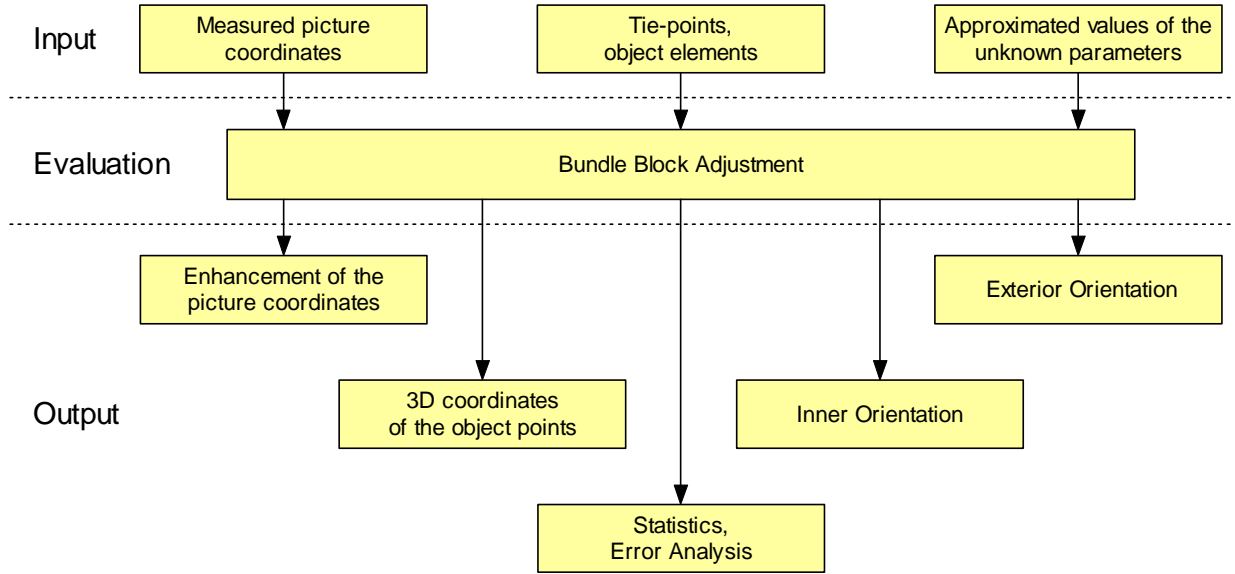


Figure 4.6: Data flow in Bundle Block Adjustment [31].

4.1.2 Bundle Block Adjustment [31]

Bundle block adjustment first suggested by Schmid in 1958 [72] is a different approach where exterior and interior orientation parameters and object coordinates of tie points are calculated by least squares adjustment. It is the dominating method nowadays because the simultaneous determination leads to the highest accuracy and the demand for high computational power can be well satisfied nowadays.

For every point in a picture we have 2 observations:

$$x' = x'_0 - c \frac{r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} \quad (4.4)$$

$$y' = y'_0 - c \frac{r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} \quad (4.5)$$

which we need to calculate the following unknowns.

- With each camera there are (except for lens distortion) 3 undetermined values (x_0, y_0, c) from inner orientation,
- with each picture 6 unknown values $(X_0, Y_0, Z_0, \omega, \varphi, \kappa)$ from the exterior orientation and
- 3 unknowns (X, Y, Z) for each point in a picture:

This over-determined equation system is linearized and solved iteratively by least squares. An example of a successful photogrammetric reconstruction is given in Figure 4.7. An accuracy of $\sim 0.25\text{mm}$ on critical surfaces and points, $\sim 5\text{mm}$ on other surfaces was achieved and the object was modelled by hand during post processing.

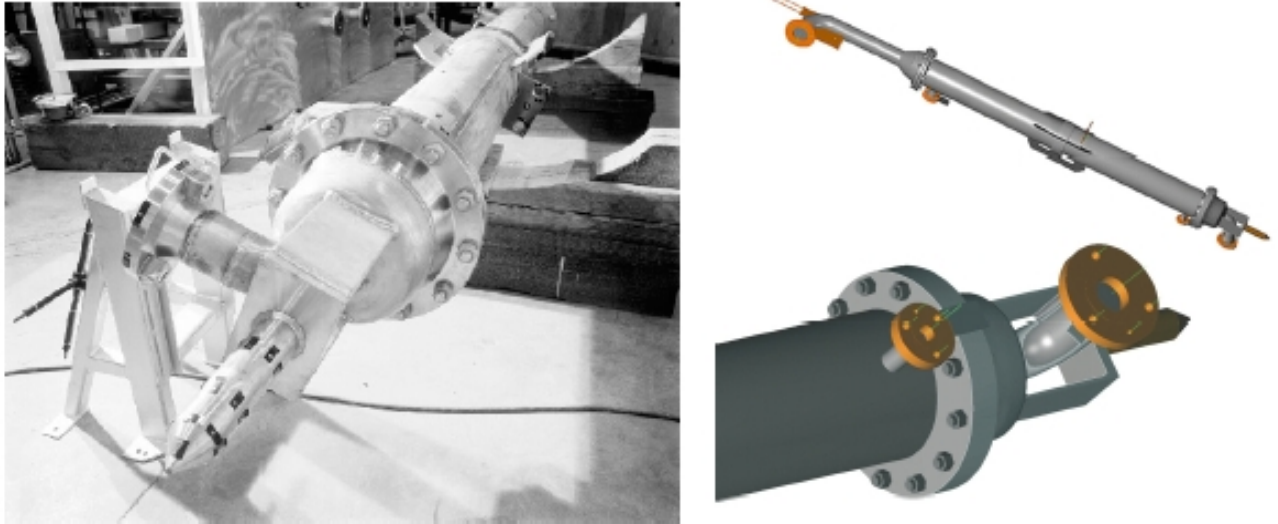


Figure 4.7: Example of a photogrammetric reconstruction [32].

4.2 Structured-Light Range Scanning for Moving Scenes [34]

In the following a radical new design of a real-time scanning system developed by Rusinkiewicz et al. at the Stanford University will be discussed.

Rusinkiewicz took part in the “Digital Michelangelo” project [33], where a high accuracy custom built scanner was used for scanning Michelangelo’s David in the Academic Gallery of Florence.

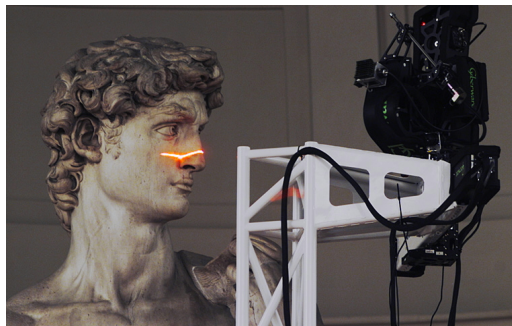


Figure 4.8: Digital Michelangelo Project [33].

The scanning of this approximately 7m high statue not only demonstrated the use of 3D scanning in humanities, it also showed some drawbacks of current systems:

The equipment was very expensive and two scanners were needed: one motor driven laser-stripe scan-head for the automatic-scan of the main parts, and a smaller one operated by hand, which was used to accomplish the scanning in hardly accessible areas, e.g. for hole-filling.

These scanners were difficult to use and the scanning process took a staff of 30 people several months. Also thousands of hours were spent afterwards on the post processing of the data

(scan-data merging and hole filling).

4.2.1 Objectives of the Scan-System-Design

With this experience in mind, the aim was to build a fast, inexpensive and easy to use 3D model acquisition system.

Rusinkiewicz et al. distinguish between object scanning (a single 3D image is taken from a rigid or non rigid object) and 3D model acquisition (build complete models of rigid objects). Instead of only focussing on the hardware, their approach was to optimize the entire scanning pipeline.

Starting with the hardware they looked for a scanning process that allows for getting as many data as possible from a single “shot”; they decided to use structured light as the scanning method.

Since they optimized every part of the scanning pipeline for speed (with a few trade-offs to accuracy) it was possible to set up a real-time scanner which gave them the opportunity to implement innovative features, which in turn helped to meet the goals of affordable cost and ease of use.

To keep the cost of their system low, they used off-the-shelf components. Additionally they estimated that a major part of cost is the calibrated motion stage, which is needed to get data from different points of view. In their design there is no need for a motion stage; the model is moved by hand relative to the scanner.

As mentioned earlier post processing and hole filling of the model usually demands skilled personal and plenty of time.

In the past the problem of hole-filling has been addressed by more or less satisfying algorithms that calculate the next best position of the scanner relative to the object for eliminating obscured surfaces and scanning the remaining parts of the object. But all these algorithms have their drawbacks. Inspired by the software-bundle that comes with the 3D Modelmaker (a hand-guided, joint arm scanner by Cyberware [67]) and that gives an immediate feedback of the scanned scene, their real-time scanning-pipeline offers a chance to the operator to see the already scanned surfaces and the areas of missing data (holes) and thus allows an interactive planning of the next positions.

4.2.2 Concept of the Scanning System

The model acquisition pipeline is illustrated in Figure 4.9. A pattern varying in time, consisting of black and white stripes, is projected onto the scene using a commercial DLP (Digital Light Processing) image projector. A standard NTSC video camera is used to capture video frames. The frames are processed to find and track the boundaries between the stripes, and a range image is obtained for each video frame. Successive range images are aligned to each other, integrated into a 3D grid, and rendered in real time.

A few aspects of this concept have to be discussed in more detail.

4.2.3 Alignment of Multiple Scans

One important stage of 3D model acquisition (not restricted to structured light scanners) is the alignment of the range images that were obtained from different views. Classical

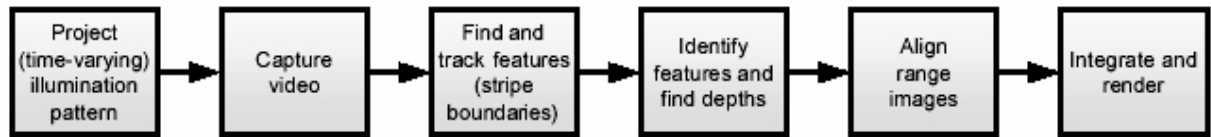


Figure 4.9: Real-time model acquisition pipeline [34].

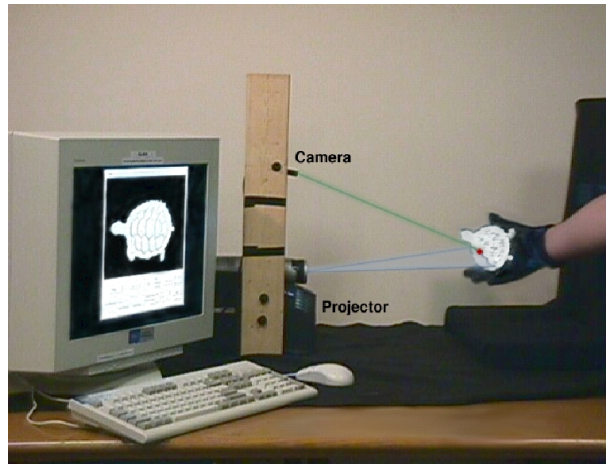


Figure 4.10: Photograph of the prototype system [34].

systems take benefit from their motion-stages, where “known motion” allows to determine the relative positions of the different range images.

A second way of obtaining the alignment between range images is to place a tracker on either the object or the scanner. Although Rusinkiewicz et al. have chosen not to use this option because of accuracy and cost considerations, they believe that in many circumstances it would provide substantial benefits in the context of his proposed pipeline.

The option they use for alignment in their pipeline is based on relating individual scans to each other based on the geometry in overlapping areas.

This automatic alignment of 3D shapes has been studied extensively in the computational geometry- and computer vision communities, mainly in the context of aligning two scans with completely unknown initial orientation. The Iterative Closest Points algorithm (ICP) introduced by Chen [35] has become the most frequently used method since it converges to the correct scan-to-scan alignment with high accuracy, if a good initial guess is given. Usually finding a good initial position is a major problem with ICP, but because of their fast scanning system they can assume that the positions from successive scans are close to each other. Since the introduction of the ICP algorithm in 1991 many variants have been developed which can be classified by the selection of points they use for comparison, by the point matching algorithm, how they weigh and reject pairs and by their error metric. Rusinkiewicz made exhaustive studies which methods were most suitable for his demands and he finally adapted Pulli’s algorithm [36] for real time use.

Also a lot of time was spent on developing a suitable illumination pattern.

4.2.4 Illumination Pattern

In literature different methods are described; each of these depend on certain assumptions, how the scene transforms the projected light into the camera image.

Some of these methods are color coded which inhibits assumptions on the reflectivity of the scene. Another class of scanners encodes information into a pattern in a certain neighborhood of projector pixels, which involves (local or global) spatial coherence assumptions. A third way of encoding information into the projector signal is to group pixels versus time rather than space. For example the projector pixels are turned on and off over time; thus when a camera pixel records a particular on- and off- intensity pattern, the corresponding projector pixel can be identified (temporal coherence is assumed).

The design space for projected light illumination patterns can therefore be described in terms of the reflectance, of the spatial coherence, and the temporal coherence assumptions they make. The strength of a spatial coherence assumption can be measured by the number of pixels involved: if the patterns that have to be identified in a given camera image require a minimum of n pixels, then the smallest identifiable features in the scene must occupy at least n camera pixels. The reflectance assumption impacts the range of colors permitted in the scene, as well as the spatial frequency of textures.

For real-time scanning Rusinkiewicz introduced an illumination pattern based on local (spatial and temporal) coherence assumptions.

- Most of the time, two horizontally adjacent camera pixels will see the same surface.
- Most of the time, the reflectance of two horizontally adjacent pixels is similar.
- Most of the time, projected features that persist for n frames can be used in making correlations. In the presented system $n = 4$ was chosen.

From these assumptions he deduced a time varying stripe-boundary code.

Stripe-boundary means that the triangulation plane is defined by the boundary between two stripes instead of the stripes themselves. This matches the goal of finding corresponding points (projector- and camera pixels) by 2 adjacent pixels and leads to high accuracy because the boundary can be determined with subpixel accuracy. In order to allow the greatest possible variation in scene reflectance, the system uses only black and white stripes.

Since the stripes on either side of a stripe boundary can each be assigned $m = 2^n$ (16) different codes over time, then roughly $m \cdot (m - 1) = 240$ directed stripe boundaries can be identified in a camera image.

To keep the stripe borders easy to track a few limitations have to be made:

Borders that do not change over time (all black 0000 next to all white 1111) cannot be distinguished from texture boundaries and thus have to be avoided.

A further limitation arises with “ghost-boundaries” (boundaries between stripes of the same color). Illumination patterns without any ghost borders would lead to codes consisting of too many frames since there are only 2^n (16) permanent visible borders.

Hence they allowed for ghost borders, but to keep it easy to identify them there is at the most one “ghost” between two visible borders, and a given stripe boundary will be visible at least at every other frame.

A graph with all allowed borders and the maximal-length code that matches the above constraints (found by brute force computation) are shown in Figure 4.11.

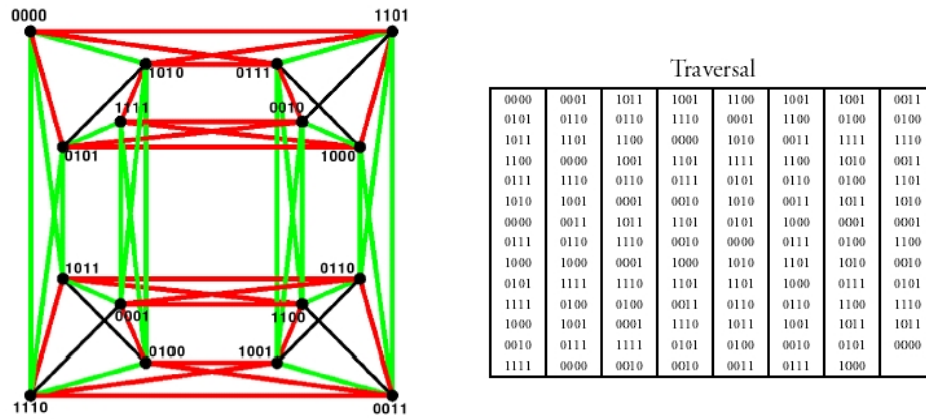


Figure 4.11: We look for a traversal of the 55 edges in this graph (with the understanding that each edge may be traversed once in each direction), with the added constraint that red and green edges must alternate along the path. A black edge may be substituted for either red or green. As mentioned in the text, the edge between 0000 and 1111 is missing: we disallow this stripe boundary as a valid code, since it is too easy to confuse with static texture. At right, a maximal-length ($2 * 55 = 110$ directed edges, 111 nodes) traversal [34].

4.2.5 Layout and Results

The layout of the system determines its working volume and resolution. For the scan presented below, the camera and projector are positioned 20cm apart from each other, with a triangulation angle of 21 degrees. This configuration produces a working volume approximately 10cm across. Near the front of the working volume, samples are spaced roughly every 0.5mm in Y (parallel to the stripe direction) and every 0.75mm in X (perpendicular to the stripes).

On the whole this real time scanning system performs well for scenes without high-frequency textures. A scan rate of 60Hz is achieved. The object may be moved, rather slowly, with a velocity up to 1cm/s. An example of a scanned object (after postprocessing) is given in Figure 4.12.

It has to be mentioned that the quality of this post-processed surface differs strongly from the output obtained by the real-time scanning pipeline. Because after the data acquisition during an offline post-process more time consuming algorithms can be used for the merging alignment and surface reconstruction, having the real-time surface as an initial guess.

4.2.6 Improved Illumination Pattern [37]

Despite of the subtle design of the illumination code there is still room for improvement:

- The “ghosts” sometimes lead to ambiguity in decoding.
- The code is not capable of capturing surface textures.

In [37] a new approach, the (b, s)-BCSL (Boundary Coded Structured Light) illumination pattern, is presented. It is the augmented basis of Rusinkiewicz’s code eliminating ghost boundaries and it also allows to capture texture. The entire code uses b colors and consists

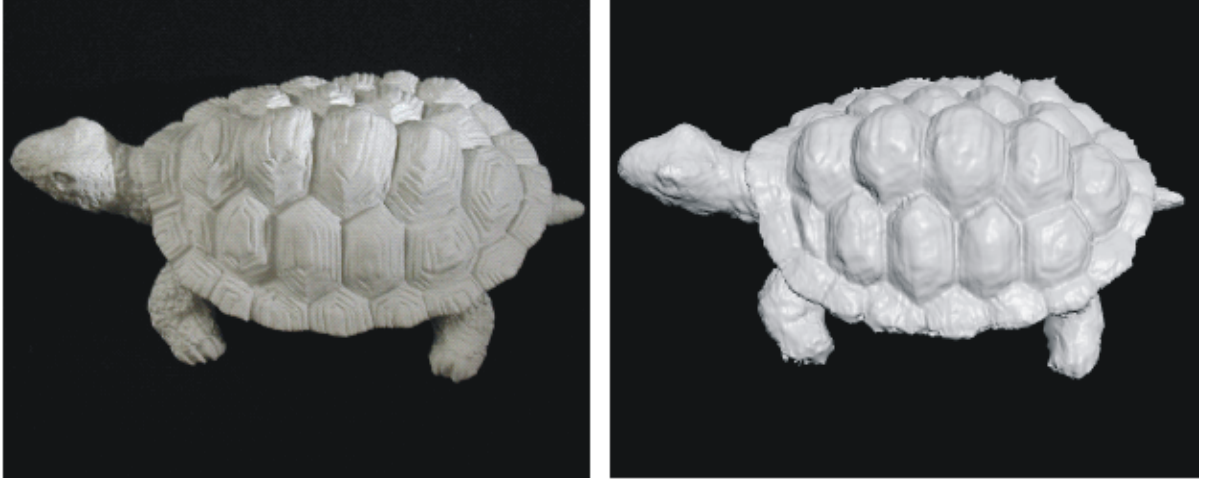


Figure 4.12: Photograph of the original (left) and reconstructed object (right) The entire scan was accomplished in ~ 4 min. From the recorded 1830 range images only a tenth of them was used for reconstruction (~ 600000 sample points) [34].

of s slides, thus $[b(b-1)]^s$ distinct borders can be coded. Despite the advantage of coding more information into a single frame by the use of color based codes, grey-codes are preferred since they are less sensitive to scene texture. By an innovative step this limitation can be eliminated: In [37] it is proposed that at every second frame the complementary pattern (of the previous one) is projected onto the object's surface (leading to a total of $2 \cdot s$ slides). Thus it is possible to recover the ambient light component (u_R, u_G, u_B) , the local intensity transfer factor (r_R, r_G, r_B) of the surface and the projector intensity for each channel (p_R, p_G, p_B) . If the camera characteristic is linear, the sensor clips intensity at a maximum value. The digitized intensity per channel (RGB) is given by:

$$I_i = \min(u_i + r_i \cdot p_i, I_{imax}) \quad (4.6)$$

Supposing that I_{Rmax} , I_{Gmax} and I_{Bmax} are never achieved the intensities are given by:

$$(I_R, I_G, I_B) = (u_R + r_R \cdot p_R, u_G + r_G \cdot p_G, u_B + r_B \cdot p_B) \quad (4.7)$$

It is possible to estimate the parameters u_i and r_i if the projector, sensor and object are fixed in relative positions, and produce sequential projected patterns varying p_i . As mentioned previously, the idea was to project two complementary slides, that is, if $p_i = 0$ on first slide then $p_i = 1$ on second. $I_i = u_i$ when $p_i = 0$, and $u_i + r_i$ when $p_i = 1$.

If just the maximum value per pixel for each channel between the complementary slides is taken, it will be equivalent to recovering the value of each pixel as if it were illuminated with white light coming from the projector, that is, $p = (1, 1, 1)$. Equivalently, if the minimum value per pixel for each channel is taken, the ambient light, that is, $p = (0, 0, 0)$ is recovered. One problem in the use of color coding is the cross-talk between the RGB sensors. In that respect, color fidelity can be improved by a color correction pre-processing step that takes into account the response of the projector-camera system. The traditional use of color in coding restricts the object surface reflectivity, because one would not want to modify the projected color when acquiring the surface. By projecting complementary

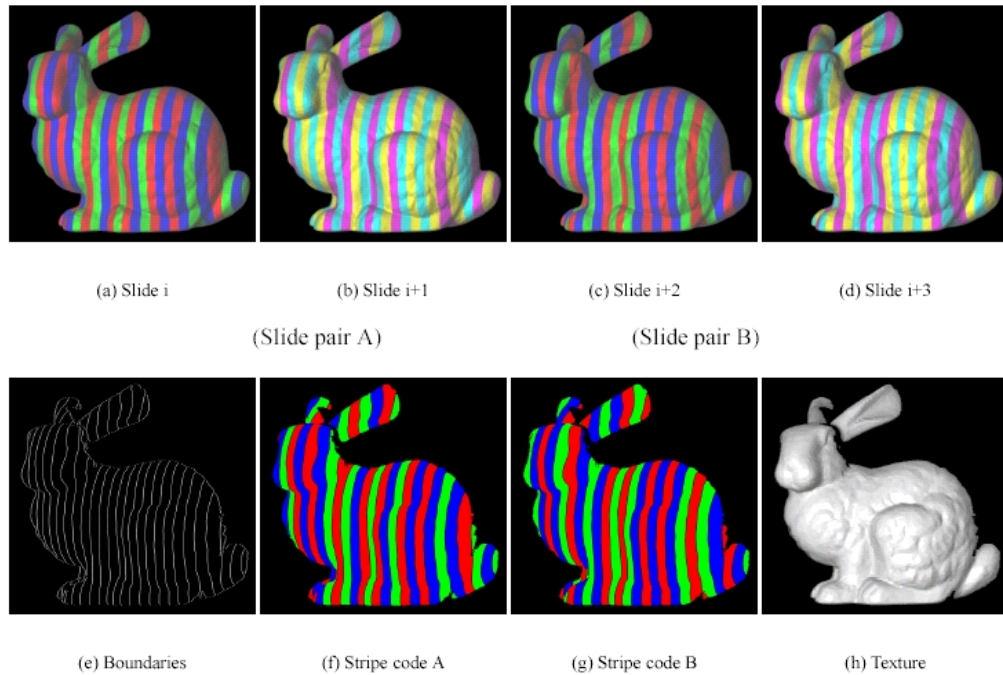


Figure 4.13: (3,2)BCSL Code projected on a bunny [37].

slides, the reflectivity restrictions are eliminated. From the signal transmission point of view we are introducing redundancy on the transmitted message replicating it on complementary slide. This is a good procedure as it reduces the probability of errors and the code-word read can be checked to assure the correctness of the received message. Note that the use of complementary colors allows to recover both the stripe and scene colors using only two slides. The considerations above are valid for all pixels, except at stripe boundaries. At those locations, one can recover color by interpolating the information from neighbor pixels at both sides of the boundary.

