Inaugural – Dissertation

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich-Mathematischen Gesamtfakultät

der

Ruprecht – Karls – Universität

Heidelberg

Vorgelegt von: Andreas Beyer (M.Sc.)
Geboren in: Mainz
Tag der mündlichen Prüfung: 27. Juli 2016

Data Fusion of Surface Meshes and Volumetric Representations

Advisors:
Prof. Dr. Hans Georg Bock
Prof. Dr. Christoph Schnörr

Abstract

The term *Data Fusion* refers to integrating knowledge from at least two independent sources of information such that the result is more than merely the sum of all inputs. In our project, the knowledge about a given specimen comprises its acquisitions from optical 3D scans and Computed Tomography with a special focus on limited-angle artifacts. In industrial quality inspection those imaging techniques are commonly used for non-destructive testing. Additional sources of information are digital descriptions for manufacturing, or tactile measurements of the specimen. Hence, we have several representations comprising the object as a whole, each with certain shortcomings and unique insights. We strive for combining all their strengths and compensating their weaknesses in order to create an enhanced representation of the acquired object. To achieve this, the identification of correspondences in the representations is the first task. We extract a subset with prominent exterior features from each input because all acquisitions include these features. To this end, regional queries from random seeds on an enclosing hull are employed. Subsequently, the relative orientation of the original data sets is calculated based on their subsets, as those comprise the—potentially defective—areas of overlap. We consider global features such as principal components and barycenters for the alignment, since in this specific case classical point-to-point comparisons are prone to error. Our alignment scheme outperforms traditional approaches and can even be enhanced by considering *limited-angle* artifacts in the reconstruction process of Computed Tomography. An analysis of local gradients in the resulting volumetric representation allows to distinguish between reliable observations and defects. Lastly, tactile measurements are extremely accurate but lack a suitable 3D representation. Thus, we also present an approach for converting them in a 3D surface suiting our work flow. As a result, the respective inputs are now aligned with each other, indicate the quality of the included information, and are in compatible format to be combined in a subsequent step. The data fusion result permits more accurate metrological tasks and increases the precision of detecting flaws in production or indications of wear-out. The final step of combining the data sets is briefly presented here along with the resulting augmented representation, but in its entirety and details subject to another PhD thesis within our joint project.

Zusammenfassung

Der Begriff *Data Fusion* bezeichnet die Zusammenführung von Daten aus zwei oder mehr unabhängigen Quellen. In ihrer Verbindung ergänzen sich die Daten statt sich zu überlagern. In unserem Projekt werden hierbei die Aufnahmen eines Werkstücks behandelt, wie sie ein optischer 3D-Scanner und ein Computertomograph generieren; besonderes Augenmerk liegt bei letzterem auf *limited-angle* Artefakten. In der industriellen Qualitätskontrolle kommen beide bildgebende Verfahren im Rahmen der zerstörungsfreien Werkstoffprüfung zum Einsatz. Zusätzlich existieren meist Konstruktionsbeschreibungen wie z.B. CAD-Daten, oder es liegen taktile Messungen vor. Somit stehen mehrere Beschreibungen des kompletten Objekts zur Verfügung, wobei darin jeweils unterschiedliche Aspekte hervorgehoben oder vernachlässigt werden. Unser Ziel besteht darin, den Mehrwert der einzelnen Eingaben zu bündeln und eine detailreichere Gesamtrepräsentation des Werkstücks zu erstellen. Hierzu müssen lokale Übereinstimmungen in den Datensätzen identifiziert werden. Da markante äußere Merkmale in allen Darstellungen gleichermaßen präsent sind, bietet sich deren Extraktion an. Es eignen sich Distanzanfragen an das Objekt, von Zufallspunkten, die auf einer das Objekt umgebenden Hülle verteilt sind. Im Folgenden muss die räumliche Lage der Objektrepräsenationen zueinander geklärt werden, dies erfolgt auf Basis der – eventuell fehlerbehafteten – Extraktionen. Eine Lagebestimmung mittels Hauptachsenanalyse und Schwerpunkt liefert in unserem Ansatz deutlich bessere Ergnbinsse fehleranfällige punktbasierter Verfahren. Zur Fehlerbestimmung computertomographisch erstellter Volumendaten wird deren Gradientenfeld untersucht, um hiervon Aussagen über die lokale Verlässlichkeit der Repräsentation abzuleiten. Ein Konvertierungsprozess der es ermöglicht hochgenaue taktile Messungen in eine 3D-Repräsentation zu überführt und dadurch für *Data Fusion* nutzbar macht, wird ebenfalls im Rahmen dieser Arbeit beschrieben. Folglich liegen nun alle Datensätze in einheitlicher Form mit bekannter Orientierung und Fehleranalyse vor. Sie können somit zusammengeführt werden, um präzisere Messungen und Verschleißprüfungen zu ermöglichen. Die resultierende, detailiertere Darstellung des Werkstücks wird vorgestellt, sowie die zur Zusammenführung entwickelte Methode. Letztere ist jedoch Gegenstand einer anderen Doktorarbeit im Rahmen unseres Projekts.