An example of the (3,2) BCSL code is given in Figure 4.13.

Concluding, the (b,s) BCSL Code seems to be one of the most promising illumination patterns. However, it has to be adapted for real-time use and currently there is no working prototype-system.

The given figures are rendered with the BMRT (Blue Moon Rendering Tools) [71] package using data sets from the Stanford 3D Scanning Repository [70].

4.3 Non-Scanning 3D TOF-Camera [13]

In his doctoral thesis [13] Lange presents an all-solid-state 3D time-of-flight range camera (based on CW-modulation) that does not require any mechanically moving parts.

Instead of scanning a laser beam and sequentially acquiring the range data point-wise, the entire scene is illuminated with a modulated light surface. In order to perform a 3D measurement, as illustrated in Figure 4.14 the received light has to be demodulated which leads to the desired phase-difference between transmitter and receiver. This however, necessitates the use of a 2D-electrooptical demodulator and detector to measure in parallel the distances of some hundreds or thousands of points of the observed scene.

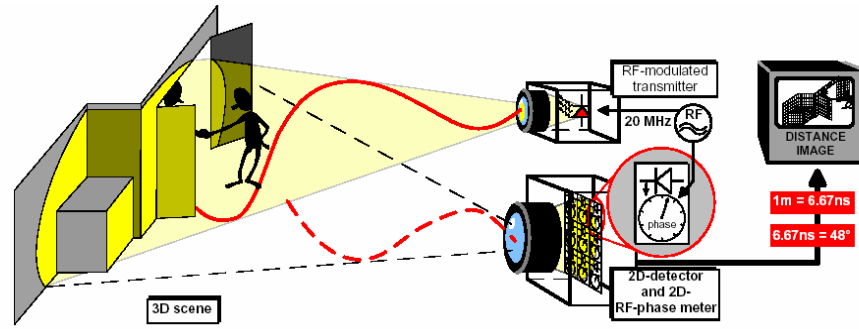


Figure 4.14: Principle of a non-scanning 3D TOF-camera [13].

The 2D detection itself can be performed with CCDs or (active) photodiode arrays, also called active pixel sensors (APS). However, in contrast to discrete photodiodes, these 2D detectors integrate the incoming light in every pixel and so the temporally coded information is lost. Thus one would need an electrical demodulator in every pixel, which demodulates the photocurrent of the pixel's photodiode. In contrast to the 1D-case, for an imaging 3D application such electrical 2D-demodulation devices are not available. Therefore the main effort was to design a customized photoASIC (Application Specific Integrated circuit) to realize this array of “demodulation pixels”. So we will also focus on the detector array and first demonstrate two demodulation techniques.

4.3.1 Demodulation and Sampling

By demodulating we mean extracting the information that is contained in a signal beside its carrier. In this case the received light is modulated in intensity and phase, where the phase modulation is caused by the 3D-information of the scene. Thus in order to measure a distance, the phase delay of this modulated light signal must be measured in the receiver.

Demodulation by cross correlation (homodyne mixing)

The signal's amplitude and phase can be extracted by synchronously demodulating the incoming modulated light within the detector. Demodulation of a received modulated signal can be performed by correlation with the original modulation signal. This process is known as cross correlation. The measurement of the cross correlation function at selectively chosen temporal positions (phases) allows the phase of the investigated periodical signal to be determined. The correlation function $\varphi_{sg}(\tau)$ is defined as follows:

$$\varphi_{sg}(\tau) = s(t) \otimes g(t) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} s(t) \cdot g(t + \tau) dt \quad (4.8)$$

With the received optical input signal $s(t)$, the modulation amplitude a and phase φ and the demodulation- or correlation signal $g(t)$ defined as

$$s(t) = a(1 + \cos(\omega t - \varphi)) \quad \text{and} \quad g(t) = \cos(\omega t) \quad (4.9)$$

the correlation function $c(\tau)$ can be calculated:

$$c(\tau) = \varphi_{sg}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} a(1 + \cos(\omega t - \varphi)) \cdot \cos(\omega t + \omega \tau) dt = \frac{a}{2} \cos(\omega \tau + \varphi) \quad (4.10)$$

By evaluating his function for different phases $\omega \tau$ the phase φ and the amplitude a of the received signal $s(t)$ can be recalculated.

Demodulation by DFT-Analysis (temporal sampling)

Another, slightly different approach is to sample the modulated signal synchronously to the emitted signal. Sampling always means to convolute the input signal with a sampling function, in this case a $rect(t/\Delta t)$ function.

$$rect(t/\Delta t) \text{ is } \begin{cases} 1 & \text{for } |t| \leq \Delta t/2 \\ 0 & \text{for } |t| > \Delta t/2 \end{cases} \quad (4.11)$$

Assuming that we sample a periodic signal we can use the equations of the DFT (Discrete Fourier Transform) in order to calculate both amplitude and phase of the base frequency and the harmonics contained in the signal.

Having a dataset with N pairs of values $x_k, f_k = f(x_k)$ where the x_k are equally spaced $x_{k+1} = x_k + \Delta x \dots (k = 0, 1, \dots, N-1)$, we can calculate its discrete Fourier-coefficients \tilde{f}_n :

$$\tilde{f}_n = \frac{1}{N} \sum_{k=0}^{N-1} e^{-\frac{2\pi i n k}{N}} f_k \quad (4.12)$$

These Fourier-coefficients contain the same information as expressed by the inverse DFT:

$$f_k = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} e^{\frac{2\pi i n k}{N}} \tilde{f}_n \quad (4.13)$$

Applying Equation 4.12 to N sampled intensity values I_k , the n^{th} harmonic of the detected light-intensity \tilde{I}_n is given by:

$$\tilde{I}_n = \frac{1}{N} \left[\sum_{k=0}^{N-1} \cos\left(\frac{2\pi n k}{N}\right) \cdot I_k + i \sum_{k=0}^{N-1} \sin\left(-\frac{2\pi n k}{N}\right) \cdot I_k \right] \quad (4.14)$$

Therefore the magnitude of the n^{th} coefficient equals

$$a_n = \frac{1}{N} \sqrt{\left[\sum_{j=0}^{N-1} \cos\left(\frac{2\pi n k}{N}\right) \cdot I_k \right]^2 + \left[\sum_{j=0}^{N-1} \sin\left(-\frac{2\pi n k}{N}\right) \cdot I_k \right]^2} \quad (4.15)$$

Having determined these coefficients we can interpolate the data points using Equation 4.13:

$$f(t) = a_0 + \left[2 \sum_{n=1}^{\frac{N}{2}-1} a_n \cos\left(2\pi n \frac{t}{T} - \varphi_n\right) \right] + a_{\frac{N}{2}} \cos\left(2\pi \frac{N}{2} \frac{t}{T}\right) \quad (4.16)$$

with

$$\varphi_n = \arctan \left[\frac{\sum_{k=0}^{N-1} I_k \sin\left(\frac{2\pi nk}{N}\right)}{\sum_{k=0}^{N-1} I_k \cos\left(\frac{2\pi nk}{N}\right)} \right] \quad (4.17)$$

These general equations can be simplified for the given application. Since the emitted signal is sinusoidally modulated, and the sampling is synchronous to the emitted signal, the detected light is given by the offset and the base frequency (no harmonics) components described by Equation 4.16.

The sampling of the optical input signal is illustrated in Figure 4.15. $A_0 \dots A_3$ represent the integrated (equally spaced) sampling points. The above equations for the phase φ , amplitude A and offset B of a sinusoidal signal obtained by $A_0 \dots A_3$ are rewritten to:

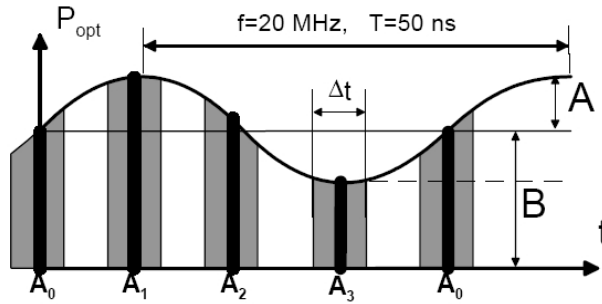


Figure 4.15: Optical sinusoidally modulated input signal, sampled with 4 sampling points per modulation period. The signal frequency of 20MHz defines the unambiguous distance range of 7.5m [13].

$$\varphi = \varphi_1 = \arctan \left[\frac{A_1 - A_3}{A_0 - A_2} \right] \quad (4.18)$$

$$B = a_0 = \frac{A_0 + A_1 + A_2 + A_3}{4\Delta t} \quad (4.19)$$

$$A = 2a_1 = \frac{\delta}{\Delta t \sin \delta} \frac{\sqrt{(A_3 - A_1)^2 + (A_0 - A_2)^2}}{2} \quad (4.20)$$

The factor $\frac{\delta}{\sin \delta}$ is to compensate for the attenuation of the measured amplitude which is caused by the integration over a finite sampling interval.

Here δ is defined as

$$\delta = \frac{\pi \Delta t}{T} \quad (4.21)$$

For the demodulation in the 3D-TOF camera the length of the sampling interval Δt was chosen to be $T/2$. This operation mode achieves the best overall performance, since it offers the most efficient use of the available optical power and additionally, for shorter integration intervals, a higher system bandwidth would be required. Furthermore to increase the SNR

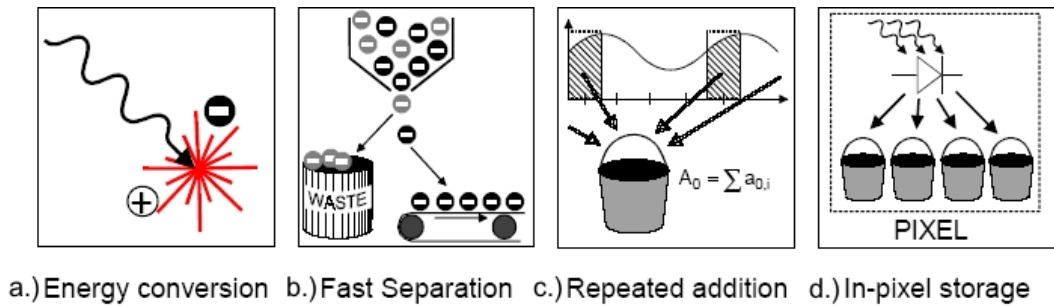


Figure 4.16: Physical demands on a demodulation pixel [13].

(signal to noise ratio) the sampling is performed accumulatively over several periods. The repetitive sampling makes the detection insensitive to other frequencies than the sampled one, therefore the detector pixels are sometimes referred to as "lock-in pixels".

4.3.2 Demodulation Pixels

The demodulation procedure implemented in these pixels is based on temporal sampling of the optical wave as described above. Thus the physical demands on the detector array can be summarized in four principal tasks (Figure 4.16):

(a) Conversion of incoming photons to electron-hole pairs, (b) fast separation of optically generated charge carriers (sampling-shutter mechanism), (c) repeated, noise free addition of these separated photoelectrons and (d) storage of the acquired sampling points. The CCD principle was chosen for the realization of all these demands, since it allows for a nearly loss-free addition and directed transport of electric signals in the charge domain.

To achieve an optimum of light sensitivity (optical fill-factor) and high dynamic range, 3 different chip designs were explored.

Multitap Lock-In CCD

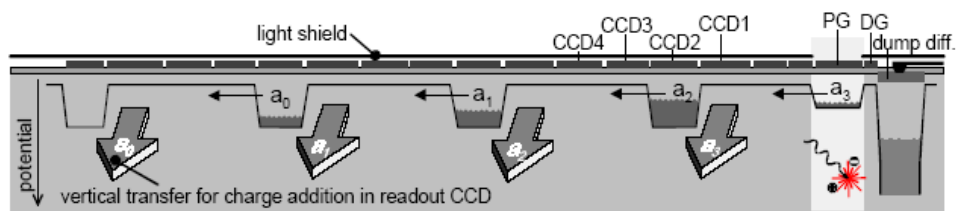


Figure 4.17: Multitap lock-in pixel: cross-sectional view [13].

In the Multitap-layout the photogate is serially attached to a 4-phase CCD line, the pipeline CCD. The generated photoelectrons are shifted to the left. By proper modulation of the gates, each CCD element carries one sample point of the received signal after four shifts. In order to perform a repetitive sampling, the acquired charge carriers are transferred to a second CCD line, the readout-CCD. This process is indicated by the grey arrows in Figure 4.17.

This design is flexible regarding the number of storage sites and thus would be suitable for more complex demodulation, but since the in-pixel storage acquires room the optical fill-factor is just $\sim 5\%$. In addition, this design is restricted to sampling frequencies below 1MHz, since the charge transfer from one latch to the other in the pipeline CCD is rather slow. However, this drawback can be partly overcome by another production process.

4-Tap Lock-In Pixel

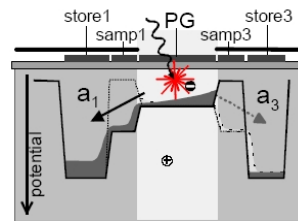


Figure 4.18: Four-tap lock-in pixel cross-sectional view (only two storage sites are shown) [13].

In this design (Figure 4.18) the photosensitive area is surrounded by 4 (charge) storage sites, offering better speed and fill-factor ($\sim 7\%$). Every storage cell requires only one transfer gate for the accumulative demodulation. However one problem is that due to production-inaccuracies the efficiency of the transfer gates are varying, resulting in an inhomogeneous response of the 4 sampling points and therefore demodulation errors occur.

Single Tap Lock-In Pixel

The limitations of the above designs are mainly caused by the implementation of the 4 independent in-pixel storage cells. In the single-tap configuration each pixel has just one storage site and thus the optical fill-factor is maximized, but as a trade-off the sampling points have to be acquired serially and stored externally.

The principle of the demodulation-process is given in Figure 4.19. By applying proper gate voltages to the photogates, the potential gradient in the semiconductor can be influenced. During the sample interval the charge carriers generated by photo-conversion under the photogates are accumulated in the "integration" cell, whereas in-between the sampling intervals by reversing the control voltages the charge carriers are shifted to the dump location (refer to Figure 4.19).

The main drawback in this design is the serial sampling, which makes the phase measurement sensitive to time-varying illumination or fast moving objects in the scene. Further half of the available optical power is wasted, since half of the time the generated photoelectrons are moved to the dump diffusion. But due to its outstanding speed ($>20\text{MHz}$) and optical fill-factor ($\sim 22\%$) this is the final design the system is based on.

4.3.3 System

The prototype has a demodulation-array of 64×25 pixels. The scene is illuminated by multiple LEDs driven with a square wave signal, but due to the LED's low pass characteristic the

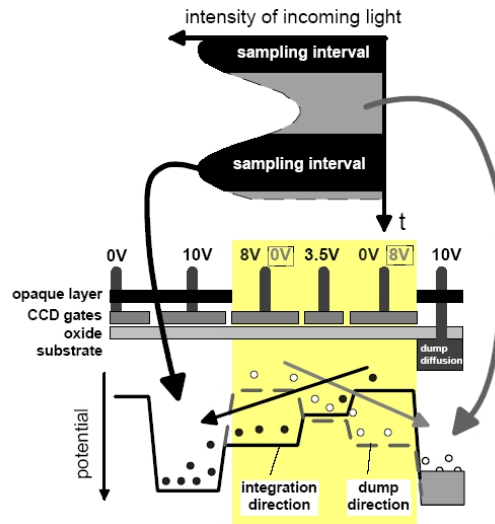


Figure 4.19: Single-tap lock-in pixel: illustration of the sampling-process [13].

emitted light is sinusoidal. With the modulation frequency of 20MHz an unambiguous distance of 7.5m is realized. At 10Hz frame-rate even for non-cooperative targets an accuracy of 10cm is achieved. An illustration of a range-pattern is shown in Figure 4.20.

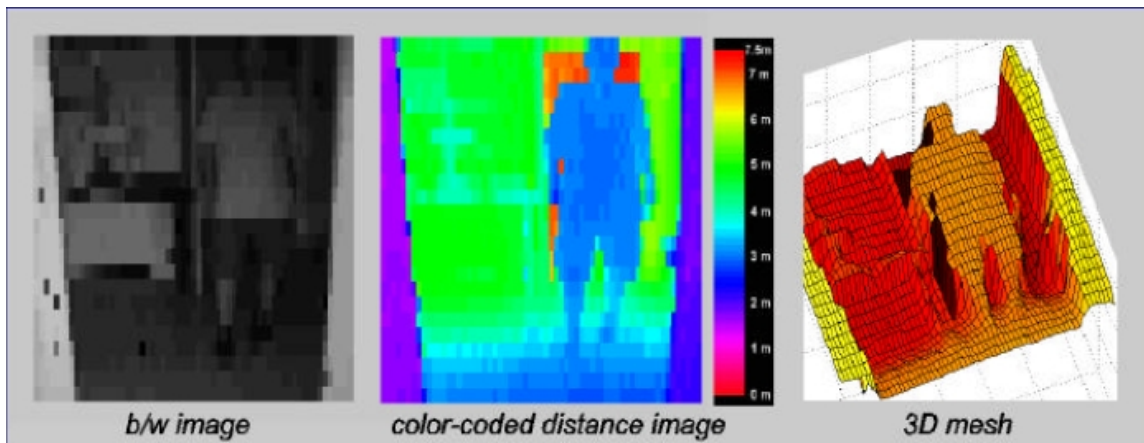


Figure 4.20: Scene (desk, door, student) scanned with 10Hz framerate [13].

The accuracy of the system is principally given by two factors: The wavelength of the emitted light, (which itself is restricted by the bandwidth of the demodulation pixels), and the resolution of the phase-measurement. This phase-resolution is limited by the quantum noise of the incident light and dark-noise. For higher resolutions the SNR has to be increased, which can be achieved by better optical fill-factors, higher illumination intensities and more effective demodulation techniques.

The system performance may be strongly increased in the future due to the ongoing miniaturization in microelectronics. Generally speaking, this is because devices become faster with decreasing minimum feature size and hence, higher demodulation frequencies (beyond 100 MHz) will become possible. Also by a special lens design (optoASIC) the fill-factor may increase up to 100 % and in combination with better demodulation concepts frame-rates of 50Hz and more are predictable.

Part II

**Methods of 3D-Surface
Reconstruction**

Chapter 5

General Statement of the Problem

In Part I of this thesis different systems to scan object surfaces have been described. The data obtained by those systems are a set of unorganized 3D coordinates of the sampled surface (which is often called a point-cloud). However, this is only one part of the 3D acquisition pipeline. For example, visualization or CAM applications, require that the surface has to be reconstructed, which means that we would have to know each single point of the surface. This is a classical reconstruction problem, not all information is available and the original shape can only be approximated. This leads us to the main problem: The geometric notion of "shape" has no associated formal meaning. This is in sharp contrast to other geometric notions, such as diameter, volume, *convex hull* (refer to the Appendix), etc. [1]. So the shape formed by a set of points is rather a vague notion and there are many possible interpretations. Since we can't have a general definition of "shape", each algorithm has to restrict itself to one interpretation. Thus beside a stringent formulation of the shape the algorithm has to define a set of rules, how to interpolate or approximate the surfaces from a given set of points. For example, these rules are concerned with the selection of points that are used for the approximation of a local tangent plane, with the interpolation of neighboring points and how to deal with boundaries.

Depending on additional information of the sampled dataset (surface normals, sample density, ...), on demands of the resulting surface (approximation or interpolation of the sample points, smoothness, watertightness), and on computing power, different algorithms have been and are currently in use. In the following a few of them will be covered.

Chapter 6

Hoppe's Algorithm

The algorithm presented in [39] is one of the first used for surface reconstruction from an unorganized point-set. In contrast to former algorithms, no structure of the data is assumed. The only input-information needed is a set of points $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \subset \mathbb{R}^3$ on or near an unknown *manifold* U (a non intersecting surface). As an output the algorithm produces a *simplicial surface*¹, that approximates U . The data points x_i are assumed to have a noise less or equal δ , and the sample-density² is denoted by ρ .

6.1 Basic Principle [39]

The algorithm is based on an implicit representation of the surface with the help of a distance-function \tilde{d}_U and consists of the following steps (refer to Figure 6.1):

- computation of a signed distance function $\tilde{d}_U : D \rightarrow \mathbb{R}$, where $D \subset \mathbb{R}^3$ is a region near the data and $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ is close to the zero set $Z(\tilde{d}_U)$. This distance function equals zero for points on the estimated surface \tilde{U} .
- a contouring algorithm to approximate the zeroset $Z(\tilde{d}_U)$ by a simplicial surface.

The contour-tracing is a well-studied problem so we will focus on the computation of the signed distance function.

6.2 The Signed Distance Function [39]

The signed distance function $\tilde{d}_U(\mathbf{p})$ from an arbitrary point $\mathbf{p} \in \mathbb{R}^3$ to a known surface U is the distance between \mathbf{p} and its closest point $\mathbf{z} \in U$, multiplied by ± 1 , depending on which side of the surface \mathbf{p} lies. To decide on which side the interior and the exterior of U is situated, we need an local approximation of the surface. This is achieved by computing oriented tangent planes for each point \mathbf{x}_i of the point set.

¹A simplicial surface is a piecewise linear surface with triangular faces.

² ρ does not reflect a "density" in a physical sense; here it is understood as the maximum distance of two neighboring sample points.

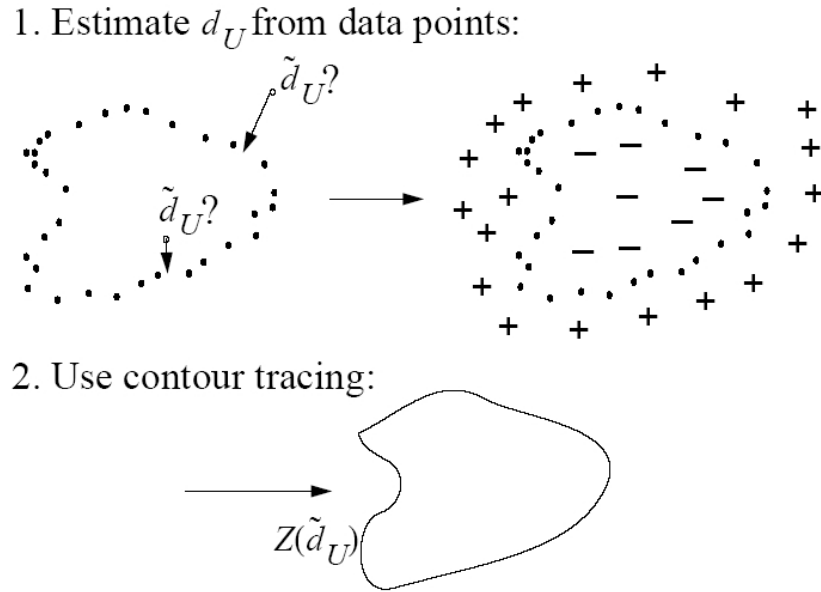


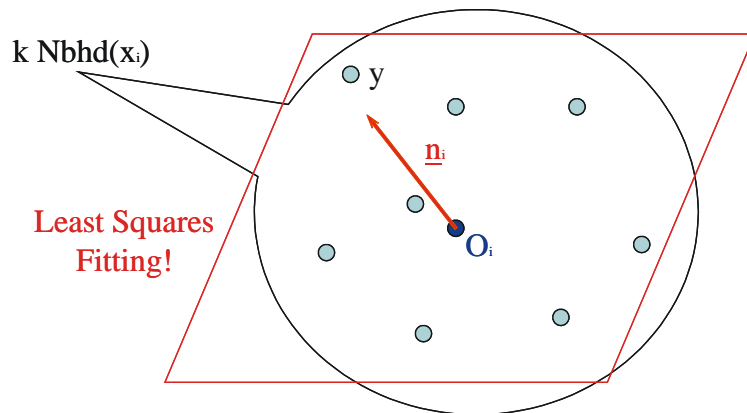
Figure 6.1: 2D-illustration of the two stages of the reconstruction algorithm [40].

6.2.1 Tangent Plane Estimation

The tangent plane $Tp(\mathbf{x}_i)$ associated with the data point \mathbf{x}_i is represented as a point \mathbf{o}_i , called the center, together with a unit normal vector $\hat{\mathbf{n}}_i$. The signed distance of an arbitrary point $\mathbf{p} \in \mathbb{R}^3$ to $Tp(\mathbf{x}_i)$ is defined to be

$$dist_i(\mathbf{p}) = (\mathbf{p} - \mathbf{o}_i) \cdot \hat{\mathbf{n}}_i. \quad (6.1)$$

The center and normal for $Tp(\mathbf{x}_i)$ are determined by collecting the k points of X nearest to \mathbf{x}_i ; this set is denoted by $Nbhd(\mathbf{x}_i)$ and is called the k -neighborhood of \mathbf{x}_i (In [39] k is a user-specified parameter, whereas in [40] the neighborhood of \mathbf{x}_i consists of all points that are within a ball with radius $\delta + \rho$ centered at \mathbf{x}_i .)

Figure 6.2: k -neighborhood and estimated tangent plane [41].

The center and unit normal are computed in a way that the plane $\{dist_i(\mathbf{p}) = 0\}$ is the least-squares best fitting plane to $Nbhd(\mathbf{x}_i)$. The center \mathbf{o}_i is taken to be the centroid of

$Nbhd(\mathbf{x}_i)$. To compute the normal $\hat{\mathbf{n}}_i$, the covariance matrix \mathbf{CV} of $Nbhd(\mathbf{x}_i)$ is formed. This is the symmetric 3×3 positive semi-definite matrix

$$\mathbf{CV} = \sum_{\mathbf{y} \in Nbhd(\mathbf{x}_i)} (\mathbf{y} - \mathbf{o}_i) \otimes (\mathbf{y} - \mathbf{o}_i) \quad (6.2)$$

where \otimes denotes the outer product vector operator³. If $\lambda_i^1 \leq \lambda_i^2 \leq \lambda_i^3$ denote the eigenvalues of \mathbf{CV} associated with the unit eigenvectors v_i^1 , v_i^2 , v_i^3 , respectively, $\hat{\mathbf{n}}_i$ is chosen to be either v_i^3 or $-v_i^3$. This selection determines the orientation of the tangent plane, and it must be done so that nearby planes are consistently oriented.

6.2.2 Consistent Tangent Plane Orientation

Let us assume two points $\mathbf{x}_i, \mathbf{x}_j \in X$ are close to each other. Ideally, when the dataset is dense and the surface is smooth, the corresponding tangent planes $Tp(\mathbf{x}_i) = (\mathbf{o}_i, \hat{\mathbf{n}}_i)$, $Tp(\mathbf{x}_j) = (\mathbf{o}_j, \hat{\mathbf{n}}_j)$ are nearly parallel, i.e. $\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j \approx \pm 1$. If the planes are consistently orientated, then $\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j \approx +1$; otherwise, either $\hat{\mathbf{n}}_i$ or $\hat{\mathbf{n}}_j$ should be flipped. We can model this problem as a graph optimization.

Let us consider the following graph G . This undirected graph $G = (V; E)$ contains a vertex $i \in V$ for each tangent plane $Tp(\mathbf{x}_i)$ and edges $(i, j) \in E$ connecting two vertices in this graph, if the centers \mathbf{o}_i and \mathbf{o}_j of the corresponding tangent planes are sufficiently close ($\|\mathbf{x}_i - \mathbf{x}_j\| < \rho + \delta$). A straight forward implementation would be to assign to each edge a cost of $\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j$, and find a combination of $+\hat{\mathbf{n}}_i$ or $-\hat{\mathbf{n}}_i$, that the total cost of the graph is maximized. However, in [40] Hoppe proved that there exists no algorithm to solve this problem in a reasonable amount of time. So he finally opted for a greedy algorithm, where edges that connect nearly parallel planes are favoured. This was achieved by assigning a cost of $1 - |\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j|$ to each edge (i, j) , computing the *minimal spanning tree*⁴ (MST) of this graph, and then propagate through this graph, having the planes properly orientated, meaning that $\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j > 0$.

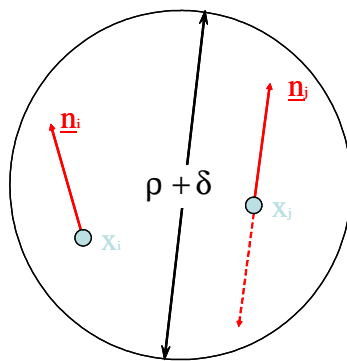


Figure 6.3: Consistent tangent plane orientation [41].

³If \mathbf{a} and \mathbf{b} have components a_i b_j respectively, then the matrix $\mathbf{a}_i \otimes \mathbf{b}_j$ has $\mathbf{a}_i \mathbf{b}_j$ as its ij -th entry.

⁴The Minimal Spanning Tree (MST) is the collection of edges that join all points in a set together, with the minimum possible sum of edge values.

6.2.3 Implementation

Having computed the orientated tangent planes, the distance function for a selected point \mathbf{p} simply equals:

$$\tilde{d}_U(\mathbf{p}) = \text{dist}_i(\mathbf{p}) = (\mathbf{p} - \mathbf{o}_i) \cdot \hat{\mathbf{n}}_i \quad (6.3)$$

Where \mathbf{o}_i is the center of the tangent plane that is next to \mathbf{p} . This works well for surfaces without boundaries (closed surfaces), whereas surfaces, that might have boundaries require a slightly more sophisticated computation.

Concerning datasets with a sample density ρ and a noise δ , a point \mathbf{z} on the tangent plane can only belong to the surface U if the smallest distance between \mathbf{z} and any point $\mathbf{x}_i \in X$ is less than $\rho + \delta$. Thus the distance function has to be extended by checking if the projection \mathbf{z} of a point \mathbf{p} onto the tangent plane $Tp(\mathbf{x}_i) = (\mathbf{o}_i, \hat{\mathbf{n}}_i)$ is on the surface U .

Procedure 6.1 Signed Distance Function (\mathbf{p})

- 1: $i \leftarrow$ index of tangent plane whose center is closest to \mathbf{p}
 {Compute \mathbf{z} as the projection of \mathbf{p} onto $Tp(\mathbf{x}_i)$ }
 - 2: $\mathbf{z} := \mathbf{p} - ((\mathbf{p} - \mathbf{o}_i) \cdot \hat{\mathbf{n}}_i) \cdot \hat{\mathbf{n}}_i$
 - 3: **if** $d(\mathbf{z}, X) < (\rho + \delta)$
 { $d(X, Y)$ denotes the Hausdorff distance, which is simply the distance between the closest points of X and Y .}
 - 4: $\tilde{d}_U(\mathbf{p}) := (\mathbf{p} - \mathbf{o}_i) \cdot \hat{\mathbf{n}}_i$ { $= \pm \|\mathbf{p} - \mathbf{z}\|$ }
 - 5: **else**
 - 6: $\tilde{d}_U(\mathbf{p}) = \text{undefined}$
 {The undefined values are used by the contouring algorithm to identify boundaries.}
 - 7: **return** $\tilde{d}_U(\mathbf{p})$
-

6.3 Contour Tracing

Contour tracing, the extraction of an *isosurface*⁵ from a scalar function, is a well-studied problem. For the implementation a variation of the *Marching Cubes* algorithm [38] was chosen that samples the function at the vertices of a cubical lattice and finds the contour intersections within tetrahedral decompositions of the cubical cells. To accurately estimate boundaries, the cube size should be set so that edges are of length less than $\rho + \delta$. In practice it is often convenient to set the cube size somewhat larger than this value, simply to increase the speed of execution and to reduce the number of triangular facets generated.

6.4 Algorithm Complexity and Conclusion [41]

Hoppe's simple algorithm can reconstruct surfaces of arbitrary topology and deals with boundaries in a natural way, as a drawback the resulting mesh is very dense and cannot

⁵Isosurfaces are shapes described by mathematical functions.

preserve sharp features. These drawbacks are solved in an extended version of this algorithm as described in [40]. However the algorithm has to deal with a complexity of $O(n^2)$ subdivided as follows:

- k-nearest neighbors to a given point $O(n+k*\log(n))$
- Nearest tangent plane origin to a given point $O(n)$
- MST graph $O(n^2)$

Chapter 7

Three Dimensional Alpha-Shapes

Based upon earlier work dealing with two dimensional *alpha-shapes* [3], the extension to three dimensions is presented in [1]. The algorithm described computes the shape formed by a set of points in \mathbb{R}^3 , the α -shape, with a performance of $O(n^2)$ in the worst case.

The α -shape is a well-defined polytope¹, derived from the *Delaunay Triangulation* (refer to the Appendix) of the point set, with a parameter $\alpha \in \mathbb{R}$ controlling the desired level of detail as illustrated in Figure 7.1.

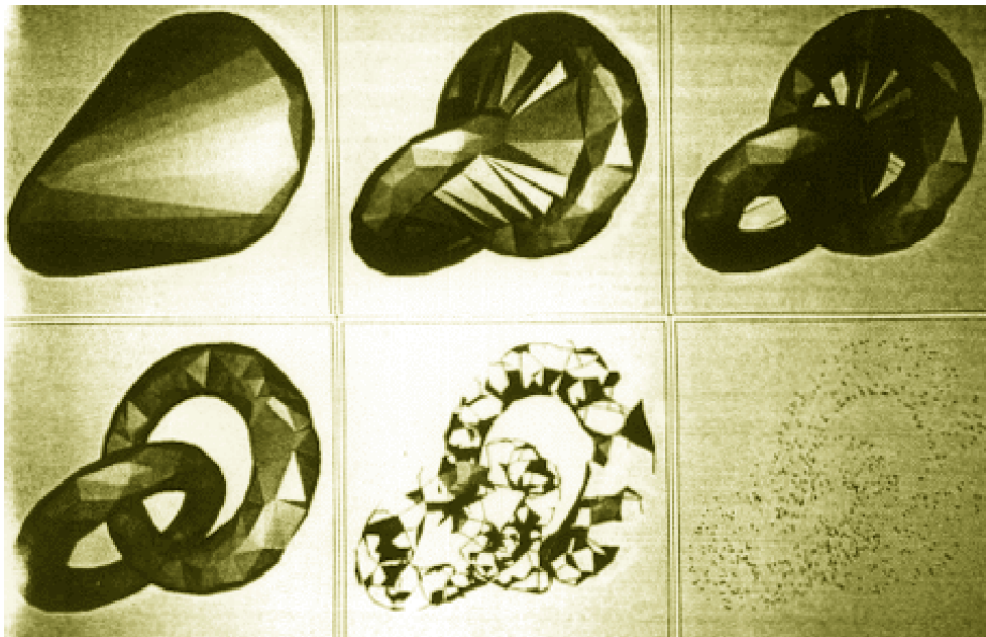


Figure 7.1: Two tori (800 points, 12197 triangles). The points are randomly generated on the surface of two linked tori. Six different α -shapes for values of α decreasing from top to bottom and left to right are shown. The first shape is the convex hull, for $\alpha = +\infty$; the last shape is the point set itself, for $\alpha = 0$. The α -value used in the fourth frame neatly separates the two tori. A further decreasing α disassembles the shape [1].

¹A polytope is the generalization to any dimension of a polygon in two dimensions, and a polyhedron in three dimensions.

7.1 Intuitive Description [2]

Imagine a huge mass of ice-cream making up the space of \mathbb{R}^d ($d = 2, 3$) and containing the points S as “hard” chocolate pieces. Using one of these sphere-formed ice-cream spoons we carve out all parts of the ice-cream block we can reach without bumping into chocolate pieces, thereby even carving out holes in the inside (e.g. parts not reachable by simply moving the spoon from the outside). We will eventually end up in an (not necessarily convex) object bounded by caps, arcs, and points. If we now straighten all “round” faces to triangles and line segments, we have an intuitive description of what is called the α -shape of S . And what is α in the game? α is the radius of the carving spoon. A very small value will allow us to eat up all of the ice-cream except for the chocolate points S . On the other hand a large value of α will prevent us even from moving the spoon between two points since it is far too large. So we will never spoon up ice cream lying in the inside of the convex hull of S , and hence the α -shape for $\alpha \rightarrow \infty$ is the convex hull of S (refer to Figure 7.2 for explanation).

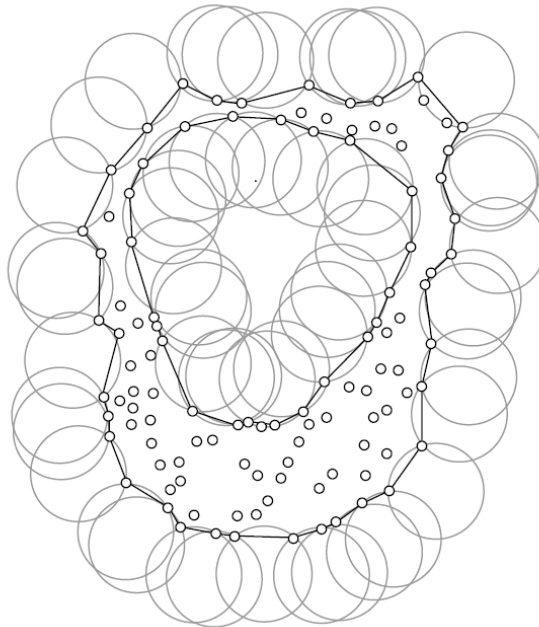


Figure 7.2: Demonstrating the process of constructing the α -shape in a 2D analogue (where the ice-cream spoon is simply a circle) [2].

7.2 Terms and Definitions

The description above will help us to understand the terms of a more precise definition given in the following.

General Position (GP)

The following assumptions ensure that the Delaunay Triangulation of the point set S is unique.

- GP1: no 4 points of S lie on a common plane and no 5 points lie on a common sphere.
- GP2: for any fixed α the smallest sphere through any 2,3 or 4 points of S has a radius different from α .

If the point set does not meet the general position criteria, it is still possible to apply the α -shape algorithm by using a technique called *SoS* (Simulation of Simplicity) [4, 1] which “stimulates” an infinitesimal perturbation of the points, so that they are in general position afterwards [2].

λ -ball:

Definition: For $0 < \lambda < \infty$ let a λ -ball be an open ball with radius λ . Furthermore a 0-ball is a point and a ∞ -ball is an open half-space. Now a certain λ -ball b (at a given location) is called empty if $b \cap S = \emptyset$.

α -hull:

Definition: The α -hull of S denoted as H_α is defined as the complement of all empty α -balls.

α -exposed simplex:

Definition: In \mathbb{R}^d any subset $T \subseteq S$ of size $|T| = k + 1$, with $0 \leq k \leq d$, defines a k -simplex σ_T that is the convex hull of T , also denoted by $conv(T)$. The general-position assumption assures that all k -simplices are properly k -dimensional. For $0 \leq k \leq d - 1$, a k -simplex σ_T is said to be α -exposed if there is an empty α -ball b with $T = \partial b \cap S$, where ∂b is the sphere or plane bounding b . A fixed α thus defines sets $F_{k,\alpha}$ of α -exposed k -simplices for $0 \leq k \leq 2$.

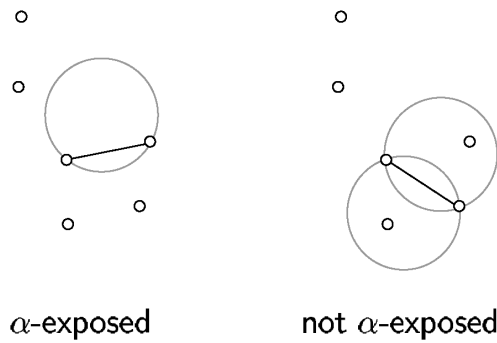


Figure 7.3: An example of an α -exposed simplex (a line segment) for the case $d = 2$ [2].

Formal definition of an α -shape:

Definition: The α -shape of S , denoted by S_α , is the polytope whose boundary consists of the α -exposed triangles in $F_{2,\alpha}$, the edges in $F_{1,\alpha}$ and the vertices in $F_{0,\alpha}$.

$$\partial S_\alpha = \{\sigma_T : T \subseteq S, |T| \leq d \text{ and } \sigma_T \text{ } \alpha\text{-exposed}\}$$

The k -simplices in $F_{k,\alpha}$ are also called the k -faces of S_α .

We still have to specify which connected components of $\mathbb{R}^3 - \partial S_\alpha$ are interior and which are exterior to S_α . Imagine that we fix the value of α and notice that for each α -exposed triangle σ_T there are two (not necessarily empty) α -balls, $b_1 \neq b_2$, so that $T \subseteq \partial b_1$ and $T \subseteq \partial b_2$. If both α -balls are empty then σ_T does not belong to the boundary of the interior of S_α . Otherwise, assume that b_1 is empty and that b_2 is not. In this case σ_T bounds the interior of S_α . More specifically, the interior of S_α and the center of b_1 lie on different sides of σ_T .

7.3 Observations on Alpha-Shapes

Since we now have a precise definition of an α -shape let us have a look at some of its properties:

The first observation is quite descriptive if you think in terms of the "ice cream scenario" at the beginning of this chapter:

Observation 1: $\lim_{\alpha \rightarrow 0} S_\alpha = S, \lim_{\alpha \rightarrow \infty} S_\alpha = \text{conv}(S)$

The next two observations are very valuable for its implementation:

Observation 2: If σ_T is an α -exposed simplex of S then $\sigma_T \in DT(S)$, where $DT(S)$ is the Delaunay Triangulation of S .

Proof: The statement definitely holds for d -simplices Δ_T because in this case the α -ball coincides with the circumsphere (or great circle, respectively) of T . So let σ_T be a k -simplex for $k < d$ and assume $\sigma_T \notin DT(S)$. Move the center of the empty α -ball continuously while adjusting the ball's radius so that the points of T always lie on its boundary. Since σ_T cannot lie on the boundary of the convex hull of S (for then it would be in $DT(S)$), the ball eventually moves to a position where it bumps on another point $\mathbf{q} \in S \setminus T$ ². Figure 7.4 shows a sketch for $d = 2$ and a 1-simplex. If $|T \cup \{\mathbf{q}\}| = d + 1$ we have found thus an empty

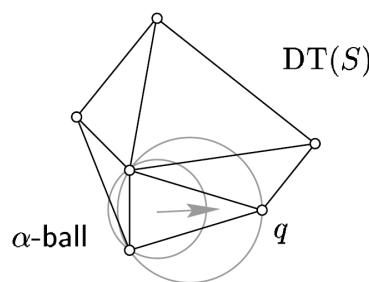


Figure 7.4: An α -exposed simplex of S is in the Delaunay-Triangulation of S [2].

circumsphere of $d + 1$ points which means that $\sigma_{T \cup \{\mathbf{q}\}}$ and its face σ_T lie in $DT(S)$. However if $|T \cup \{\mathbf{q}\}| < d + 1$ we repeat the process until the α -ball touches $d + 1$ points. \square

Observation 2 implies Observation 3, but due to its importance it is mentioned separately here:

² $S \setminus T$ means the elements of the set S diminished by the elements of the (sub)set T .

Observation 3: For any $0 \leq \alpha \leq \infty$ we have $\partial S_\alpha \subseteq DT(S)$.

Knowing that the border of the α -shape ∂S_α -that is what we are finally interested in- is a subset of the Delaunay Triangulation, it is sufficient to inspect only the Delaunay triangles, edges and vertices for the computation of ∂S_α . If there exists an empty ball with radius α touching them, they are part of ∂S_α .

Although this method is straightforward, it has several drawbacks. Having a fixed value of α , the test for α -exposed triangles is quite simple, since there are only two balls touching all vertices of a certain triangle, but considering edges or vertices we have to deal with infinite-simal many balls, that have to be checked for emptiness, therefore an elaborate computation is required.

Edelsbrunner and Mücke [1] invented a more efficient implementation where a simple test (the α -test) is used to accept a simplex as part of S_α and in addition the algorithm computes the α -shape for all values of α implicitly.

The idea behind their implementation is to compute first a simplicial complex -the α -complex C_α - which has the same boundary as the α -shape, and then to specify an interval for each simplex of $C_\alpha(S)$, for which values of α this simplex is part of the boundary ∂C_α .

α -complex:

Definition For a given point set $S \subseteq \mathbb{R}^3$ and $0 \leq \alpha \leq \infty$, the α -complex $C_\alpha(S)$ of S is a simplicial subcomplex of $DT(S)$. A k -simplex ($0 \leq k \leq d$) $\sigma_T \in DT(S)$ is in $C_\alpha(S)$ if the two conditions (C) hold:

- (C1): The radius of its circumsphere $\rho_T < \alpha$ and the circumsphere (centered at μ_T) is empty, or
- (C2): σ_T is a face of another simplex in $C_\alpha(S)$.

Notice that this is actually the set of all simplices in $DT(S)$ satisfying (C1), enlarged by as many faces fulfilling (C2) as needed to make the set a simplicial complex. The condition (C1) is called the α -test here. In contrast to ∂S_α , the α -complex can contain d -dimensional simplices; on the other hand, $C_\alpha(S)$ contains all points S by definition (which need not be the case for S_α).

In [2] proofs are given that this constructed polytope has indeed the α -shape as its boundary as illustrated in Figure 7.5 :

$$\partial C_\alpha(S) = \partial S_\alpha(S) \tag{7.1}$$

7.4 Implementation [2]

Since it is not really obvious how to compute the implicit representation of the α -shape for all values of α , a plain but straightforward approach consisting of 3 steps, where steps 2 and 3 have to be repeated for every value of α is presented here:

1. Compute the Delaunay Triangulation of S , knowing that the boundary of our α -shape is contained in it.

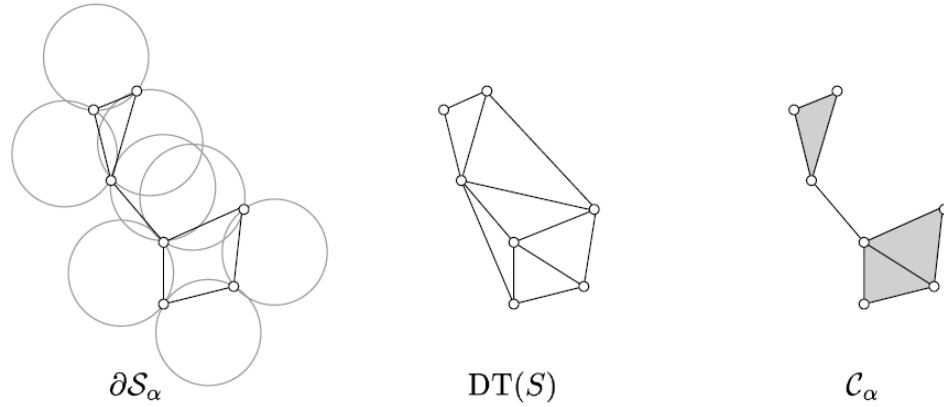


Figure 7.5: The α -shape, Delaunay Triangulation and the α -complex of a point set S for $d = 2$ [2].

2. Then determine $C_\alpha(S)$ by inspecting all simplices σ_T in $DT(S)$: If the circumsphere of σ_T is empty and the radius $\rho_T < \alpha$ (this is the α -test) we accept σ_T as a member of $C_\alpha(S)$, together with all its faces.
3. All d -simplices of $C_\alpha(S)$ make up the interior of S_α . So we have to extract the boundary of $C_\alpha(S)$ in order to obtain $\partial S_\alpha(S)$.

To make this algorithm work we need three things. First, the Delaunay Triangulation (which is not a problem, there are algorithms to do that). Then, a test to check whether or not the ρ_T -ball is empty. This too can be done, for instance, by checking whether \mathbf{p} lies in the said ball for every $\mathbf{p} \in S \setminus T$. Finally, we need a way to see whether a simplex σ_T in C_α lies on the boundary. For this, let us assume that the Delaunay Triangulation algorithm returns (in addition to the triangulation) for every simplex, whether or not it is on the boundary $\partial \text{conv}(S)$ of the convex hull. Then the boundary of the α -shape can be determined using Observation 4.

Observation 4: Let σ_T be a simplex in $C_\alpha(S)$. If $\sigma_T \in \partial \text{conv}(S)$, then it is obviously on the boundary of $C_\alpha(S)$. Otherwise, it is in the interior of $C_\alpha(S)$ iff³ all of the simplices in $DT(S)$ properly containing σ_T lie in $C_\alpha(S)$ too.

7.4.1 Implicit Representation

As already mentioned in the simple algorithm stated above, steps 2 and 3 have to be passed for each value of α separately. Edelsbrunner's algorithm computes the α -shape implicitly for all values of α . This is achieved by specifying an interval $B_T(a_T, b_T)$ of α -values for every simplex σ_T in $DT(S)$, when it is part of the boundary of $C_\alpha(S)$.

$$\sigma_T \text{ is } \begin{cases} \text{not in } C_\alpha \text{ (for } \alpha < a) \\ \text{in } \partial C_\alpha \text{ (for } \alpha \in (a, b)) \\ \text{interior to } C_\alpha \text{ (for } \alpha \in (b, \infty)) \end{cases}$$

³iff: if and only if.

The calculation of these intervals is treated in detail in [2].

7.5 Conclusion

The implementation of the α -shape algorithm is quite simple and leads to good results on equally dense sampled surfaces.

The main drawback of α -shapes is obvious - it is the proper choice of the parameter α . Usually this is an interactive process where α is varied until the result "looks good", but for data sets with varying density no satisfying result is possible with "classical" α -shapes.

But there are extensions to cope with this problem:

- weighted α -shapes
- density scaling
- anisotropic scaling

A brief overview of these extensions is also given in [2]. A commercially successful version of the α -shape (Geomagic Studio [5]) is using further proprietary extensions.

Hoppe points out another problem in [40]: If the sample X is noisy, or if the underlying surface U is not sufficiently smooth, the α -shape of X will in general have finite thickness, and not be, as one would desire, a 2-dimensional manifold. It may be possible, as a post-process, to "flatten" such an α -shape into a surface.

Chapter 8

Power-Crust Algorithm

The *Power-Crust* [42] is a new definition of a shape derived from sample points of a 3D object surface. The algorithm presented in the following is based on the *Medial Axis Transformation* (refer to the Appendix). At first an approximation of the *Medial Axis* (refer to the Appendix) is computed and then an inverse transformation to produce the surface representation. The algorithm can be applied to arbitrary datasets and the output will result in a watertight boundary of a three-dimensional polyhedral solid (the Power-Crust) and an approximation of the Medial-Axis (the *Power-Shape*). There exist theoretical guarantees, that the Power-Crust of a pointset is close to the sampled surface, if the pointset fulfills the sampling condition of an "r-sample" which will be explained below. (Refer to Figure 8.1 for a first impression.)



Figure 8.1: A set of input points from the surface of an object, the watertight output mesh, and a simplified version of the approximate inner-medial axis [42].

8.1 Sketch of the Algorithm

The Medial Axis Transformation can be described as a representation of an object by infinitesimally many balls touching the object's surface in at least two points tangentially and

having maximal radii. Furtheron these balls do not intersect the object's surface and are either completely interior or exterior of the object. Since the surface is unknown and we only have a point-cloud, this transformation cannot be applied directly. If we try fitting empty balls of maximal size in the same way into the "interior" or "exterior" of the sampled points, there will be many balls with centers close to the surface intersecting the object's boundary. To avoid this, only a subset of these balls is used. Per sample point, only the two balls with centers farthest from the sample point (one in the outside and one in the inside) are selected. The centers of these balls are called the *Poles* and the balls themselves are referred to as *Polar Balls*. The union of the inner Polar Balls already gives a good approximation of the object's shape and the outer Polar Balls form a good approximation of the complement of the object. The centers of these balls are vertices of the approximated Medial-Axis (the Power-Shape), and the Power-Crust is the set of polygons (not necessarily triangles) consisting of all points, that have the same *Power-Distance* (a weighted distance function, where the weight is based on the radii of the balls) to an adjacent inner and outer ball, representing the object's surface.

8.2 Geometric Definitions [42]

Let F be the boundary of a three-dimensional object. To avoid having to deal with points located towards infinity, we assume that F is contained in a bounded open region Q . F divides Q into interior and exterior solids.

Medial Axis Transformation (MAT):

We represent a ball $B = B_{\mathbf{c},\rho} \subset Q$ by its center \mathbf{c} and radius ρ . We state that B is empty (with respect to F) if the interior of B contains no point of F . A medial ball is a maximally sized empty ball; that is, it is completely contained in no other empty ball. The center of a medial ball is either a point with more than one closest point on F , or a center of curvature of F .

Definition: The Medial Axis Transform of a surface F is the set of medial balls. The centers of the medial balls form the Medial Axis of F ; the MAT includes the radii as well. We could equivalently define the Medial Axis as the set of all points with more than one closest point on F . Notice that either way the Medial Axis includes both, a part inside of F and a part outside of F . The Medial Axis of a three-dimensional solid is generally a (non-regular) two-dimensional surface, but it accurately reflects the topology of the solid in that it has the same connected components, loops, and so on.

Power-Diagram:

Definition: The Power-Distance d_{pow} between an ordinary unweighted point \mathbf{x} in \mathbb{R}^3 and a weighted point \mathbf{c} represented by a ball $B_{\mathbf{c},\rho}$ is

$$d_{pow}(\mathbf{x}, B_{\mathbf{c},\rho}) = d^2(\mathbf{c}, \mathbf{x}) - \rho^2$$

Here function d represents the usual Euclidean distance.

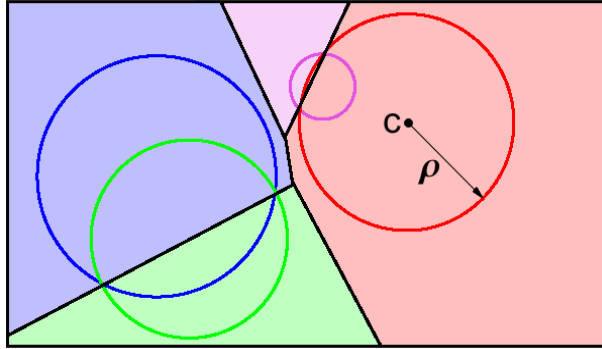


Figure 8.2: Picture of a 2D Power Diagram. Each weighted point \mathbf{c} is represented by a ball $B_{\mathbf{c},\rho}$, where the point is the center of the ball and the weight is represented by the radius ρ [43].

Definition: The Power-Diagram is a kind of weighted *Voronoi Diagram* (refer to the Appendix). It is the subdivision of Q into cells, each cell consisting of the points $\mathbf{x} \in Q$ closest—in the sense of the Power-Distance d_{pow} —to a particular weighted point $\mathbf{v} \in V$; Figure 8.2 gives a two-dimensional example.

The advantage of using d_{pow} rather than some more natural weighted distance function is that the Power-Diagram has polyhedral cells and can be computed by essentially the same algorithm as the usual Voronoi Diagram. As in Figure 8.2, the face of the Power-Diagram separating the cells of two intersecting balls is a subset of the plane where the boundaries of the two balls intersect.

8.3 Further Definitions [42]

Constructions of the Poles, the Power-Crust and the Power-Shape are based on special properties of the Voronoi Diagram of a set of points densely distributed on an object surface. Clearly it is not possible to give theoretical guarantees that the reconstructed surface is close to the original. For arbitrary sampled point sets we will therefore need to define a condition when the sampling is sufficiently dense. In practice, we want to implement the algorithm in order to give good results even when this assumption is not met.

Sampling assumption:

For the analysis we assume, that surface F is smooth and without boundary. Our assumption concerning the density of sample S is taken from [44]:

Definition: The Local Feature Size function, $f(\mathbf{x})$, is the minimum Euclidean distance from point \mathbf{x} to any point of the Medial Axis. S is an r -sample from F when the distance from any point $\mathbf{x} \in F$ to the nearest sample is at most $rf(\mathbf{x})$.

The theoretical guarantees are proved for values of r as small as 0.06, but in practice samples with $r < 0.5$ are treated quite well.

In the definition $f(\mathbf{x})$ serves as a local measure of “level of detail” on F ; when the Medial Axis is close to the surface either the curvature is high or some other patch of the surface

is nearby. Note that when F is smooth, the distance from any point to the Medial Axis is strictly greater than zero. At a sharp corner of F the Medial Axis meets the surface and according to our definition the sampling density would have to be infinite.

Poles:

In two dimensions each Voronoi vertex converges to the Medial Axis if the sample density increases towards infinity. Unfortunately this does not hold for three dimensions. Even if the sample-density is arbitrarily high, there will be Voronoi vertices close to the objects surface and far away from the Medial Axis. These critical Voronoi vertices are dual to the centers of very flat tetrahedrons in the Delaunay Triangulation and the corresponding Voronoi balls cause the "warts" in the MAT as illustrated in Figure 8.3.

The concept of Poles was first introduced in [44, 45] to cope with this problem. To get a better approximation of the Medial Axis only a subset of the Voronoi vertices is selected; for each Voronoi cell only two vertices are selected: the Poles. The selection of these special vertices is motivated by the following observation:

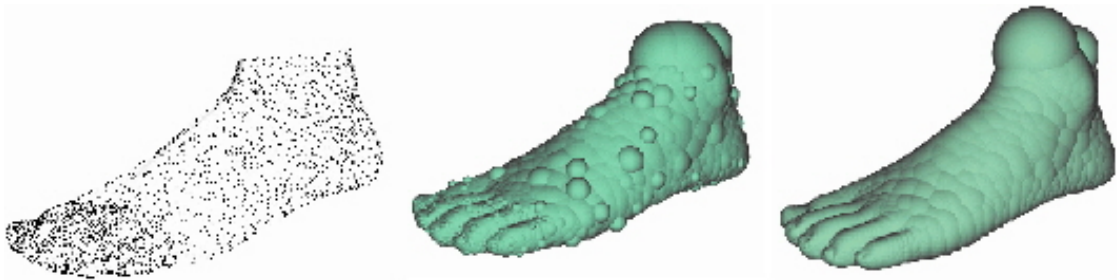


Figure 8.3: A point-cloud from the surface of a foot model, the union of all inner Voronoi balls and the union of the corresponding set of interior polar balls, forming a good approximation of the shape [43].

Where the sampling is dense, the Voronoi cell of every sample-point \mathbf{s} is long and skinny and perpendicular to the surface. This happens because in directions tangent to the surface the Voronoi cell is bounded by the proximity of other samples on the same local patch of surface; (Refer to Figure 8.4 for some intuition.) The Voronoi cell of \mathbf{s} extends perpendicularly away from the surface. It cannot extend much farther than the Medial Axis, because there \mathbf{s} is no longer the closest surface point and samples on some other patch of surface will be closer. Thus, the Voronoi vertices at the two ends of the long and skinny Voronoi cell should lie near the Medial Axis. This motivates the selection of the Poles as an approximation of the Medial Axis.

Definition: The Poles of a sample $\mathbf{s} \in S$ are the farthest vertex of its Voronoi cell in the interior of F and the farthest vertex of its Voronoi cell on the exterior of F .

Let V be the set of Poles, for all $\mathbf{s} \in S$. If S is a dense sample, the set V of Poles excludes Voronoi vertices close to F and forms a good estimate of the Medial Axis, albeit as a discrete set of points. Note also that the vectors from \mathbf{s} to its Poles approximate the surface normals at \mathbf{s} .

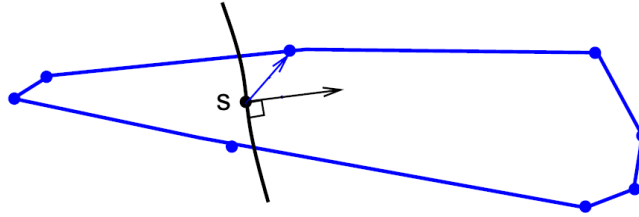


Figure 8.4: The Voronoi cell of sample s is long and skinny and perpendicular to the surface [42].

Polar Balls:

Definition: A Polar Ball $B_{\mathbf{c},\rho}$ is a Voronoi ball, having a Pole at its center \mathbf{c} , and a radius ρ which is the Euclidian-distance to the nearest sample point.

Power-Crust:

The Power-Crust is the estimated surface of the algorithm. For a formal definition we consider the Power-Diagram of the Polar Balls, which subdivides \mathbb{R}^3 into a set of cells.

Definition: The Power-Crust is the boundary between the Power-Diagram cells belonging to inner Poles and Power-Diagram cells belonging to outer Poles.

Since most points of the interior solid bounded by F are inside the union of the inner Polar Balls, and outside of the union of outer Polar Balls, they belong to cells of the Power-Diagram corresponding to inner Poles. Similarly most points in the exterior solid belong to cells corresponding to outer Poles. A two-dimensional face of the Power-Crust separates cells corresponding to an inner and an outer Pole. The two Polar Balls should intersect shallowly, if at all, since the inner Polar Ball is mostly inside the object and the outer Polar Ball is mostly outside. Thus the Power-Crust face lies near the boundaries of both unions of balls, and hence near the boundary F of the object.

Power-Shape:

The definition of the Power-Crust implies a way to connect the Poles to form a topologically correct approximation of the Medial Axis as a simplicial complex M , which we call the Power-Shape. The vertices of M are the Poles themselves. Inner Poles whose cells are adjacent in the Power-Diagram are connected by simplices in M , as are adjacent outer Poles. The Power-Shape is a subset of the weighted Delaunay Triangulation (also known as the regular triangulation) dual to the Power-Diagram, just as the Delaunay Triangulation is dual to the usual unweighted Voronoi Diagram. While the Medial Axis of F is a two-dimensional surface, the Power-Shape generally contains some very flat, but solid, tetrahedra.

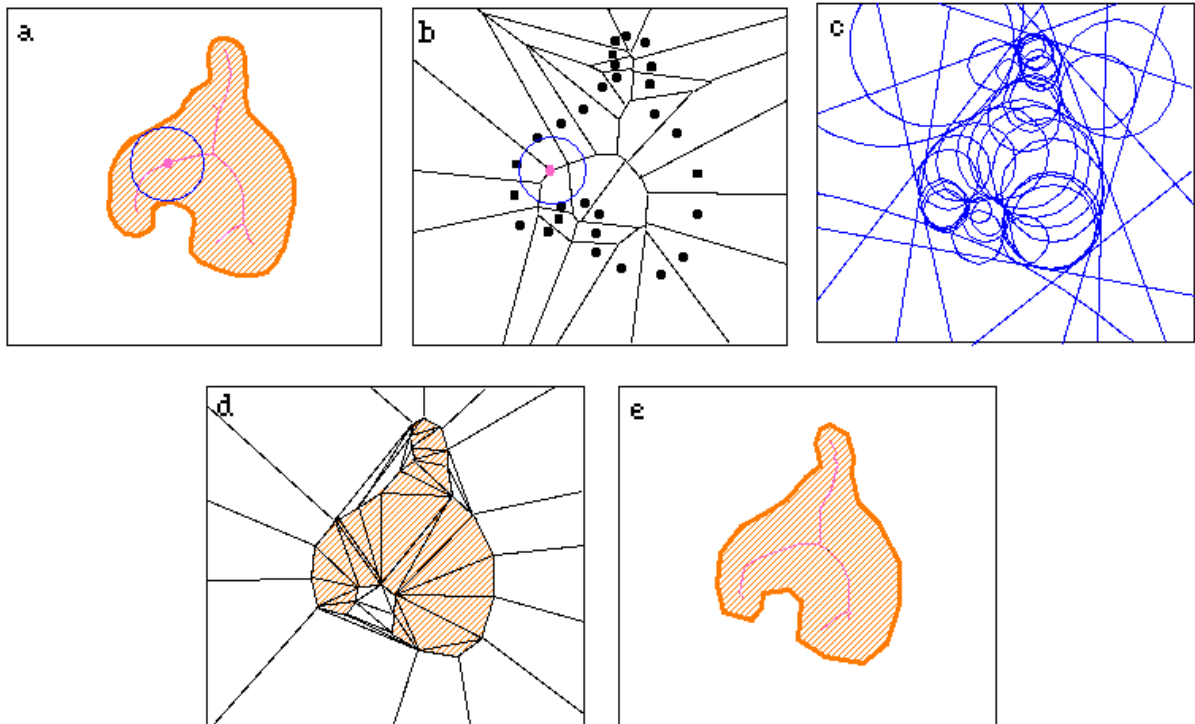


Figure 8.5: A two-dimensional example of the Power-Crust construction:

a.) An object with its Medial Axis (shown in violet). b.) The Voronoi Diagram of a point sample S from the object boundary, with a Voronoi ball surrounding one of the Poles. In 2D all Voronoi vertices can be considered Poles but not in 3D. c.) The inner and outer Polar Balls. The infinite Polar Balls degenerate to half spaces. d.) The Power-Diagram cells of the Poles. e.) The Power-Crust and the Power-Shape of the interior solid [43].

8.4 Basic Algorithm [42]

Overview

The basic algorithm is a straightforward reflection of our strategy: First estimate the MAT, and then use it to define the surface approximation (refer to Figure 8.5).

1. Adding vertices of a bounding box
2. Compute the Voronoi Diagram of the sample points S
3. For each sample point compute its Poles.
4. Compute the Power-Diagram of the Poles.
5. Label each Pole either inside or outside.
6. Compute the Power-Diagram faces separating the cells of inside and outside Poles as the Power-Crust.
7. Compute the Delaunay Triangulation faces connecting inside Poles as the Power-Shape.

Adding vertices of a bounding box

To avoid dealing with Poles in infinity, we add a set Z of eight points, the vertices of a large box surrounding F to S so that both Poles of each sample in S are bounded. In our implementation, this box is five times larger than the minimum bounding box of S .

Voronoi Diagram

The computation of the Voronoi Diagram is as usual, the main time-consuming part of the algorithm. In [42] Kern Clarkson's Hull [73] has been chosen which performs $O(n^2)$ in the worst case, but applied to a dataset of sampled surfaces a computation in near linear time is achieved.

This is approved in a recent report [46] of Delaunay Triangulations applied to data from scanned surfaces, where the best performing algorithms are capable of triangulating $1E+6$ points in about 400 seconds on a Pentium III workstation.

Selection of the Poles

Having computed the Voronoi Diagram of the sample points S , the two Poles for each sample point \mathbf{s} are selected. A Pole is a vertex in the Voronoi-Diagramm. The first Pole $\mathbf{p}_1(\mathbf{s})$ is simply the farthest Voronoi vertex \mathbf{v} of the Voronoi cell surrounding \mathbf{s} . The other Pole $\mathbf{p}_2(\mathbf{s})$ is the farthest vertex such that the two vectors from \mathbf{s} to $\mathbf{p}_1(\mathbf{s})$ and \mathbf{s} to $\mathbf{p}_2(\mathbf{s})$ make an angle of more than $\pi/2$.

It can be proved, if F is an r -sample for an r small enough -that \mathbf{p}_1 and \mathbf{p}_2 are indeed the farthest Voronoi vertices of \mathbf{s} on either side of F .

Computing the Power-Diagramm

A Delaunay Triangulation program with the special distance function d_{pow} has to be adapted for this task.

Labeling Poles inside or outside

Having computed the Power-Diagram of the Polar Balls, a graph on the Power-Diagram cells is defined: Two cells are connected in the graph if they share a two-dimensional face. In addition, two cells are connected if they belong to the two Poles of the same sample \mathbf{s} . By traversing this graph the Poles are labeled "inner" or "outer".

The determination is based on two lemmas:

- If one Pole of \mathbf{s} is the outer Pole the other is the inner and vice versa.
- If Polar Balls of neighboring Poles intersect shallowly they are likely to be of the opposite kind, otherwise deeply intersecting Poles are likely to be of the same kind (both inner or both outer).

A naive algorithm would begin by labeling Poles adjacent to points forming the bounding box Z as outer and then propagating labels as follows:

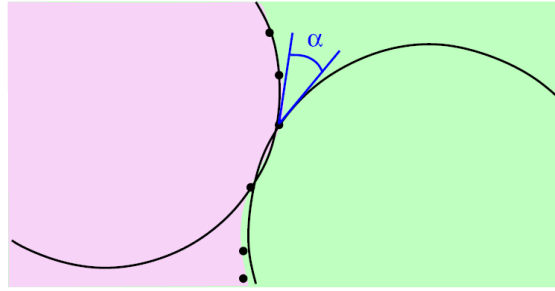


Figure 8.6: An inner and an outer Polar Ball can intersect shallowly, if at all, because the dense set of samples separates them. Angle α measures the depth of the intersection [42].

If a Pole \mathbf{p} , labeled "outer", has got an unlabeled neighbor \mathbf{q} , this neighbor is also regarded as outer, provided the Polar Balls of \mathbf{p} and \mathbf{q} intersect deeply (that is, the angle α in Figure 8.6 is large).

And for each sample \mathbf{s} for which \mathbf{p} is a Pole (there might be more than one), the other Pole of \mathbf{s} is labeled "inner". Inner Poles are labeled similarly: Deeply intersecting neighbors get labeled "inner", and the opposite Pole of the same sample gets labeled "outer". But because the sampling assumption is not met everywhere, a naive implementation could fail dramatically - once an error is made, it propagates.

As a solution, the order in which the Poles are propagated is determined by using the following greedy heuristic: Since an unlabeled Polar Ball can be "inner" or "outer", based on the labels already assigned, the labels of the most confidently identified Poles are marked and propagated first.

In [42] this strategy is observed performing successfully on all sample data sets.

8.5 Extensions [42]

Applied to well sampled datasets the Power-Crust is geometrically close to the original surface, but in practice many datasets do not meet the sampling condition. Hence to produce good results the following extensions are added:

Omitting Poles:

The quality of the Power-Crust depends on how well Polar Balls approximate the MAT. Above we mentioned that well sampled datasets result in skinny Voronoi cells. In practice, due to not smooth surfaces or too sparse sampled datasets, this is not fulfilled and the Polar Balls give no suitable approximation for the Medial Axis.

Thus for a robust implementation a lower bound of $f(\mathbf{s})$ is calculated for every sample point \mathbf{s} . Then both Poles of \mathbf{s} are checked, if their distance to \mathbf{s} , $d(\mathbf{s}, \mathbf{p}_{1,2})$ is greater than this lower bound of $f(\mathbf{s})$, if it is not, this Pole is omitted.

It has been observed, that the computation of the Power-Crust from this partial MAT produces still good models.

Sharp corners

Near a sharp corner, the Poles on the inside of the corner will fail the above ‘skinnyness’ test, while those on the outside will pass. But discarding only the Poles on the inside of the corner causes the Power-Crust algorithm to collapse and to round off the corner. Instead, if the user indicates that the model contains sharp corners, we discard both \mathbf{p}_1 and \mathbf{p}_2 for any sample that fails the ‘skinnyness’ test. Nearby Power-Crust faces extend into the region which is left uncovered by the discarded Poles, extending adjacent smooth surfaces linearly into the empty region, until they meet at a sharp angle. Figure 8.7 shows a two-dimensional example of this behavior.

Notice that a sharp edge can be reconstructed nicely even though there are no sample points on the edge itself.

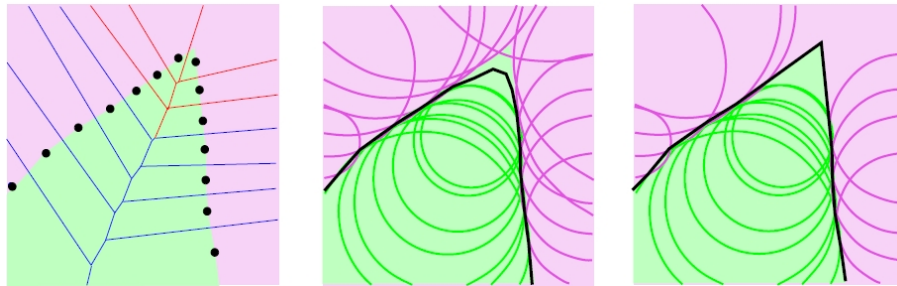


Figure 8.7: Forming of sharp corners by omitting poles on the inside and outside [42].

Holes:

Basically the Power-Crust is a watertight surface, however, to represent objects having holes a further extension is needed. The idea behind this is that small holes due to undersampling are filled but large ones are left open. Holes can be characterized by a deep intersection of adjacent inner and outer Polar Balls. So polygons with deep intersecting Polar Balls (compared to a user specified parameter) are omitted.

8.6 Conclusion

Power-Crust is an algorithm with theoretical guarantees and leads to well shaped watertight objects on r -sampled datasets. We observed the Power-Crust to produce a good approximation to the original surface even if the dataset does not fulfill the sampling conditions.

As a drawback the Power-Crust algorithm introduces extra points in the output and produces a polyhedral not necessarily a triangulated surface [50].

Chapter 9

Cocone Algorithm

The *Cocone* is a surface reconstruction algorithm, based on the *Crust* [44] algorithm. Cocone computes a piecewise linear approximation to a smooth surface without boundaries from a set of sample points. The estimated surface consists only of triangles and in contrast to Power-Crust no extra points are generated, therefore all vertices are sample points; in addition all faces are Delaunay-Faces. It is also more efficient in the way that only one Delaunay Triangulation is needed. It can be proved that the algorithm produces a surface that is close to the original (sampled) surface, when applied to an ε -sampled (see below) dataset. We will also shortly discuss extensions of the basic algorithm.

9.1 Definitions [48]

Let the surface Σ be a smooth manifold without boundaries embedded in \mathbb{R}^3 .

Local Feature Size:

We define a distance function called Local Feature Size:

The Local Feature Size function, $f(\mathbf{x})$, is the minimum Euclidean distance from point \mathbf{x} to any point of the Medial Axis (refer to the Appendix).

ε -sample:

We use the same sampling condition as described above in the Power-Crust Algorithm:

A sample P of Σ is an ε -sample if each point $\mathbf{x} \in \Sigma$ has a sample point $\mathbf{p} \in P$ that

$$\|\mathbf{x} - \mathbf{p}\| \leq \varepsilon f(\mathbf{x})$$

In the following we will denote any ray from \mathbf{p} to a point $\mathbf{y} \in V_{\mathbf{p}}$ as $\hat{\mathbf{y}}$.

Poles:

The farthest Voronoi vertex $\mathbf{v}_{\mathbf{p}}$ in $V_{\mathbf{p}}$ is called the positive Pole of \mathbf{p} . We call $\hat{\mathbf{v}}_{\mathbf{p}} = \mathbf{v}_{\mathbf{p}} - \mathbf{p}$, the Pole vector for \mathbf{p} . If $V_{\mathbf{p}}$ is unbounded, $\mathbf{v}_{\mathbf{p}}$ is taken at infinity, and the direction of $\hat{\mathbf{v}}_{\mathbf{p}}$ is taken as the average of all directions given by the unbounded Voronoi edges.

Cocone (Co-Cone):

The set $C_{\mathbf{p}} = \{\mathbf{y} \in V_{\mathbf{p}} : \angle \widehat{\mathbf{y}}, \widehat{\mathbf{v}}_{\mathbf{p}} > \frac{\pi}{2} - \Theta\}$ is called the Cocone of \mathbf{p} . In words: $C_{\mathbf{p}}$ is the complement of the double cone (clipped within $V_{\mathbf{p}}$) with apex \mathbf{p} and an opening angle of $\frac{\pi}{2} - \Theta$ around the axis aligned with the Pole vector $\widehat{\mathbf{v}}_{\mathbf{p}}$. See Figure 9.1 for an example of a Cocone.

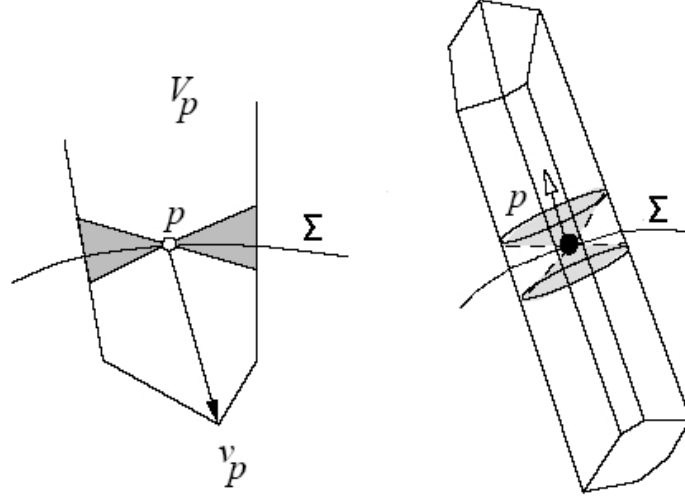


Figure 9.1: The Cocone clipped in the Voronoi cell of a sample point \mathbf{p} in two dimensions (left) and three dimensions (right). Σ denotes the sampled surface. In the left the Cocone (the Complementary Cone) is shaded, in the right its boundary is shaded [47].

Restricted Voronoi-Diagram:

A cell in the restricted Voronoi Diagram $V_{\mathbf{p},\Sigma}$ is defined as the intersection of a Voronoi cell $V_{\mathbf{p}}$ with Σ . $V_{P,\Sigma} = \{V_{\mathbf{p},\Sigma} = V_{\mathbf{p}} \cap \Sigma \mid \mathbf{p} \in P\}$

Restricted Delaunay Triangulation:

As with Voronoi Diagrams we can define a simplicial complex dual to the restricted Voronoi Diagrams.

$k + 1$ points in P form a Delaunay k -simplex σ_T , if their restricted Voronoi cells have a non-empty intersection:

$$D_{P,\Sigma} = \{\sigma_T \mid \cap V_{\mathbf{p},\Sigma} \neq \emptyset \text{ where } \mathbf{p} \text{ is any vertex in } \sigma_T\}$$

Specifically, an edge \mathbf{pq} is in $D_{P,\Sigma}$ iff $V_{\mathbf{p},\Sigma} \cap V_{\mathbf{q},\Sigma}$ is non-empty, a triangle \mathbf{pqr} is in $D_{P,\Sigma}$ iff $V_{\mathbf{p},\Sigma} \cap V_{\mathbf{q},\Sigma} \cap V_{\mathbf{r},\Sigma}$ is non-empty.

9.2 Theoretical Background

Since our sampling condition is based on the local feature size $f(\mathbf{x})$ we will mention two helpful properties of $f(\mathbf{x})$ first:

Lemma (Lipschitz Continuity):

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \|\mathbf{x} - \mathbf{y}\| \quad (9.1)$$

for any two points \mathbf{x} and \mathbf{y} in Σ .

Lemma (Feature Translation):

For any two points \mathbf{x} and \mathbf{y} in Σ with $\|\mathbf{x} - \mathbf{y}\| \leq \varepsilon f(\mathbf{x})$ we have

$$\|\mathbf{x} - \mathbf{y}\| \leq \frac{\varepsilon}{1 - \varepsilon} f(\mathbf{y}). \quad (9.2)$$

The lemmas below demonstrate, that the Pole-vector $\widehat{\mathbf{v}}_{\mathbf{p}}$ of a sample point \mathbf{p} gives a good approximation of the surface normal $\mathbf{n}_{\mathbf{p}}$ at \mathbf{p} and that the surface-normals from neighboring sample points do not vary much for ε -sampled pointsets and small values of ε .

Lemma (Medial Lemma):

Let \mathbf{m}_1 and \mathbf{m}_2 be the two Medial Axis points which are the centers of the two Medial Balls at \mathbf{p} . The Voronoi cell $V_{\mathbf{p}}$ must contain \mathbf{m}_1 and \mathbf{m}_2 .

Lemma (Normal):

Let \mathbf{v} be any point in $V_{\mathbf{p}}$ with $\|\mathbf{v} - \mathbf{p}\| \geq \mu f(\mathbf{p})$, furtheron let P be an ε sample of Σ . Then

$$\angle(\mathbf{v} - \mathbf{p}), \mathbf{n}_{\mathbf{p}} \leq \arcsin \frac{\varepsilon}{\mu(1 - \varepsilon)} + \arcsin \frac{\varepsilon}{1 - \varepsilon} \quad (9.3)$$

Lemma (Pole):

The angle between the normal $\mathbf{n}_{\mathbf{p}}$ at \mathbf{p} and the Pole vector $\widehat{\mathbf{v}}_{\mathbf{p}}$ satisfies

$$\angle \mathbf{n}_{\mathbf{p}}, \widehat{\mathbf{v}}_{\mathbf{p}} \leq 2 \arcsin \frac{\varepsilon}{1 - \varepsilon} \quad (9.4)$$

Lemma (Normal Variation):

Let \mathbf{x} and \mathbf{y} be any two points with $\|\mathbf{x} - \mathbf{y}\| \leq \rho f(\mathbf{x})$ for $\rho < \frac{1}{3}$. Then we have

$$\angle \mathbf{n}_{\mathbf{x}}, \mathbf{n}_{\mathbf{y}} \leq \frac{\rho}{1 - 3\rho} \quad (9.5)$$

Lemma (Edge Normal):

The angle $\angle \mathbf{t}_{\mathbf{p}\mathbf{q}}, \mathbf{n}_{\mathbf{p}}$ is more than $\frac{\pi}{2} - \arcsin \frac{\|\mathbf{p} - \mathbf{q}\|}{2f(\mathbf{p})}$

$$\angle \mathbf{t}_{\mathbf{p}\mathbf{q}}, \mathbf{n}_{\mathbf{p}} \geq \frac{\pi}{2} - \arcsin \frac{\|\mathbf{p} - \mathbf{q}\|}{2f(\mathbf{p})} \quad (9.6)$$

A proof all of these Lemmas is given in the Appendix of this thesis.

9.3 Basic Algorithm

The main theorem in [47] states, that from a set of triangles -which will be referred to as the candidate triangles T - a 2-manifold N can be extracted that is geometrically close and homeomorphic¹ to the original surface Σ , if the following three conditions are met:

1. T contains all triangles whose dual Voronoi edges intersect Σ . (This is equivalent to the set of all restricted Delaunay triangles)
2. Each triangle in T is small, that is, the radius of its circumcircle is much smaller than the distance to the Medial Axis at its vertices.
3. All triangles in T are "flat", that is, the triangle normals form small angles with the surface normals at their vertices.

In [47] further proofs are given, that all restricted Delaunay triangles fulfill the conditions 2 and 3, thus the restricted Delaunay Triangulation gives a good approximation of the original surface. Unfortunately, to compute the restricted Delaunay Triangulation, the surface must be known in advance. As a solution the restricted Voronoi Diagram is approximated using the concept of Co-Cones (as explained below), by computing a set that includes all of the restricted Delaunay triangles (and maybe some more). This set is pruned to extract a manifold surface as an output [48].

So the complete Cocone algorithm consists of the following steps:

- Delaunay Triangulation of the pointset $P \iff$ calculation of the Voronoi Diagram V_P ,
- approximation of the restricted Voronoi Diagram $V_{P,\Sigma}$ and calculation of the candidate triangles T ,
- extract a manifold N out of T .

9.3.1 Approximation of the Restricted Voronoi Diagram [48]

First we observe that each restricted Voronoi cell $V_{\mathbf{p},\Sigma}$ is almost flat if the sample-density is sufficiently high. This follows from the Normal Variation Lemma, as points in $V_{\mathbf{p},\Sigma}$ cannot be far apart if ε is small. In particular $V_{\mathbf{p},\Sigma}$ lies within a small neighborhood of the tangent plane $\tau_{\mathbf{p}}$ at \mathbf{p} . So we need two approximations: (i) an approximation for $\tau_{\mathbf{p}}$ (or equivalently to $\mathbf{n}_{\mathbf{p}}$), (ii) an approximation for $V_{\mathbf{p},\Sigma}$ based on the approximation of $\mathbf{n}_{\mathbf{p}}$. Concerning these observations $V_{\mathbf{p},\Sigma}$ must lie within the volume of the complement of a double cone. In the following it will be shown that this cone-complement is the Cocone $C_{\mathbf{p}}$ having an opening angle of $\frac{\pi}{2} - \Theta$, where $\Theta \leq \frac{\pi}{8}$ for an ε -sample with $\varepsilon \leq 0.1$.

Recalling the Pole Lemma, the Pole vector $\widehat{\mathbf{v}}_{\mathbf{p}}$ approximates $\mathbf{n}_{\mathbf{p}}$ (the surface normal at \mathbf{p}) well: In particular the angle between $\angle \mathbf{n}_{\mathbf{p}}, \widehat{\mathbf{v}}_{\mathbf{p}}$ is smaller than $\alpha = 2 \arcsin \frac{\varepsilon}{1-\varepsilon}$.

Furthermore, considering the Edge Normal Lemma we can show that the angle between $\mathbf{n}_{\mathbf{p}}$ and the vector from \mathbf{p} to any point \mathbf{y} in its restricted Voronoi cell is at least $\beta = \frac{\pi}{2} - \arcsin \frac{\varepsilon}{2(1-\varepsilon)}$.

¹topologically equivalent

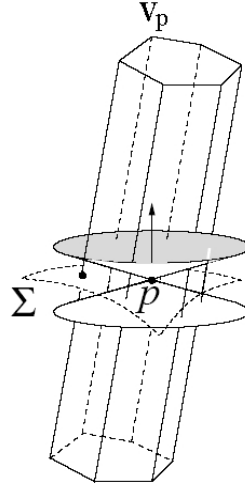


Figure 9.2: The positive pole $\hat{\mathbf{v}}_p$ helps to estimate the normal \mathbf{n}_p and the cocone helps to calculate $V_{p,\Sigma}$ [48].

Proof: The distance $\|\mathbf{y} - \mathbf{p}\| \leq \varepsilon f(\mathbf{y})$, since $\mathbf{y} \in V_{p,\Sigma}$ and P is an ε -sample. By the Lipschitz Lemma $f(\mathbf{y}) \leq f(\mathbf{p}) + \|\mathbf{y} - \mathbf{p}\|$ giving $f(\mathbf{y}) \leq \frac{1}{1-\varepsilon}f(\mathbf{p})$, and hence $\|\mathbf{y} - \mathbf{p}\| \leq \varepsilon f(\mathbf{y}) \leq \frac{\varepsilon}{1-\varepsilon}f(\mathbf{p})$. Plugging this into the Edge Normal Lemma yields the result. \square

The conclusion of these two observations is, that the angle between $\hat{\mathbf{v}}_p$ and the ray from any $\mathbf{y} \in V_{p,\Sigma}$ to \mathbf{p} is at least $\beta - \alpha$. Therefore the restricted Voronoi cell $V_{p,\Sigma}$ is completely contained between in the Cocone of \mathbf{p} with

$$\Theta \leq \frac{\pi}{2} - \beta + \alpha \quad (9.7)$$

as illustrated in Figure 9.2. By plugging in the values of α, β we obtain:

$$\Theta \leq \arcsin \frac{\varepsilon}{2(1-\varepsilon)} + 2 \arcsin \frac{\varepsilon}{1-\varepsilon} \quad (9.8)$$

For an ε -sample with $\varepsilon \leq 0.1$ we obtain generous upper bounds:

$$\Theta \leq \varepsilon + \frac{\pi}{8} - \varepsilon = \frac{\pi}{8}. \quad (9.9)$$

Now having determined the opening angle of the Cocone, the computation of the candidate triangles is straightforward. For every Voronoi cell V_p we mark all edges of the cell, that contain at least one point that is in the Cocone C_p . If an edge is marked by all of its three adjacent Voronoi cells we accept its dual Delaunay triangle as a candidate triangle. Obviously this set contains all restricted Delaunay triangles, but probably a few more, so we have to reject triangles as will be explained in the following section.

9.3.2 Manifold extraction [48, 49]

This step first prunes away triangles that are incident to sharp edges; an edge e is sharp, if any two consecutive triangles around it form an angle of more than $\frac{3\pi}{2}$. Edges with a single triangle are also sharp by this definition. This pruning is carried out in an cascading

manner, since by deleting these triangles other sharp edges can be created. After pruning we still retain the set of restricted Delaunay triangles, since adjacent restricted Delaunay triangles of sufficient dense sampled datasets do not meet at sharp angles. However, the remaining triangles may still not form a surface. They may form "layers" creating a non-manifold. A manifold surface is extracted out of this possibly layered set by walking inside (or outside) the space covered by these triangles. In addition these triangles have to be orientated consistently as well.

A pseudo code of the cascading pruning step is listed below where the candidate triangles are referred to as K .

Procedure 9.1 PRUNING(K)

```

1: Pending := 0
2: for each edge  $e \in K$ 
3:   Pending.push( $e$ )
4: while Pending  $\neq$  0
5:    $e :=$  Pending.pop()
6:   if IsSharp( $e$ )=true
7:     for each  $t \in e.Triangles$ 
8:        $K := K \setminus \{t\}$ 
9:       for each  $e' \in t.Edges \setminus \{e\}$ 
10:        Pending.push( $e'$ )
11: return  $K$ 

```

First the stack *Pending* is initialized empty (line 1). Then all edges in the complex K are pushed onto this stack (lines 2 and 3). Together with every triangle t we sort a list *Edges* of its edges. With each edge e we store a set *Triangles* of triangles incident to e . We assume that we have a function *IsSharp* that requires an edge e as input and returns true if e is sharp and false if e is not sharp. As long as the stack *Pending* is not empty, an edge e is popped from the stack. If this edge is sharp, all those incident triangles are removed from the complex K . All edges other than e that are incident to the deleted triangle are pushed onto the stack *Pending* (lines 4-10), since these edges may become sharp due to the deletion of the triangle t . Finally the reduced complex K is returned (line 11).

The next step WALK extracts a manifold.

First we initialize the set *Surface* empty (line1). Then we choose an arbitrary triangle t from the complex K , orient it and insert it into the *Surface* (lines 2 and 3). Next we initialize the stack *Pending* empty (line 4). From the orientation of the chosen triangle t we derive an orientation for all edges incident to t . These edges are stored in a field *Edges* associated with every triangle. We denote an oriented edge e by \vec{e} . For each oriented edge \vec{e} of a triangle t we push a pair (\vec{e}, t) onto the stack (lines 5 and 6). As long as the stack *Pending* is not empty we pop its top element (\vec{e}, t) (line 8). If the edge e is not processed so far we use the field *processed* to mark it processed and compute the surface neighbor of (\vec{e}, t) , i.e. the triangle t' incident to e that "best fits" t (lines 9-11). If t' does not exist, the triangle t is incident to a boundary. Otherwise we insert t' in the set *Surface* (lines 12 and 13). We assume that the function *SurfaceNeighbor* orients t' such that its orientation matches

the orientation of t and push all the pairs of orientated edges \vec{e}' besides \vec{e} incident to t' together with the triangle t' itself onto the stack Pending (lines 14-15). Finally we return the surface Surface (line 16).

The above method works under the assumption, that the surface is orientable, i.e., surfaces like the Möbius strip are not allowed.

Procedure 9.2 WALK(K)

```

1: Surface := 0
2: choose arbitrary oriented  $t \in K$ 
3: Surface.insert( $t$ )
4: Pending := 0
5: for each  $e \in t.Edges$ 
6:   Pending.push(( $\vec{e}, t$ ))
7: while Pending  $\neq 0$ 
8:   ( $\vec{e}, t$ ) := Pending.pop()
9:   if  $e.processed = \text{false}$ 
10:     $e.processed := \text{true}$ 
11:     $t' := \text{SurfaceNeighbor}(K, \vec{e}, t)$ 
12:    if  $t' \neq 0$ 
13:      Surface.insert( $t'$ )
14:      for each  $\vec{e}' \in t'.Edges \setminus \{\vec{e}\}$ 
15:        Pending.push(( $\vec{e}', t'$ ))
16: return Surface

```

The walking step uses a function "SurfaceNeighbor", which selects the topmost triangles among a set of triangles incident to e whose normals (orientated according to the orientation of t) make an angle smaller than $\frac{\pi}{2}$ with the normal of t , see Figure 9.3.

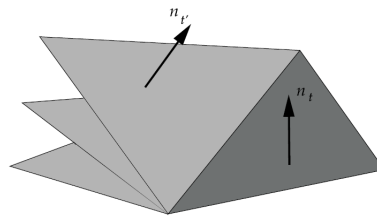


Figure 9.3: The surface neighbor of the right dark shaded triangle is the topmost triangle among the light shaded triangles [49].

9.4 Conclusion and Extensions

In theory Cocone produces a piecewise-linear manifold close to the sampled surface. In practice the required sampling assumptions are often not met and the manifold extraction is not eminently robust.

There are several extensions to the basic algorithm:

- In [49] a boundary detection (based on the geometry of the Voronoi cells, in particular the ratio of cell-height to width) is described. Thus together with an adapted manifold extraction surfaces with boundaries can be constructed as well.
- Tight-Cocone [50] produces watertight surfaces, using Cocone for a preliminary surface computation. This extension consists of a hole detection and a hole filling step.
- Super-Cocone [51] is a variant of the Cocone algorithm intended to deal with very large datasets. This is achieved by partitioning the pointset into octree-boxes, and then carefully matching adjacent surface patches. The reconstruction of pointsets with 2 million points is accomplished in about 100 minutes using a Pentium III workstation.

Part III
Experiments

Chapter 10

Experimental Setup

The aim of this thesis was to establish a 3D surface scanning environment as a basis for further developments. We extended a commercial touch-probe scanner to contactless operation and developed a user friendly software package for the scanning- and reconstruction process. In this chapter we will describe this scanning system, the reconstruction software and present some results.

10.1 Hardware

Our scanner is built upon a commercial hand-guided 3D touch-probe scanner, the Immersion Microscribe 3DX [74], that was available in our laboratory.

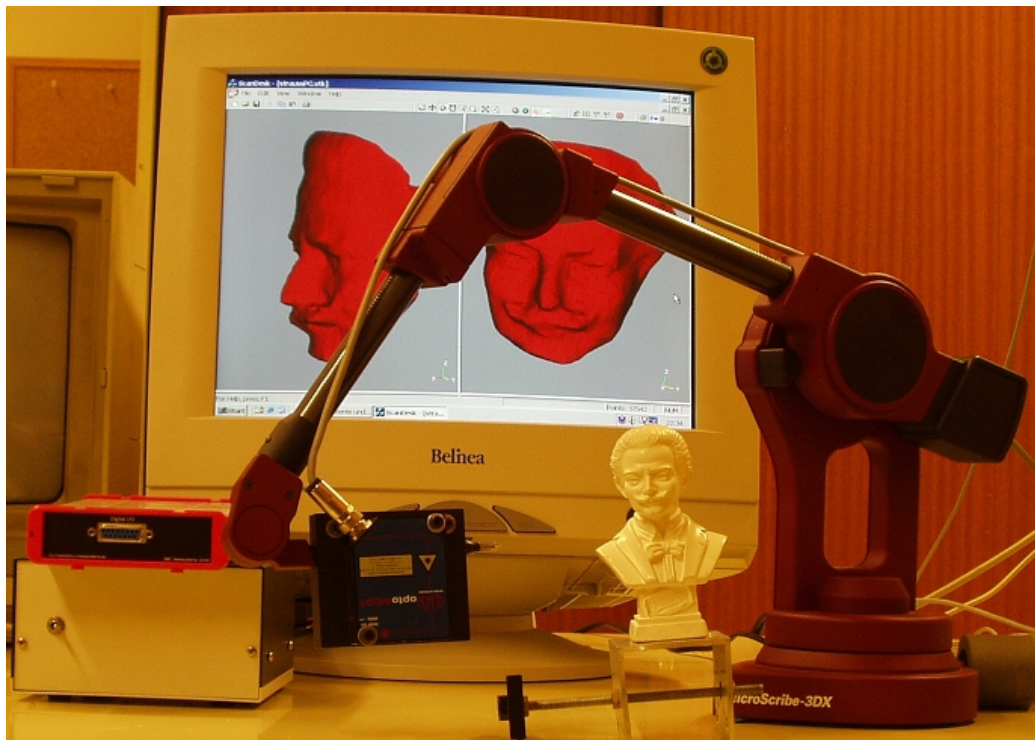


Figure 10.1: Photograph of the 3D scanning-system in our laboratory.

10.1.1 Touch-Probe Scanner

The Microscribe 3DX is a hand-guided touch-probe scanner. The working principle is rather simple. The scanner -looking like a robot arm- has several segments made of carbon-fiber tubes, that are linked by articulated joints. Thus the arm can be moved in any direction. The last segment of the arm -referred to as stylus- has a metal tip representing the actual sampling position. In each of the articulated joints in-between two segments of the arm there is an optical sensor measuring the angle between these two segments. The Microscribe has 5 degrees of freedom therefore there are 5 of these angular decoders. The scanner's heavy base houses a microprocessor that provides each of the decoded angles by communication via a RS232-interface.

During the scanning process the sampling probe is moved by hand across the surface of the object of interest.

Knowing all the angles of the joints and lengths of each segment the position of the stylus tip (the direction of the stylus as well) can be computed.

The arm itself is counterbalanced and features low friction bearings to provide a smooth and effortless manipulation.

The optical decoders in the joints work similar to those in a computer mouse. Since they are only capable of providing incremental information, the scanner must be calibrated. This is achieved by setting up the scanner in a given initial position, where each of the arm's joints form defined angles.

The overall arm length determines the maximum scanning volume. This is a sphere with about 1m in diameter. In practice the measuring volume is always smaller since the stylus tip needs to touch the object and thus the freedom of movement of the other arm's segments is limited. The achieved position accuracy of $\sim 0.3\text{mm}$ is quite good.

The hand-guided operation allows selective sampling with varying point densities, since only one point is captured in a single measurement step. On the other hand scanning is rather time consuming. Furthermore to obtain a sufficient dense sample data set and to avoid over-sampling a skilled operator and an indication or feedback of already scanned parts is necessary. The contact based measurement allows to scan even transparent media, but there are several drawbacks. If too much force is applied the metal tip can scratch the object, moreover during the measurement the object to be digitized can be displaced and thus the measurement disturbed. In addition when sampling objects with rough or complex surfaces it is difficult to keep the stylus tip in contact with the surface.

To avoid all these problems, this scanner has been extended into a non-contact system by adding a laser triangulation sensor.

10.1.2 Laser Triangulation Sensor

We were looking for a small distance sensor with high precision and so have finally chosen the optoNCDT 1400, a laser probe that is manufactured by Micro Epsilon [61].

The Micro Epsilon optoNCDT 1400 (refer to Figure 10.2) is an optoelectronic displacement measurement system with an integrated digital signal processor. The system measures distances without contact against a wide variety of material surfaces, using the (active) triangulation principle. A laser diode projects a visible light spot onto the surface of the target. The spot is imaged onto a CCD-array by the receiver lens. In the integral controller

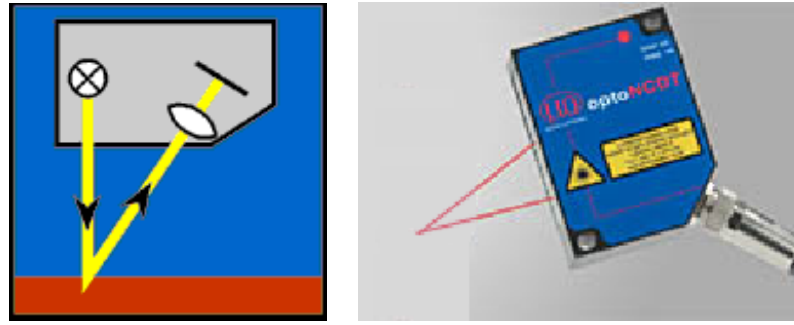


Figure 10.2: Schematics and photography of the laser probe [61].

the measured values are digitally processed; an analog current signal is available as output. The sensor has a rather small triangulation-angle of $\sim 30^\circ$ and is thus not very vulnerable to shadowing effects.

This probe allows the measurement of distances in the range between 25 and 75mm with an accuracy of $20\mu\text{m}$. The sampling rate is 1kHz.

The limited measurement range is no constraint to the scanner, because a larger range would lead to higher inaccuracies due to the limited angle resolution of the scanner.

An advantage over laser stripe projections is that the intensity is higher and therefore the measurement is more robust against textures and surfaces with low reflectance.

We experienced that measurements of the probe were reliable even for such "non-cooperative" objects, but to avoid any problems concerning surface texture our test objects were painted white.

10.1.3 Mounting the Laser Triangulation Sensor

One thing in extending the scanner to non-contact operation is to physically mount the laser probe onto the arm's stylus. We first thought of mounting the laser-probe onto the digitizer so that the beam is coaxial to the arm's stylus. However, since this means the tip has to be replaced and therefore the calibration of the arm would be hard to realize, we opted for a setup where the stylus and the laser beam are collinear. As a further benefit of this setup we can use both the laser probe and the stylus for measurements. As a drawback we have to know two additional offset-parameters compared to a coaxial mounting (the displacement of the laser-probe relative to the axis of the stylus tip). All three offset-parameters are found by an interactive calibration process:

By targeting the same point once with the tip and then again with the laser spot, we get three equations and thus the three offset parameters can be calculated by the scanning software. This process involves human skills of observation (targeting the same point), hence for a more automated implementation we suggest to have a special "calibration-target" that features a small cavity (so that the stylus tip has a defined position) and a light detector just below this mark. Thus, if the laser spot is projected exactly onto the calibration point the signal of this detector can be used to trigger the calibration process. However since our probe mounting kept the probe firmly in place we did not need to calibrate very often. Therefore satisfying calibration-parameters were found by using the above mentioned method refined by averaging.

Having the laser probe physically attached to the scanner with a holder precisely milled of

a foamed urea block, the next step is to combine the distance measurement from the laser probe with the position of the stylus in order to calculate the coordinates of the laser spot projected onto the surface.

To make the current signal of the laser-probe available to the computer it was converted into a voltage and then digitized by a 12 Bit AD-converter attached to the USB-port of our computer, the "BMC-Messsysteme" MEM-ADDA-USB [75] interface. Knowing the displacement of the sensor relative to the stylus, the stylus direction, and the beam length the position of the light spot can be calculated. In practice this calculation is very simple since in the routines of the SDK (Software Development Kit) for the scanner parameters can be set to operate the scanner with others than the standard stylus tip. Thus prior the calculation of the tip position we set the parameters according to the offset values respectively the beam length.

Unfortunately the selected AD-converter has a sampling rate of $\sim 100\text{Hz}$, which is rather slow and thus lead to a timing problem as explained below:

10.1.4 Addressing the timing problem

Since for a single measurement we need to gain information from the stylus and the beam length, the total measurement time increases, furthermore since the measurements are performed successively there is an delay between each measurement.

The problem with this delay is that for measuring the coordinates of one point we need both informations simultaneously. We first thought of determining the delay between the request of a measurement value and its sample point in time -for both the stylus tip coordinates and the laser-beam length. If we then use in our program a (compensating) delay between both measurement-requests we could thus retrieve measurements values of a certain point in time. However, since we did not work with a real-time operating system we have no guarantee that instructions are carried out in an exact time slot, therefore this attempt is not practicable. An ideal solution would be to trigger both measurements with an external signal. Unfortunately our scanner is not capable of processing such an input so we finally decided for a simpler approach. Assuming that the stylus is moved during the (short) measurement time without a shift of direction, we can give an upper bound for the error by the following procedure:

First the coordinates of the stylus tip are checked then the length of the laser beam and afterwards the stylus position is tracked again. Then we evaluate the coordinates for both situations and if the difference between them is smaller than a given threshold (in the order of $\sim 0.2\text{mm}$) we accept the average of both measurements otherwise it is ignored. The trade-off of this concept is a further decreasing scan-rate, an additionally introduced inaccuracy, and still no confident measurement.

With this method the total point-measurement time increase from 2ms to 10ms (compared to contact operation).

10.1.5 Scanning Procedure

After the scanner is calibrated, it is ready for the digitizing process. The scanner is controlled by two foot pedals. Since we can use both the stylus tip and the laser-probe for digitizing, we assigned the left pedal for non-contact operation (being more comfortable) and the right to

retrieve points with the stylus tip (offering a higher scan-rate and for scanning transparent media).

In the non-contact mode the scanning itself is somehow similar to airbrush-painting, but instead of spraying paint the laser-spot is swept across the surface and thus sample points are acquired.

The laser probe has a limited measurement range, so the distance to the surface of the object must not be arbitrary. To give feedback of the measurement range there are three LEDs indicating when the probe is working (green), the measurement range is exceeded (red), or when the optimal working range is reached (orange). Since the scanning software keeps track of the measurement range this limited measurement range may be seen as an advantage: If during the scanning process the laser is not projected onto the object, usually the measurement range is exceeded and thus no points of the background are sampled.

Before digitizing an object the operator has to take care that the object's surface can be accessed well in all areas (with the scanner's probe). For larger objects this is not always possible. As a solution we tie every object to a coordinate system determined by three surface points.

Having this fixed coordinate system we allow the object to be turned and moved. Then any displacement is identified by these control points again and the primary scanned data can be aligned with the new position.

10.1.6 Accuracy and Performance

It is not very easy to exactly determine the precision of the modified scanner.

By the conversion to non-contact mode we have introduced several sources of errors. The achieved precision of the coaxial probe mounting has to be questioned, and the increased length of the last arm segment -due to the laser-beam- has to be taken into account as well. Considering the laser-probe alone we have to think of its measurement error, the size of the laser-spot, and the resolution of the AD-converter. Further inaccuracy is -as already mentioned above- caused by non-simultaneous measurements of the laser-probe and the digitizer arm.

To give a rough estimation for the scanner's accuracy we assume in the following a perfect coaxial mounting of the laser-probe and a precisely known offset.

For the static error we can ignore the timing problem and if we take into account that the increased arm length is rather small (compared to the overall arm length), and the error of the AD-converter can be neglected as well, we can give an lower bound for the overall error by simply summing up the errors of the laser-probe ($\sim 0.1\text{mm}$) and the digitizer arm ($\sim 0.22\text{mm}$) resulting in $\sim 0.3\text{mm}$.

If we take the timing problem into account the accuracy further decreases. Here an estimation can be given by the sum of the static error and the value of the threshold for the allowed movement during the measurement (as mentioned in the section "Addressing the timing problem").

However, in practice a measurement error smaller than $\sim 0.5\text{mm}$ has been observed.

The maximal scan-rate we measured is about 100 points/s.

10.2 Software

During this thesis a scanning software package, "SCANDESK" was developed. The software was written in Visual C++ 6.0 making use of the Visual Toolkit (VTK) [60], a powerful open source, freely available, software system for computer graphics. Efforts have been undertaken in our software to make it comfortable and easy to use. SCANDESK provides the following functionality:

- communication with the scanner (retrieving sampled points)
- visualizing the already scanned points
- decimation of the scanned data to reduce noise and redundancy
- surface reconstruction and visualization
- smoothing for a more convenient representation
- exporting the generated surface in "STL"-format so that further processing (external viewer/editor, CAM...) is possible.

10.2.1 Sampling

SCANDESK provides a scanner-setting dialog where all the parameters for the scanner can be set. This includes a simple calibration routine to estimate the offset between the laser-probe and the stylus.

Once the scanner is successfully connected, the pedal status, stylus position, direction, and laser beam-length are monitored continuously. To achieve a rapid update of the measured values, this is realized in a separate thread of the program. Only if a movement of the digitizer arm is detected, either (depending on the pressed pedal) the stylus's or laser probe's coordinates are calculated. If the measurement appears to be valid, the 3D coordinates are "reported" to the main thread of the SCANDESK. There these coordinates are successively added to the sampled point set only if a pedal is pressed and the laser-probe is used within its measurement range. The obtained point cloud is displayed for an interactive operation. Thus in addition to a fast surface representation algorithm (realized by Gaussian splatting) [60] regions of undersampling can be spotted by the operator.

10.2.2 Decimation

Decimation of the obtained sampled points prior to reconstruction is important for two reasons: First, since the computation-time for the reconstruction process strongly depends on the number of points, sampled points that do not carry significant information on the surface-topology should be singled out. Second, the noise of the dataset can be reduced by averaging. Both is of great importance especially for our system, since it offers the possibility to scan areas with varying point-densities. This is primarily an advantage with respect to automated scanning systems, however, excessive scanning can increase both redundancy and noise on the sampled data.

To examine the amount of redundancy contained in the point-set, algorithms have been developed that observe a neighborhood of a point and evaluate the local density of this

neighborhood, the variation of the estimated surface normals, or the change of other parameters like color (not used here). The more variation is detected, the more weight is assigned to the specific point. In a second step according to these weights the dataset is decimated. We implemented a simple decimation algorithm, only based on the local density of the point-set. In this algorithm a ball with a given radius around each sample point is observed and all the points contained in this ball are substituted by their centroid. Besides speeding up the reconstruction, this algorithm is mainly intended to decrease the noise of the sampled points. Thus a good (initial) guess for the radius of the ball is the scanner-accuracy. We also experimented with a slightly more sophisticated algorithm, where we did not "concentrate" the dataset contained in a ball around an arbitrary chosen point, instead we determined the point in this neighborhood with the maximum local density. This variant, however, is much slower especially when the radius (and thus the number of neighbors) increases. We observed, that the achieved improvement did not justify the increased computing time in practice.

10.2.3 Reconstruction

SCANDESK is able to compute a simplicial surface having either the scanned points or an imported point-set as input.

For this reconstruction process from 3 different algorithms are available

- a simple version of Hoppe's algorithm
- the Power-Crust and
- the Tight-Cocone algorithm

We did not re-implement any of this algorithms ourselves, instead we used the `vtkSurfaceReconstructionFilter` (Hoppe's algorithm) [60], the Windows port of the Power-Crust [42] kindly provided by Nina Amenta, and the Tight-Cocone [50] by courtesy of Tamal Dey.

The variant of Hoppe's algorithm can be used for a fast representation of the surface by selecting a large value for the sample spacing-parameter, as a trade-off the reconstructed model is poor of details. While the algorithm performs quite well on equally dense samples, it can produce surfaces with artefacts and is not competitive to the Power-Crust- or Cocone algorithm.

Power-Crust offers the best looking results but it suffers from high computation time and memory consumption, and its poor implementation -that was available to us- often caused computer-crashes. Thus in practice we prefer using the Cocone algorithm resulting in comparable good models but computation is faster and more stable. The performance difference between the Power-Crust and the Cocone algorithm is mainly due to the different algorithms used for triangulation. Thus the Power-Crust could strongly benefit from a more efficient implementation.

10.2.4 Smoothing

Often the reconstructed surface looks somehow crinkled. This distortion is caused by noise in the underlying point cloud. In most cases a more convenient representation of the scanned

object can be achieved by smoothing the generated surface mesh. In general smoothing algorithms reduce high frequency information in the geometry of the mesh. This reduces the noise, but also diminishes the amount of detail in the smoothed mesh, thus excessive smoothing should be avoided.

We used the `vtkSmoothPolyDataFilter` provided by the VTK [60]. This algorithm is a variant of (first order) Laplacian smoothing. The algorithm proceeds as follows: For each vertex v in the mesh a list of vertices is computed, which contains all vertices v' that are directly connected to the vertex v . Next, an iteration phase begins over all vertices v , where the new position v_{new} of the vertex is given by the centroid of these adjacent vertices v' .

In this algorithm, the smoothing itself is controlled by two parameters, the number of iterations and a relaxation factor β . As shown in Equation 10.1, this parameter β defines the ratio of the displacement between the original and the new position, in relation to the distance between the centroid of the to the vertex v adjacent vertices ($adj(v)$) and the original position of v .

$$v_{new} = v - \beta \left[v - \frac{1}{|adj(v)|} \sum_{v' \in adj(v)} v' \right] \quad (10.1)$$

This is one of the simplest and most common smoothing methods used for meshes. The method is inexpensive to compute but it does not guarantee an improvement in the mesh quality (as described in [63]) and the surface may shrink towards its centroid.

A better approach is presented by Mencl in [62], where a combination of alternating application of Laplacian smoothing of second order and edge swapping is reported to reduce noise successfully. However this has not yet been implemented here.

10.3 Stages of the Surface Reconstruction Process

In the following we will illustrate the stages of the reconstruction process. To verify our software we have used datasets obtained by scanning with the Line Triangulation 3D Scanner in our Laboratory [64] (briefly described in chapter 3). We have chosen this scanner as a reference system, because of its excellent accuracy we can expect a good reconstruction. In addition we wanted to show the scanner's full potential, which was not possible with the previously used algorithms.

Our object of interest is a small gypsum bust of Johann Strauss (approximately 100mm in height). by scanning the bust with the Line Triangulation Scanner we have acquired a point-cloud with $\sim 120\,000$ points that is reconstructed using Tight Cocone.

A direct reconstruction of the original scanned data gives already a good representation (see Figure 10.3).

By application of the decimation algorithm, the number of points is decimated to $\sim 58\,000$ points and the noise in the dataset is significantly reduced, whereas not much detail is lost, as demonstrated in Figure.10.4.

By finally smoothing the surface we get a very satisfying reconstruction of the original object (Figure 10.5).

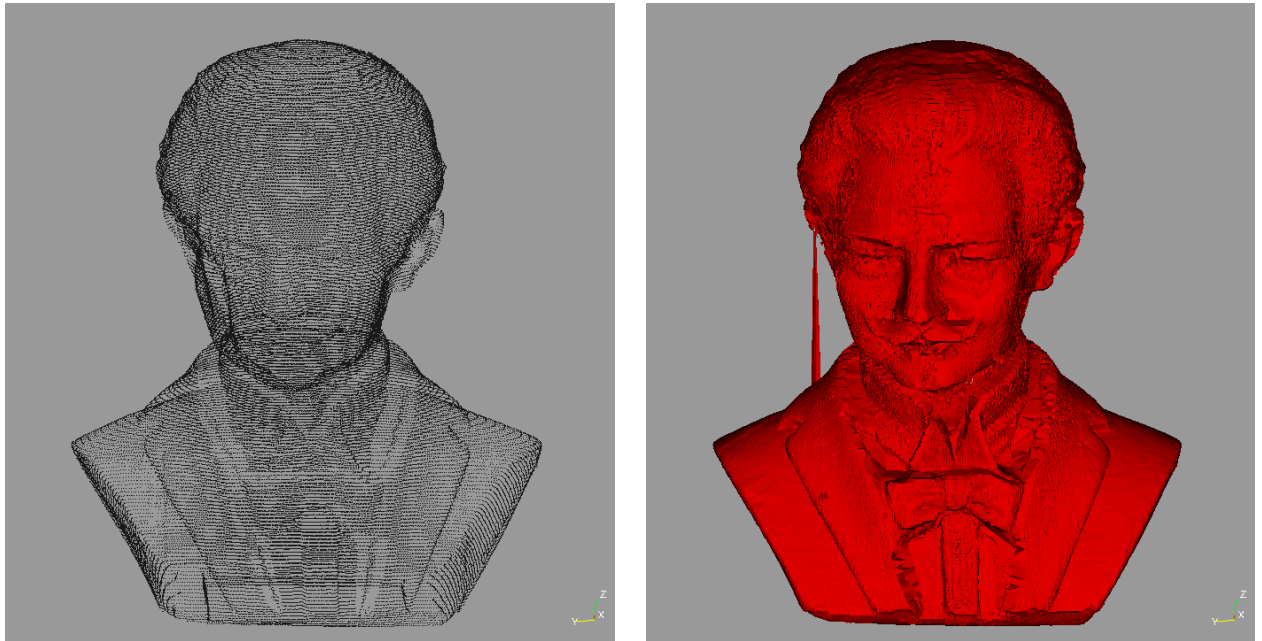


Figure 10.3: Original data: point-cloud 121.816 points (left), reconstructed surface (right). Notice the artefact on the left hand side of the head, that is due to a violation of Cocone's sampling assumptions.

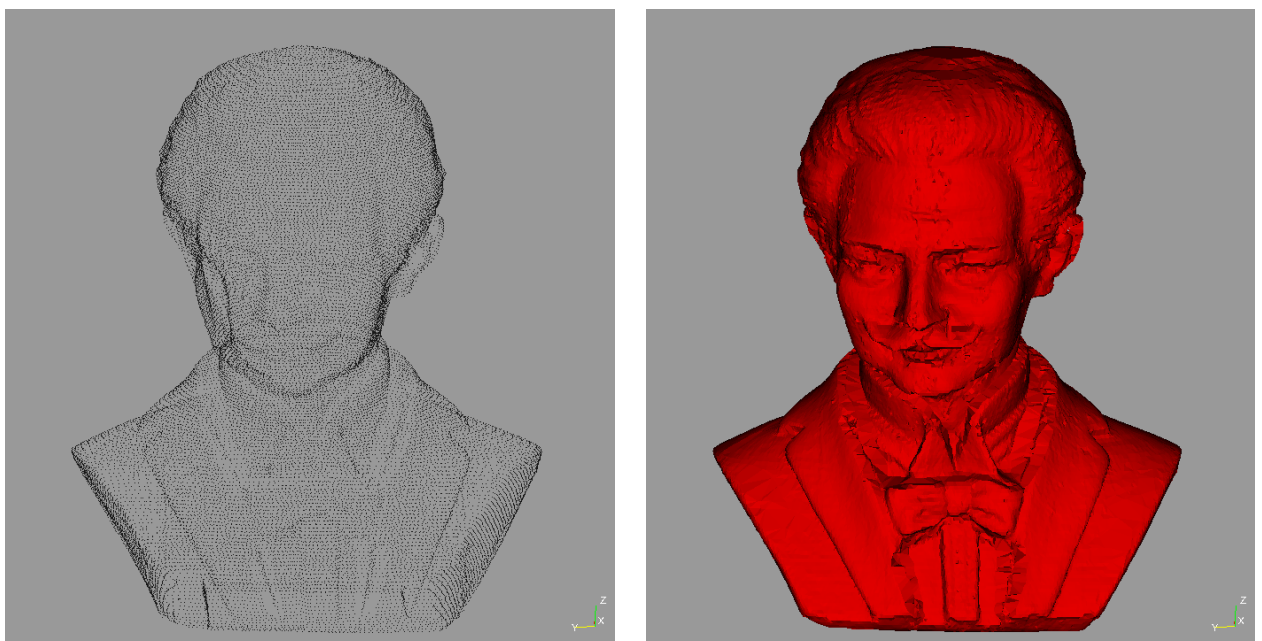


Figure 10.4: Decimated Data: point-cloud 58.148 points (left), reconstructed surface (right).



Figure 10.5: Photograph of the gypsum bust (height ~ 100 mm) (left), final reconstruction after Decimation and Smoothing (right).

10.3.1 Example of Scanned Data

To test the capabilities of our developed scanner we sampled the head of the bust and reconstructed the object. As demonstrated in Figure 10.6 our scanner works properly, but it cannot compete with the Line Triangulation Scanner scanner in terms of precision.

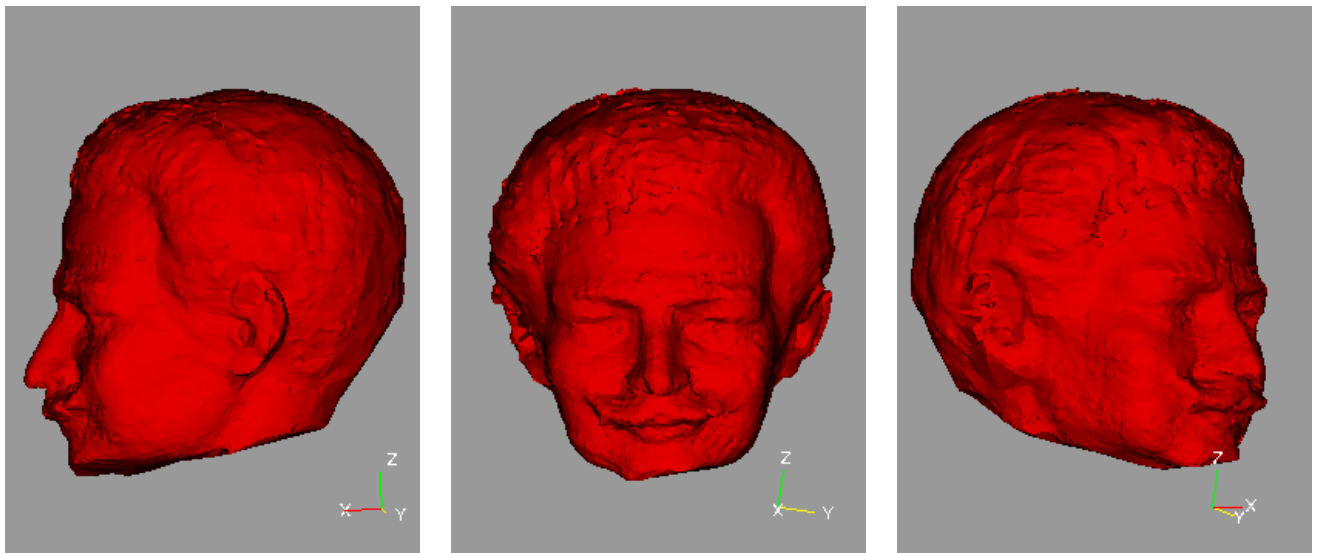


Figure 10.6: Reconstruction of the decimated data (14018 points) after smoothing.

10.4 Conclusion and Outlook

The aim of this diploma thesis was giving an introduction and overview of the topics involved in 3D Surface Scanning and to build a scanning system for practical use. Our idea was to achieve a high accuracy by performing single point measurements only and accept a low scan-rate as the trade-off.

Unfortunately the experiments exhibit that is was rather a first step than a final working solution.

We were able to set up a working system, but we are not entirely satisfied with the achieved reconstruction of the scanned objects (refer to Figure 10.6). We believe that the rather poor reconstruction is mainly due to two reasons. First the scanning data is arbitrarily dense (depending on the skill of the operator) and second for the small object that has been used as an example the accuracy of our scanning system is not sufficiently high. The low accuracy results in noisy surface meshes. The main factors for the low accuracy are the precision of the digitizer arm, and the low sampling rate of the AD-converter. Since we have no influence on the touch-probe scanner's accuracy we only see potential for improvements by the use of a faster AD converter.

Although we observed that the implemented surface reconstruction algorithms (Cocone, Power-Crust) preformed very well on other sample data sets, they ran into problems with the data obtained by our scanner. This is due to the non-uniform sampling density. Along the paths where the probe is moved, the point density is significantly higher than in other directions. These variations are miss-interpreted by the reconstruction algorithms and can result in holes of the surface mesh. To minimize these effects we successfully employed our point-decimation algorithm. On the other hand by applying the decimation algorithm, the amount of detail is reduced as well.

The hand-guided operation of our scanner offers the possibility to selectively scan certain areas, but in combination with single-point measurements it is very time demanding to achieve a dense sampling (even with the provided feedback capability of our software). Therefore we would recommend using a 2D probe (a sensor with laser stripe projection) instead of the 1D laser probe. Although the accuracy is likely to further decrease, the scanning will become much more comfortable and the scan rate will dramatically increase. Thus by averaging the (denser) point cloud statistical errors can be eliminated, which will provide a higher precision. However, a similar system is already available on the market, and our touch-probe scanner has too few degrees of freedom for such a line sensor. At least one more axis (for the direction of the laserstripe) would be necessary for a comfortable operation.

Concluding we see our scanner not as a stand-alone solution, in fact it might be better suitable for complementary scans and for hole filling, or to digitize objects with a highly symmetric shape.

Also in our software there is room for improvement. A better smoothing algorithm, and (manual or automatic) editing tools for eliminating unwanted or erroneous sample points as well as a reimplementaion of the Power-Crust algorithm (using faster Delaunay Triangulation routines) would give further impacts to the usability of the software.

Part IV
Appendix

Appendix A

Geometric Definitions and Proofs

A.1 Glossary

A.1.1 Convex Hull [52]

Preliminary Definition: A subset S of the plane is called convex if - and only if - for every pair of points \mathbf{p}, \mathbf{q} in S , the line segment \mathbf{pq} is completely contained in S .

Convex Hull Definition: The convex hull $\text{conv}(S)$ of a set S is the smallest convex set that contains S .

A.1.2 General Position

Definition [1]:

A point set S is in General Position, if:

- \mathbb{R}^2 : no 3 points of S lie on a common line and no 4 Points lie on a common circle
- \mathbb{R}^3 : no 4 points of S lie on a common plane and no 5 Points lie on a common sphere

These assumptions ensure that the Delaunay Triangulation of the point set S is unique and not degenerated.

A.1.3 Simplex

Definition [1]:

In \mathbb{R}^d any set T of size $|T| = k + 1$, with $0 \leq k \leq d$, defines a k -simplex σ_T that is the convex hull of T . The general-position assumption assures that all k -simplices (plural of “simplex”) are properly k -dimensional.

A.1.4 Voronoi Diagram [1]

Definition: For a point $\mathbf{p} \in S$, define $V_{\mathbf{p}}$, the Voronoi cell of \mathbf{p} , as the set of points $\mathbf{x} \in \mathbb{R}^3$ that the Euclidean distance between \mathbf{x} and \mathbf{p} is less than or equal to the distance between \mathbf{x} and any other point of S . $V_{\mathbf{p}} = \{\mathbf{x} \in \mathbb{R}^3 : \forall \mathbf{q} \in S - \{\mathbf{p}\}, \|\mathbf{x} - \mathbf{p}\| \leq \|\mathbf{x} - \mathbf{q}\|\}$. Each

Voronoi cell is a convex polyhedron, and the collection of all Voronoi cells, one for each point of S , defines the Voronoi Diagram of S , denoted by V (refer to Figure A.1).

The vertices of this diagram are called Voronoi vertices. In 3D a Voronoi vertex v is shared by the cells of at least four samples, which are all closest to v . The Voronoi Ball at v is the ball centered at v passing through its closest samples.

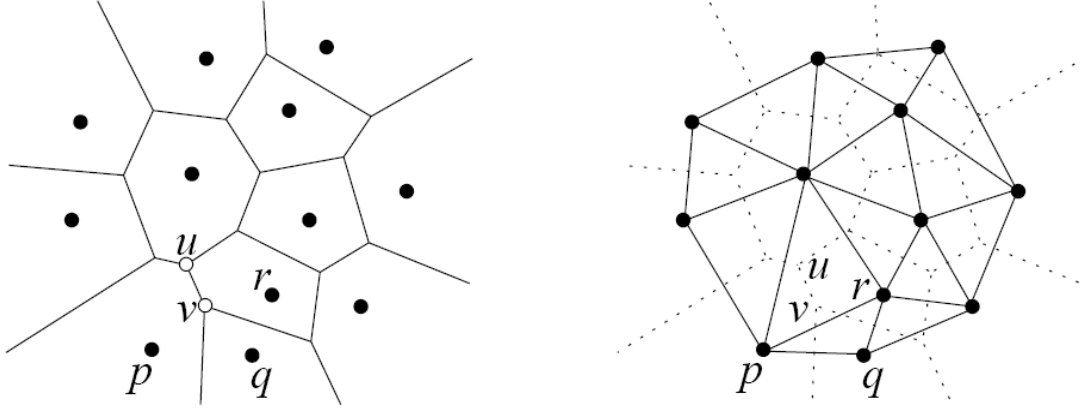


Figure A.1: Voronoi Diagram (left) and Delaunay Triangulation (right) of a point set in a plane [48].

A.1.5 Delaunay Triangulation [1]

A finite point set $S \subseteq \mathbb{R}^3$ defines a special triangulation known as the Delaunay Triangulation of S . Assuming General Position of the points, this triangulation is unique and decomposes the convex hull of S into tetrahedra. The triangulation is named after the Russian geometer Boris Delaunay (also Delone). As explained below, the Delaunay Triangulation of S is dual to another complex defined by S , known as the Voronoi Diagram.

Definition 1 [1]: The Delaunay Triangulation of a point set S in \mathbb{R}^d is the set of all ($0 \leq k \leq d$) k -simplices $\sigma_T = \text{conv}(T)$, $T \subseteq S$, $|T| = k + 1$, for which there are empty open balls b with $\partial b \cap S = T$.

A triangle is in the Delaunay Triangulation if - and only if - its circumscribing ball is empty. An edge is in the Delaunay Triangulation if - and only if - its circumscribing ball is empty. An alternative definition shows that the Delaunay Triangulation is dual to the Voronoi Diagram (refer to Figure A.1):

Definition 2 [48]: $k + 1$ points in P form a Delaunay k -simplex σ_T , if their Voronoi cells have a non-empty intersection:

$$D_P = \{\sigma_T \mid \bigcap V_{\mathbf{p}} \neq \emptyset \text{ where } \mathbf{p} \text{ is any vertex in } \sigma_T\}$$

So in \mathbb{R}^d each k -simplex σ_T in D_P is dual to a $d - k$ dimensional Voronoi face.

In \mathbb{R}^3 the Delaunay Triangulation contains four types of simplices dual to each of the four types of Voronoi faces. Delaunay vertices are dual to Voronoi cells, Delaunay edges are dual to Voronoi facets, Delaunay triangles are dual to Voronoi edges and Delaunay tetrahedrons are dual to Voronoi vertices.

A.1.6 Medial Axis [43]

The Medial Axis of an object is the closure of the set of points with more than one closest point on the surface of the object (refer to Figure A.2). Notice that a point of the Medial Axis is the center of a ball touching the surface in at least two points, but completely contained in the object. The union of all of these balls completely fills up the object. The Medial Axis Transformation is the representation of the object by this set of balls. Notice that the Medial Axis can be considered as the continuous version of the Voronoi Diagram - the set of points with more than one closest point on the input point set S gives the Voronoi Diagram.

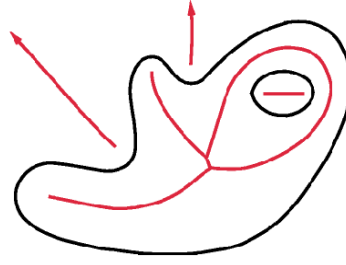


Figure A.2: Example of a two dimensional Medial Axis [43].

A.2 Proofs

We will reproduce the proofs used in the Cocone algorithm, as given in [48]:

A.2.1 Definitions

The surface Σ is a smooth manifold without boundaries embedded in \mathbb{R}^3 .

Local Feature Size We define a distance function called Local Feature Size:

$f : \Sigma \rightarrow \mathbb{R}$ where $f(\mathbf{x})$ is the least distance to any point of the Medial Axis of any $\mathbf{x} \in \Sigma$.

ε -sample We use the same sampling condition as described above in the power crust Algorithm:

A sample P of Σ is an ε -sample if each point $\mathbf{x} \in \Sigma$ has a sample point $\mathbf{p} \in P$ so that $\|\mathbf{x} - \mathbf{p}\| \leq \varepsilon f(\mathbf{x})$

In the following we will denote any ray from \mathbf{p} to a point $\mathbf{y} \in V_{\mathbf{p}}$ as $\hat{\mathbf{y}}$.

Poles:

[48]: The farthest Voronoi vertex $\mathbf{v}_{\mathbf{p}}$ in $V_{\mathbf{p}}$ is called the positive pole of \mathbf{p} . The negative pole of \mathbf{p} is the farthest point $\mathbf{p}^- \in V_{\mathbf{p}}$ from \mathbf{p} so that the two vectors from \mathbf{p} to $\mathbf{v}_{\mathbf{p}}$ and \mathbf{p}^- form an angle of more than $\frac{\pi}{2}$. We call $\hat{\mathbf{v}}_{\mathbf{p}} = \mathbf{v}_{\mathbf{p}} - \mathbf{p}$, the pole vector for \mathbf{p} . If $V_{\mathbf{p}}$ is unbounded, $\mathbf{v}_{\mathbf{p}}$

is taken at infinity, and the direction of $\widehat{\mathbf{v}}_{\mathbf{p}}$ is taken as the average of all direction given by the unbounded Voronoi edges.

A.2.2 Lemmas

Lemma (Lipschitz Continuity): $f(\mathbf{x}) \leq f(\mathbf{y}) + \|\mathbf{x} - \mathbf{y}\|$

for any two points \mathbf{x} and \mathbf{y} in Σ .

Proof: Let \mathbf{m} be a point on the Medial Axis so that $f(\mathbf{y}) = \|\mathbf{y} - \mathbf{m}\|$. By triangular inequality,

$$\|\mathbf{x} - \mathbf{m}\| \leq \|\mathbf{y} - \mathbf{m}\| + \|\mathbf{x} - \mathbf{y}\|, \text{ and}$$

$$f(\mathbf{x}) \leq \|\mathbf{x} - \mathbf{m}\| \leq f(\mathbf{y}) + \|\mathbf{x} - \mathbf{y}\| \quad \square$$

Lemma (Feature Translation): For any two points \mathbf{x} and \mathbf{y} in Σ with $\|\mathbf{x} - \mathbf{y}\| \leq \varepsilon f(\mathbf{x})$ we have

$$\|\mathbf{x} - \mathbf{y}\| \leq \frac{\varepsilon}{1-\varepsilon} f(\mathbf{y}).$$

Proof: We have

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \|\mathbf{x} - \mathbf{y}\|$$

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \varepsilon f(\mathbf{x})$$

$$f(\mathbf{x}) \leq \frac{1}{1-\varepsilon} f(\mathbf{y})$$

Plug the above inequality in $\|\mathbf{x} - \mathbf{y}\| \leq \varepsilon f(\mathbf{x})$ to obtain the result. \square

Lemma (Medial Lemma): Let \mathbf{m}_1 and \mathbf{m}_2 be the two Medial Axis points which are the centers of the two Medial Balls at p . The Voronoi cell $V_{\mathbf{p}}$ must contain \mathbf{m}_1 and \mathbf{m}_2 .

Proof: Denote the Medial Axis Ball with center \mathbf{m}_1 as B . The ball B meets the surface Σ only tangentially at more than one point, one of which is p . Thus, B is empty of any point from Σ , and in P particular. Therefore, the center \mathbf{m}_1 has \mathbf{p} as nearest point in P . By definition of the Voronoi cells, \mathbf{m}_1 must be in $V_{\mathbf{p}}$. Similar argument applies to the other Medial Axis point \mathbf{m}_2 . \square

Lemma (Normal): Let \mathbf{v} be any point in $V_{\mathbf{p}}$ with $\|\mathbf{v} - \mathbf{p}\| \geq \mu f(\mathbf{p})$, further let P be an ε sample of Σ . Then

$$\angle(\mathbf{v} - \mathbf{p}), \mathbf{n}_{\mathbf{p}} \leq \arcsin \frac{\varepsilon}{\mu(1-\varepsilon)} + \arcsin \frac{\varepsilon}{1-\varepsilon}.$$

Proof: Let \mathbf{m}_1 and \mathbf{m}_2 be the two centers of the Medial Balls touching Σ at \mathbf{p} where \mathbf{m}_1 is on the same side of Σ as v is. Both \mathbf{m}_1 and \mathbf{m}_2 are in $V_{\mathbf{p}}$ by the medial Lemma. The line joining \mathbf{m}_1 and \mathbf{p} is normal to Σ at \mathbf{p} by the definition of the Medial Balls. Similarly the line joining \mathbf{m}_2 and \mathbf{p} is also normal to Σ at \mathbf{p} . Therefore, \mathbf{m}_1 , \mathbf{p} and \mathbf{m}_2 are collinear (refer to Figure A.3). Consider the triangle \mathbf{pvm}_2 . We are interested in the angle $\angle \mathbf{m}_1 \mathbf{p} \mathbf{v}$ which is equal to $\angle(\mathbf{v} - \mathbf{p}), \mathbf{n}_{\mathbf{p}}$. From the triangle \mathbf{pvm}_2 we have

$$\angle \mathbf{m}_1 \mathbf{p} \mathbf{v} = \angle \mathbf{pvm}_2 + \angle \mathbf{vm}_2 \mathbf{p}$$

to measure the two angles on the right-hand side, drop the perpendicular \mathbf{px} from \mathbf{p} onto the segment \mathbf{vm}_2 . The line segment intersects Σ , say at \mathbf{y} , since \mathbf{m}_1 and \mathbf{m}_2 and hence \mathbf{v} and \mathbf{m}_2 lie on opposite sides of Σ . Furthermore \mathbf{y} must lie within $V_{\mathbf{p}}$ since any point in the line segment joining two points \mathbf{v} and \mathbf{m}_2 in a convex set $V_{\mathbf{p}}$ must lie within the same convex set. This means \mathbf{y} has \mathbf{p} as nearest sample point, and thus

$$\|\mathbf{x} - \mathbf{p}\| \leq \|\mathbf{y} - \mathbf{p}\| \leq \varepsilon f(\mathbf{y}) \text{ by } \varepsilon\text{-sampling condition.}$$

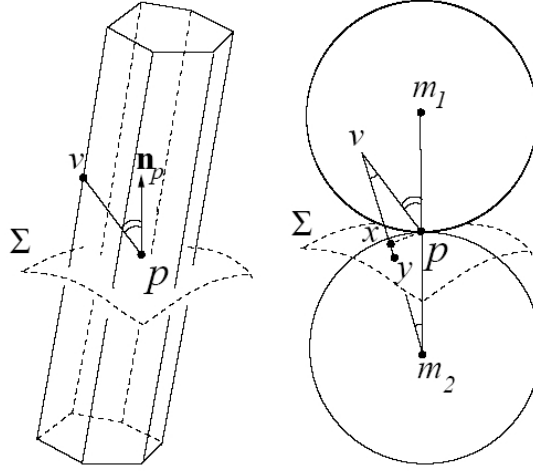


Figure A.3: Illustration for Normal Lemma [48].

Using the Feature Translation Lemma we get

$$\|\mathbf{x} - \mathbf{p}\| \leq \frac{\varepsilon}{1-\varepsilon} f(\mathbf{p})$$

We have

$$\angle \mathbf{pvm}_2 = \arcsin \frac{\|\mathbf{x} - \mathbf{p}\|}{\|\mathbf{v} - \mathbf{p}\|} \leq \arcsin \frac{\varepsilon}{\mu(1-\varepsilon)} \text{ as } \|\mathbf{v} - \mathbf{p}\| \geq \mu f(\mathbf{p}), \text{ and}$$

$$\angle \mathbf{vm}_2\mathbf{p} = \arcsin \frac{\|\mathbf{x} - \mathbf{p}\|}{\|\mathbf{m}_2 - \mathbf{p}\|} \leq \arcsin \frac{\varepsilon}{1-\varepsilon} \text{ as } \|\mathbf{m}_2 - \mathbf{p}\| \geq f(\mathbf{p}).$$

The assertion of the lemma follows immediately. \square

Lemma (Pole): The angle between the normal $\mathbf{n}_\mathbf{p}$ at \mathbf{p} and the pole vector $\widehat{\mathbf{v}}_\mathbf{p}$ satisfies $\angle \mathbf{n}_\mathbf{p}, \widehat{\mathbf{v}}_\mathbf{p} \leq 2 \arcsin \frac{\varepsilon}{1-\varepsilon}$

Proof: Since the Voronoi cell $V_\mathbf{p}$ contains the centers of the Medial Balls at \mathbf{p} , we must have $\|\mathbf{v}_\mathbf{p} - \mathbf{p}\| \geq f(\mathbf{p})$. Thus, plugging $\mu = 1$ in the statement of the Normal Lemma we obtain the result immediately. \square

Lemma (Normal Variation): Let \mathbf{x} and \mathbf{y} be any two points with $\|\mathbf{x} - \mathbf{y}\| \leq \rho f(\mathbf{x})$ for $\rho < \frac{1}{3}$. Then we have

$$\angle \mathbf{n}_\mathbf{x}, \mathbf{n}_\mathbf{y} \leq \frac{\rho}{1-3\rho}.$$

Proof: Let $\ell(t)$ denote any point on the segment \mathbf{xy} parameterized by its distance t from \mathbf{x} . Let $\mathbf{x}(t)$ be the nearest point on Σ for $\ell(t)$. The rate of change of normal $\mathbf{n}_{\mathbf{x}(t)}$ at $\mathbf{x}(t)$ is $\mathbf{n}'_t = \frac{d\mathbf{n}_{\mathbf{x}(t)}}{dt}$ as t changes. The total variation in normal between \mathbf{x} and \mathbf{y} is

$$\angle \mathbf{n}_\mathbf{x}, \mathbf{n}_\mathbf{y} \leq \int_{\mathbf{x}}^{\mathbf{y}} |\mathbf{n}'_t| dt \leq \|\mathbf{x} - \mathbf{y}\| \max_t |\mathbf{n}'_t|.$$

The surface Σ is squeezed in between two Medial Balls at $\mathbf{x}(t)$. These two balls cannot be larger than the curvature ball at $\mathbf{x}(t)$. This means Σ cannot turn faster than the larger of the two Medial Balls at $\mathbf{x}(t)$, refer to Figure A.4. We have

$$dt \geq (f(\mathbf{x}(t)) - \|\mathbf{x}(t) - \ell(t)\|) \sin d\Theta$$

As $\sin d\Theta \rightarrow d\Theta$ where $d\Theta \rightarrow 0$

$$|\mathbf{n}'_t| = \lim_{d\Theta \rightarrow 0} \left| \frac{d\Theta}{dt} \right| \leq \frac{1}{(f(\mathbf{x}(t)) - \|\mathbf{x}(t) - \ell(t)\|)} \leq \frac{1}{(f(\mathbf{x}(t)) - \rho f(\mathbf{x}))}$$

since

$$\|\mathbf{x}(t) - \ell(t)\| \leq \|\mathbf{x} - \ell(t)\| \leq \rho f(\mathbf{x}).$$

Also,

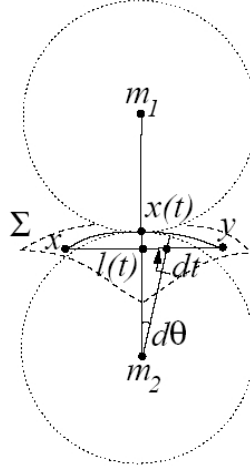


Figure A.4: Illustration for Normal Variation Lemma [48].

$$\|\mathbf{x}(t) - \ell(t)\| \leq \|\mathbf{x} - \ell(t)\| + \|\mathbf{x}(t) - \ell(t)\| \leq 2\rho f(\mathbf{x}).$$

By Lipschitz Continuity Lemma $f(\mathbf{x}(t)) \geq (1 - 2\rho)f(\mathbf{x})$. Therefore

$$|\mathbf{n}'_t| \leq \frac{1}{(1-3\rho)f(\mathbf{x})}$$

$$\angle \mathbf{n}_x, \mathbf{n}_y \leq \|\mathbf{x} - \mathbf{y}\| \frac{1}{(1-3\rho)f(\mathbf{x})} \leq \frac{\rho}{1-3\rho}. \quad \square$$

Lemma (Edge Normal): The angle $\angle \mathbf{t}_{\mathbf{p}\mathbf{q}}, \mathbf{n}_p$ is more than $\frac{\pi}{2} - \arcsin \frac{\|\mathbf{p}-\mathbf{q}\|}{2f(\mathbf{p})}$

Proof: Consider the two Medial Balls sandwiching the surface Σ at \mathbf{p} . The edge $\mathbf{p}\mathbf{q}$ cannot completely lie inside any of these balls. So the smallest angle $\mathbf{p}\mathbf{q}$ formed with \mathbf{n}_p cannot be

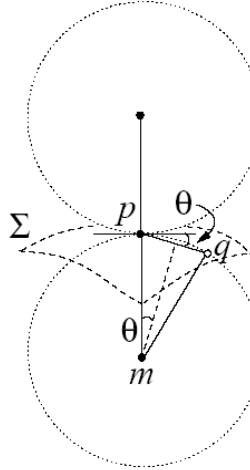


Figure A.5: Illustration for Edge Normal Lemma [48].

smaller than the angle $\mathbf{p}\mathbf{q}$ formed with \mathbf{n}_p when \mathbf{q} is on the boundary of any of these two balls. In this case it is obvious from Figure A.5 that

$$\sin \Theta = \frac{\|\mathbf{p}-\mathbf{q}\|}{2\|\mathbf{m}-\mathbf{p}\|} \leq \frac{\|\mathbf{p}-\mathbf{q}\|}{2f(\mathbf{p})}. \quad \text{Therefore,}$$

$$\angle \mathbf{t}_{\mathbf{p}\mathbf{q}}, \mathbf{n}_p = \frac{\pi}{2} - \Theta \geq \frac{\pi}{2} - \frac{\|\mathbf{p}-\mathbf{q}\|}{2f(\mathbf{p})} \quad \square$$

Appendix B

SCANDESK Manual

The SCANDESK is a software package for use with the scanning and reconstruction process. It communicates with our scanner and is also used for data processing and visualization.

B.1 Installation and Components

The following files are required:

scandesk.exe	the main program
vtkCore (version 4.2)	this package contains the VTK-DLLs used by scandesk.exe
lmscn32.dll	dynamic link library for the communication with the scanner
mem.ocx	an ADO object, providing the communication with the AD-converter
tcocone.exe	DOS command-line version of the Tight Cocone algorithm
powercrust.exe	DOS command-line of the Power-Crust algorithm
orient.exe	add-on to the Power-Crust, that takes care for a consistent orientation
simplify.exe	reduces the complexity of the Power-Crust output
lmscn.dat	ASCII- file containing control parameters of the scanner
ReconParams.dat	ASCII- file containing settings for the reconstruction algorithms

It is necessary to have all the executable files and the dynamic link libraries in the system path.

B.2 User's Guide

SCANDESK is a multiple document application, therefore several projects can be opened simultaneously. The screen is divided into four panes (refer to Figure B.1), on each of them an independent and different view of the data (scanned and reconstructed) can be chosen. The program is controlled by the buttons in the toolbars. There is the Standard Windows Toolbar (for file handling, printing and screenshots), a View Toolbar, the Scanner Toolbar, and the Data Manipulation Toolbar. Clicking on them with the left mouse button usually starts a procedure whereas right-clicking on the buttons in the Scanner- and Data Manipulation Toolbar displays a dialog for setting parameters of these procedures. In the statusbar are displayed: the actual coordinates of the probe of scanner and the number of scanned points of the Pointcloud Object .

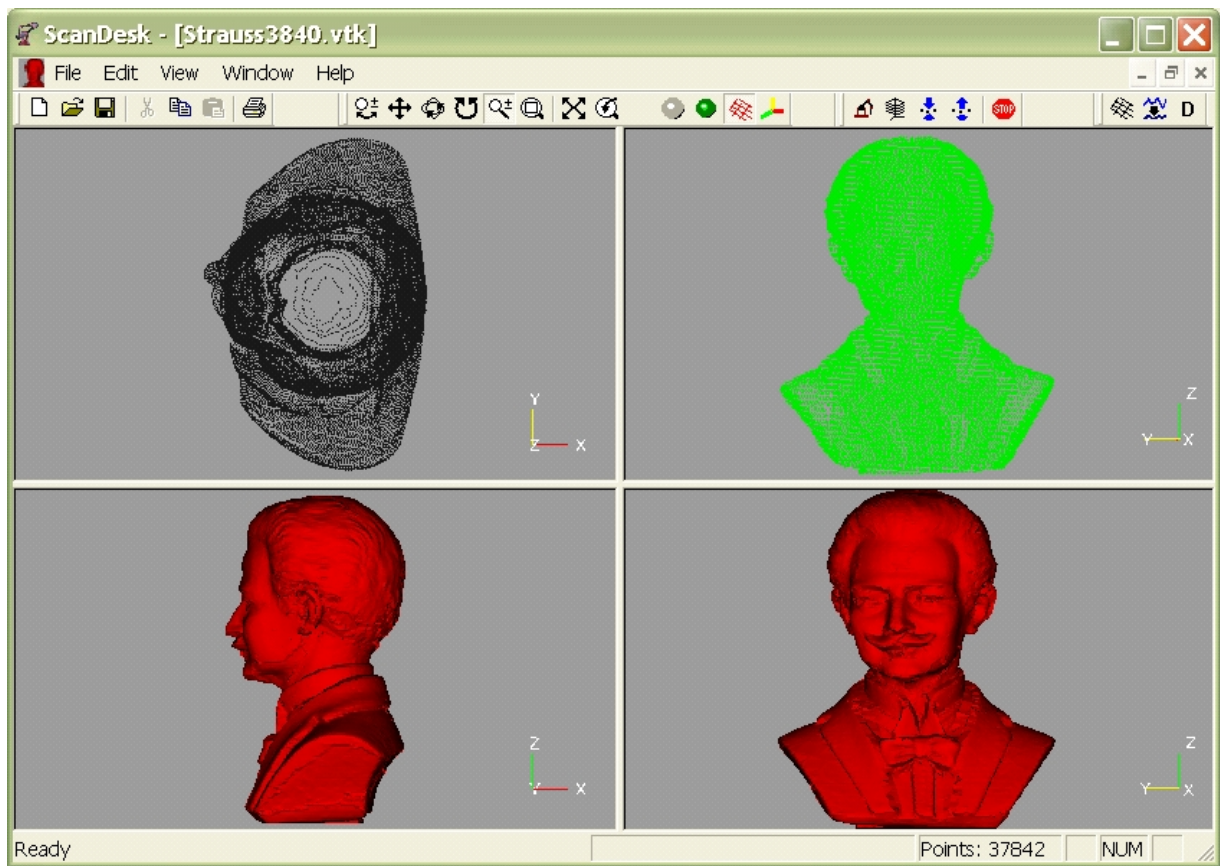


Figure B.1: SCANDESK - Screenshot.

B.2.1 Data-Objects and File-Handling

Internally SCANDESK uses three different data-objects for every project:

- The Pointcloud Object -a set of points representing the scanned data
- The Surface Object -a mesh of the (reconstructed) object
- The Vertices Object -a set of points, that contains all the vertices of the surface mesh.

The following file formats are supported:

- .asc a set of points in ASCII format: x y z
- .pts a set of points in ASCII format: p x y z
- .asd a set of points according to the dxf ASCII format
- .stl stereolithography format (ASCII or binary)
- .vtk Visual Tool Kit file-format (ASCII or binary)
- .off, .jv Geomview file-format

By opening a file either a set of points or a polygonal mesh can be loaded.

The point data is assigned to both, the Pointcloud Object and the Vertices Object, whereas gridded data is (additionally) assigned to the Surface-Object.

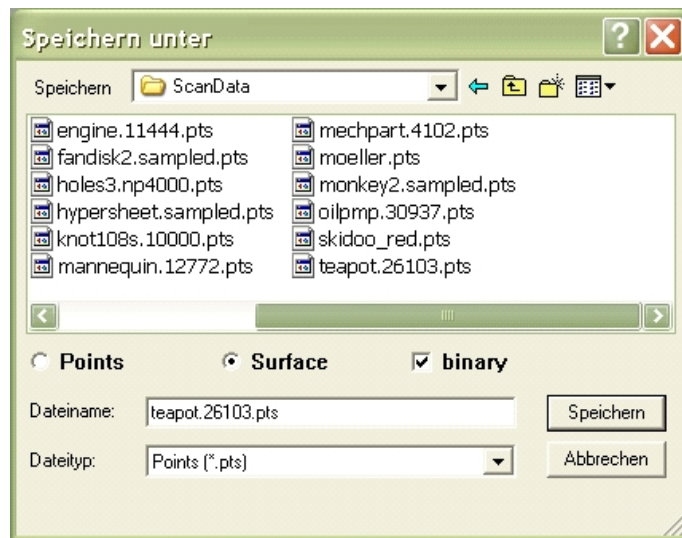
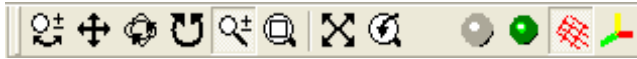






Figure B.2: Save File Dialog.

The Save-dialog has two additional controls compared to the standard dialog. There is a radio button that is used to indicate which object is to be stored (the point cloud or the surface mesh with the vertices) and second a check box "binary". If this one is checked, the data is written in binary instead of ASCII format (if supported).

B.2.2 The View Toolbar



This toolbar provides view-changing functionality (Zoom, Pan, Rotate...), and controls the visibility of the Pointcloud-Object , the Vertices-Object , the Surface-Object , and the coordinate axis .

Using the left mouse button the visibility is toggled only in the active pane, whereas the right mouse button affects all panes.

B.2.3 The Scanner Toolbar



Activate Scanner

When this button is activated the program attempts to connect the scanner using the parameters from the "lmscn.dat" file.

Right-clicking on the icon brings up the Connection Settings Dialog, by which the connection parameters (serial port, baudrate, channel...) can be set, further the scanner offsets can be entered directly or estimated by starting the calibration process.



Switch to Slice Scanning

In this operating mode points are sampled into parallel planes with a defined normal distance. (Parameters for both the plane normal and the normal distance can be set by right-clicking on this button.)



Set Reference Points

By activating this button the user is asked to mark 3 reference points with the scanner. These reference points are later used to identify the position of the scanned object.

Right-clicking brings up a dialog that provides loading (or saving) these points from (or to) a file.



Transform Data to new Position

Having the scan data tied to the reference points (by either running the "Set Reference Points"- method or loading them from a file), with this feature the scan data can be matched to the object's current position. This gives the possibility of continuing a suspended scan.

Starting this procedure the program asks for marking the three reference points again (the order in which this is done is important). Then it calculates the new position of the object and aligns the scan data to this position.

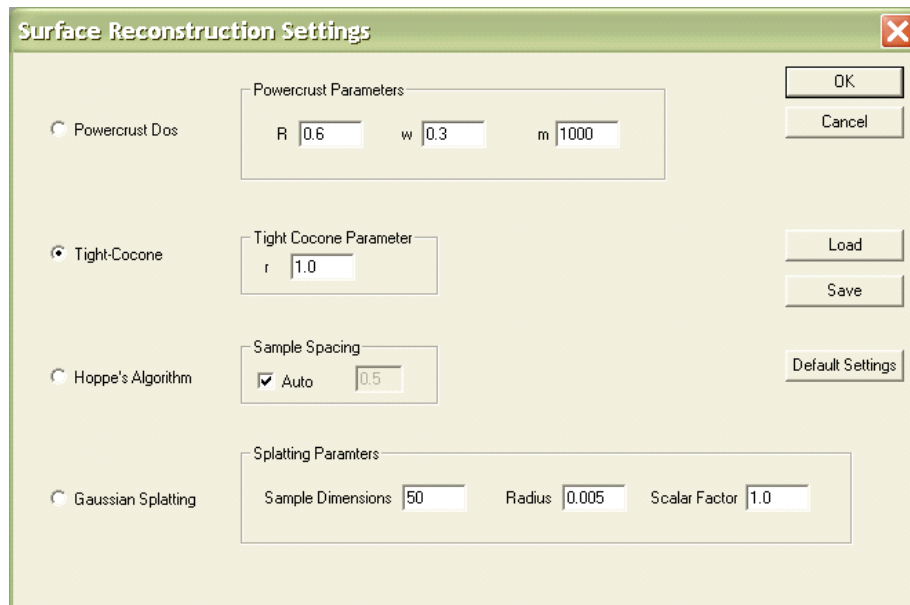


Figure B.3: Surface Reconstruction Settings Dialog.



Pause Scanner

With this button the scanner can be paused and thus computing power saved. Note that during the execution of the reconstruction-algorithms the scanner is paused automatically.

B.2.4 The Data Manipulation Toolbar



The data manipulation toolbar provides the functionality for surface reconstruction, smoothing, and point-set decimation:



Surface Reconstruction

By clicking this button the surface reconstruction is started.

By clicking on the icon with the right mouse button the settings dialog is displayed.

This dialog is used to select a reconstruction algorithm and to set the required parameters. Further this parameters can be loaded or saved from or to a file.

Power-Crust Options

- **-R** This parameter is an estimate for the sampling density constant. 1.0 or larger turns it off (default = 0.6). Used to estimate whether Voronoi cells are “well-shaped”, for handling noise and for the detection of sharp-corners. This is the r in r -sampling, that is, the minimum distance to the nearest sample on the surface, as a fraction of distance to the medial axis. When r is small, sampling is expected to be very dense, the Voronoi cells are expected to be really long and skinny and the poles of fat Voronoi cells are thrown away. On noise-free inputs with no sharp corners, the value for $-R$

can be set ≥ 1 to get (probably) a good reconstruction from a sparser input sample. Setting $-R$ small might help to get good reconstructions from dense but noisy samples.

- $-w$ is $\cos(\pi - \alpha)$, where α is the angle between “deep” intersecting balls (default = 0.3). Some poles might fail to be labeled by the regular algorithm. This parameter is passed on a second-pass clean-up function for trying to label unlabeled poles. The $-w$ value should be made smaller when there are a lot of messages about ‘unlabeled pole’.
- $-m$ multiplier. The first thing the algorithm does is to multiply all the floating point numbers in the input by this multiplier and round them into integers.

Tight-Cocone Options

- $-r$ The larger the value of this parameter, the stricter Tight-Cocone gets in choosing triangles (default = 1.0).

Hoppe’s Algorithm Options

With the Sample Spacing parameter the spacing of the 3D sampling grid is set. If it is not set, a reasonable guess will be made. This parameter controls the level of detail and thus the computation time. The larger the value of this parameter, the faster is the reconstruction.

Gaussian Splatting Options [60]

The method of Gaussian splatting injects the input points to a sample grid and computes for each of these voxels a scalar value according to a Gaussian distribution:

$$f(x) = \text{ScaleFactor} * \exp(\text{ExponentFactor} * (r/\text{Radius})^2)$$

where p is a given point, x is the current voxel sample point, and r is the distance $|x-p|$.

A surface of this values is extracted by a contour tracing algorithm.

The parameters that can be set are the number of voxels (per dimension), the Radius and the ScaleFactor.



Surface Smoothing

By pressing this button Laplacian smoothing is applied to the Surface Object.

This filter can be toggled on and off, by left-clicking on the icon; the Settings dialog is displayed by right-clicking.

The parameters for the smoothing algorithm are:

- The Number of Iterations: This is the number of passes the smoothing is performed over the entire surface mesh.
- The Relaxation Factor, determines how strong a surface vertex is moved towards the centroid of its adjacent vertices (in a single pass).

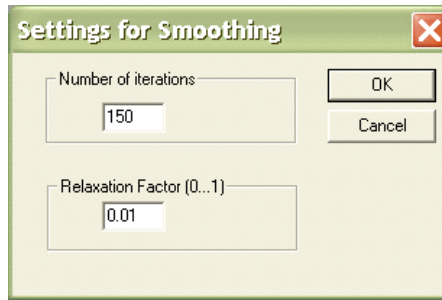


Figure B.4: Smoothing Settings Dialog.

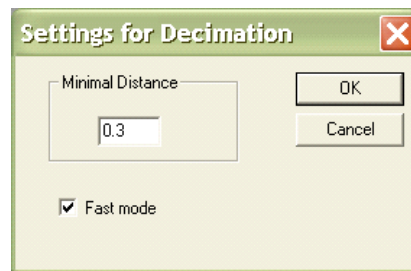


Figure B.5: Decimation Settings Dialog.

D Point Decimation

The Point Decimation filter reduces the number of points in the Pointcloud Object. The decimation filter can be toggled on and off, by left-clicking on the icon; the Settings dialog is displayed by right-clicking. The following parameters can be set

- Minimal Distance: this is a threshold for the nearest distance (between every two points) in the Pointcloud Object
- Fast mode: By deactivating the checkbox a more sophisticated decimation algorithm can be selected. This algorithm achieves a slightly better decimation, but takes much more time for the computation.

Acknowledgements

Most of my work is based on new sources which I would not have had noticed without the internet. It also provided me with useful information helping me in other aspects of my work. I just like to mention a few of the most important websites:

<http://citeseer.nj.nec.com/> for their excellent scientific library, www.codeguru.com for leading me a safe path through the dangers of programming C++, and the technical university of Munich for their comprehensive English-Dictionary <http://dict.leo.org>.

I would like to thank Tamal Dey for his excellent work and for supporting me with his surface reconstruction algorithm Cocone, Nina Amenta for making available her outstanding Power-Crust algorithm and Hugues Hoppe for providing sample data sets that were additionally used to verify our software.

I am also deeply grateful to my parents, who enabled me to continue my studies in the way I intended to.

Bibliography

- [1] Herbert Edelsbrunner, Ernst Peter Mücke; *Three-dimensional alpha shapes*, ACM Trans. Graph., 13, (1994), pp. 43–72.
- [2] Kaspar Fischer; *Introduction to alpha shapes*, <http://www.inf.ethz.ch/personal/fischerk/pubs/as.pdf>, (2000)
- [3] H. Edelsbrunner, D. G. Kirkpatrick, R. Seidel; *On the shape of points in the plane*, IEEE Transactions on Information Theory, IT-29(4), (1983), pp. 551-559
- [4] H. Edelsbrunner, E. P. Mücke; *Simulation of simplicity: A technique to cope with degeneratcases in geometric algorithms*, ACM Trans Graph., 9(1), (1990), pp. 66-104
- [5] <http://www.geomagic.com/products/>
- [6] Yves Salvadé; *Distance Measurement by Multiple-Wavelength Interferometry*, Doctoral Thesis, University of Neuchâtel, (1999)
- [7] J. Trautner, K. Walcher, R. Danielian, G. Leuchs, B. Bodermann, H. R. Telle; *Multiple-wavelength interferometry for absolute distance measurement and three-dimensional imaging*, Technisches Messen, 6, (2000), pp. 406-409
- [8] R. Dändliker, R. Thalmann, D. Prongué; *Two-wavelength laser interferometry using superheterodyne detection*, Opt. Lett., 13, (1988), pp. 339-341
- [9] James C. Wyant; *White Light Interferometry*, http://www.optics.arizona.edu/jcwyant/pdf/Meeting_papers/WhiteLightInterferometry.pdf, (2002)
- [10] G. Häusler; *„Kohärenzradar“ – ein optischer 3D-Sensor mit einer Genauigkeit von 1 μ m*, VDI, (1999)
- [11] G. Ammon, P. Andretzky, S. Blossey, G. Bohn, P. Ettl, H. Habermeier, B. Harand, G. Häusler; *KORAD - new modifications of white light interferometry for large object shape acquisition*, Proc. of the EOS Topical Meeting on Optoelectronics Distance Measurements and Applications, (1997)
- [12] P. Andretzky, M. Lindner, J. Neumann, G. Bohn, M. Schmidt, G. Häusler; *Modifications of the coherence radar for in vivo profilometry in dermatology*, Proc. of SPIE, (1998)
- [13] Robert Lange; *3D Time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology*, Doctoral Thesis, University of Siegen, (2000)

- [14] Brian Curless, Marc Levoy; *Better optical triangulation through spacetime analysis*, 5th International Conference on Computer Vision, Boston, Massachusetts, (1995)
- [15] Brian Lee Curless; *New Methods for Surface Reconstruction from Range Images*, Doctoral Thesis, Stanford University, (1997)
- [16] David Strawser, Jon Cellini; *Training Manual for Ball State University Cyberware Model 15, 3-D Laser Scanner*,
<http://www.bsu.edu/classes/flowers2/training/m15/m15man.htm>
- [17] Hermann Slevogt; *Technische Optik*, Sammlung Göschel, Band 9002, (1974)
- [18] Rudolf Schwarte; *Principles of 3-D Imaging Techniques*, Handbook of Computer Vision and Applications, B. Jähne, H. Haussecker, P. Geissler (Eds.), Academic Press, (1999)
- [19] R. Schwarte, H. Heinol, Z. Xu, J. Li, B. Buxbaum; *Pseudo/Noise (PN)-Laser Radar without Scanner for Extremely Fast 3D-Imaging and -Navigation*, MIOP 97 - Microwaves and Optonics, (1997), pp. 168-176
- [20] <http://www.wolfbeck.com/>
- [21] Fang-Jung Shiou, Ming-Jiunn Chen; *Intermittent Process Measurement of a Freeform Surface Profile with a Circular Triangulation Laser Probe on a Machining Centre*, The International Journal of Advanced Manufacturing Technology, Springer-Verlag London Ltd, pp.365-376, (2003)
- [22] <http://www.riegl.co.at/>
- [23] <http://www.polhemus.com/>
- [24] <http://www.leica-geosystems.com/ims/product/accessories/laser-tracker/>
- [25] Elvi Räisänen-Ruotsalainen, Timo Rahkonen, Juha Kostamovaara;
A High Resolution Time-to-Digital Converter Based on Time-to-Voltage Interpolation,
<http://www.imec.be/esscirc/papers-97/48.pdf>
- [26] Horst G. Heinol; *Untersuchung und Entwicklung von modulationslaufzeitbasierten 3D-Sichtsystemen*, Doctoral Thesis, University of Siegen, (2001)
- [27] Yalin Xiong, Steve Shafer; *Depth from Focusing and Defocusing*, Proc. of the Intl. Conf. of Comp. Vision and Pat. Recogn., (1993), pp. 68–73
- [28] J. Meel; *Spread spectrum Introduction*, Sirius Communications, (1999)
- [29] Z. Xu; *Investigation of 3D-Imaging Systems Based on Modulated Light and Optical RF-Interferometry (ORFI) - ZESS-Forschungsberichte*, Shaker Verlag Aachen, ISBN 3-8265-6736-6, (1999).
- [30] E. Hecht, A.Zajac; *Optics*, Addison-Wesley, (1974).
- [31] Thomas Luhmann; *Nahbereichsphotogrammetrie: Grundlagen Methoden und Anwendungen*, Herbert Wichmann Verlag, (2000).

- [32] <http://www.vexcel.com/>
- [33] Marc Levoy; *The Digital Michelangelo Project*, <http://graphics.stanford.edu/projects/mich/>, (1998-1999)
- [34] Szymon Marek Rusinkiewicz; *Realtime Acquisition and rendering of large 3D models*, Doctoral Thesis, Stanford University, (2001)
- [35] Y. Chen, G. Medioni; *Object Modeling by Registration of Multiple Range Images*, Proc. IEEE Conf. on Robotics and Automation, (1991).
- [36] K. Pulli; *Multiview Registration for Large Data Sets*, Proc. 3DIM, (1999)
- [37] Asla. M. Sá, Paulo Cezar P. Carvalho, Luiz Velho; *(b,s)-BCSL: Structured light color boundary coding for 3D photography*, VMV 2002 Proceedings, (2002).
- [38] William E. Lorensen, Harvey E. Cline; *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*, Computer Graphics (Proceedings of SIGGRAPH '87), Vol. 21(4), (1987), pp. 163-169
- [39] Hugues Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle; *Surface reconstruction from unorganized points*, Computer Graphics (Proceedings of SIGGRAPH '92), pp. 71-78, (1992)
- [40] Hugues Hoppe; *Surface reconstruction from unorganized points*, Doctoral Thesis, University of Washington, (1994)
- [41] Sergei Azernikov; *Surface reconstruction from unorganized points*, <http://www.ee.technion.ac.il/~ayellet/VisAnim/Students-01/hoppe92%5B1%5D.ppt>
- [42] Nina Amenta, Sunghee Choi, Ravi Kolluri; *The Power Crust*, Sixth ACM Symposium on Solid Modeling and Applications, (2001), pp. 249-260
- [43] Nina Amenta; <http://www.cs.utexas.edu/users/amenta/>
- [44] Nina Amenta, Marshall Bern, Manolis Kamvyselis; *A new Voronoi-based surface reconstruction algorithm*, Siggraph, (1998), pp. 415-421.
- [45] Nina Amenta, Marshall Bern; *Surface reconstruction by Voronoi filtering*, Discrete and Computational Geometry, 22, (1999), pp. 481-504
- [46] Sunghee Choi, Nina Amenta; *Delaunay triangulation programs on surface data*, The 13th ACM-SIAM Symposium on Discrete Algorithms, (2002), pp. 135-136
- [47] N. Amenta, S. Choi, T. K. Dey, N. Leekha; *A simple algorithm for homeomorphic surface reconstruction*, Proc. 16th ACM Symposium on Computational Geometry, (2000), pp 213-222.
- [48] Tamal Dey; *Lecture Notes-Course of shapebased Modelling*, <http://www.cis.ohio-state.edu/%7Etamaldey/course/788/>

- [49] T. K. Dey, J. Giesen, N. Leekha, R. Wenger; *Detecting boundaries for surface reconstruction using co-cones*, Intl. J. Computer Graphics & CAD/CAM, vol. 16, (2001), pp. 141–159.
- [50] T. K. Dey, S. Goswami; *Tight Cocone; A water tight surface reconstructor*, Technical Report OSU-CISRC-12/02-TR31, The Ohio State University, (2002).
- [51] T. K. Dey, J. Giesen, J. Hudson; *Delaunay based shape reconstruction from large data*, Proc. IEEE Symposium in Parallel and Large Data Visualization and Graphics (PVG2001), (2001), pp. 19–27.
- [52] <http://s01.middlebury.edu/CX330A/ch1.html>
- [53] <http://perso.club-internet.fr/dpo/numerisation3d/>
- [54] <http://www.simple3d.com/>
- [55] <http://www.remtek.com/>
- [56] <http://www.geo-instrument.com/>
- [57] E. Lombardo, M. Martorelli, V. Nigrelli; *Non-Contact Roughness Measurement in Rapid Prototypes by Conoscopic Holography*, XII ADM International Conference, (2001)
- [58] <http://www.acu-gage.com/conovision.htm>
- [59] <http://www.optimet.com/>
- [60] <http://www.vtk.org/>
- [61] <http://www.optoncdt.com/>
- [62] Robert Mencl; *Reconstruction of Surfaces from Unorganized Three-Dimensional Point Clouds*, Doctoral Thesis, University of Dortmund, (2001)
- [63] Nurhan Cetin; *A local mesh optimization algorithm for the generation of well-graded meshes*, M.ScThesis, ETH Zuerich, (2000)
- [64] Stefan Pelech; *Ein Experiment zur 3D Oberflächenvermessung nach dem Lichtschnittverfahren*, Diploma Thesis, Vienna University of Technology, (2001)
- [65] <http://www.auto.tuwien.ac.at/~blieb/woop/fibsearc.html>
- [66] <http://www.steinbichler.com/>
- [67] <http://www.cyberware.com/>
- [68] <http://www.gtcocalcomp.com/>
- [69] http://phot.epfl.ch/workshop/wks96/art_3_1.html
- [70] <http://graphics.stanford.edu/data/3Dscanrep/>
- [71] <http://www.siggraph.org/education/materials/renderman/BMRT/docs/bmrt.html>

- [72] H. Schmid; *Eine allgemeine analytische Lösung für die Aufgabe der Photogrammetrie*, Bildmessung und Luftbildwesen, (1958), pp.103-113
- [73] <http://cm.bell-labs.com/netlib/voronoi/hull.html>
- [74] <http://www.immersion.com/>
- [75] <http://www.bmc-messsysteme.de/>