Danksagung

An erster Stelle möchte ich allen Personen meinen Dank aussprechen, die mich in den letzten Jahren untersützt haben. Mein besonderer Dank gilt meinem Doktorvater Prof. Hans Georg Bock, ohne dessen Einsatz das *ILATO Projekt* und damit verbunden meine Finanzierung, nicht zustande gekommen wären. Er ermöglichte die Arbeit an diesem spannenden Thema, räumte mir große Gestaltungsmöglichkeiten in meiner Forschung ein und gab in konstruktiven Besprechungen wertvollen Rat. Nicht weniger dankbar bin ich meiner Mentorin Dr. Susanne Krömker, die mir stets mit sanftem Druck und äußerst geduldig die nächsten Schritte auf den richtigen Weg gewiesen hat. Ihre unermütliche Bereitschaft mit mir jede Publikation und Präsentation zu redigieren, Plücker-Koordinaten aufzudröseln oder andere mathematische Fragestellungen zu erörtern, hatte auf den Verlauf meiner Arbeit ebenso positiven Einfluß, wie ihr Gespür dafür, vor welchem Raum ich zu welcher Zeit ganz zufällig bestimmte Personen treffen sollte.

Ich danke allen meinen Kollegen und Freunden des 5. Stockwerks Prof. Filip Sadlo, Dr. Hubert Mara, Julia Portl, Bastian Rieck, Bartosz Bogacz sowie Boyan Zheng, die immer ansprechbar waren für fachlichen Rat, Diskussionen und Korrekturen. All jenen danke ich auch für die gute gemeinsame Zeit mit viel Spaß, Schokolade, Karaoke, Nerf-Duellen und Drachenboot-Action. Außerdem danke ich allen Studenten, die als Hiwi, mit einem Praktikum oder, wie Tristan Klemm, durch eine Abschlußarbeit einen Beitrag zum *ILATO Projekt* leisteten.

Mein Dank gilt ebenfalls Prof. Christoph Schnörr für äußerst hilfreiche Besprechungen, in denen er mir unter anderem aufzeigte, welche weiteren relevanten CT-Daten vom Projektpartner anzufordern sind. Großen Dank möchte ich an die Projektpartner richten. Ich danke Dr. Johannes Schlöder für die Mitwirkung am Projektantrag und Hilfestellungen bei Verständnisfragen. Ein besonderer Dank geht in die Schweiz an Prof. Urs Sennhauser, Alexander Flisch, Dr. Philipp Schütz und insbesondere Empa Doktorand Liu Yu für die fantastische Arbeitsatmosphäre, eine sehr motivierende, kurzweilige und erfolgreiche Zusammenarbeit. Ich danke der *HGS MathComp* und Dr. Michael Winckler dafür, mir einen mehrwöchiger Forschungsaufenthalt in der Schweiz zu ermöglichten und auch für die Bereitstellung des 3D-Scanners und die Finanzierung von Scan-HiWis wie Sonja Speck. Außerdem danke ich für die Unterstützung dabei, das Drachenbootrennen als jährliches HGS-Event zu etablieren. Nicht unerwähnt lassen will ich die MitarbeiterInnen der Sekretariate und Verwaltung, für die immer unkomplizierte und zügige Bearbeitung meiner Anfragen und Bitten.

Ganz besonders möchte ich mich bei meinen Eltern und Geschwistern bedanken, die stets großes Vertrauen in mich und meine Fähigkeiten hatten und mir meine Studium ermöglichten. Von Herzen danke ich meiner Frau Melanie und meinem Sohn Philipp die oft auf mich verzichten mussten und mir alle Kraft und Rückhalt gaben die ich brauchte.

Contents

Chapter 1

Introduction and Problem Statement

The following is a brief introduction about the scope of this work and the associated project. The sources of input data, their peculiarities and relevance are highlighted as well as the key problem in merging them into a single representation. The second chapter presents several sample objects and their characteristics, introduces basic principles and integrated components. Preliminary work and supporting algorithms are presented in the third chapter. The main contributions and essential implementations are introduced in chapter four and evaluated in chapter five. Finally chapter six discusses the results, provides an outlook and starting points for future work.

1.1 Motivation

In the field of industrial quality inspection tactile measurements are more and more superseded by non-destructive measurement methods. Approaches such as structured light 3D scanners [SM92] or laser-based distance measurements generate highly accurate representations of the test object's surface. During this process an irregular triangular mesh of the object's surface with a spatial resolution up to $\sim 10 \mu m$ is created. The time needed to acquire this information varies between seconds and minutes but does not provide interior structures. As an alternative method to discover interior damages or indication of wear-out, Computed Tomography (CT) scans create volumetric data as a digital representation of internal structures with a resolution of $\sim 75 \mu m$. The time needed to complete a CT scan is significantly higher than its optical counterpart and can take hours or days depending on size and density of specimen. All those models can be matched against existing digital descriptions used during production such as CAD

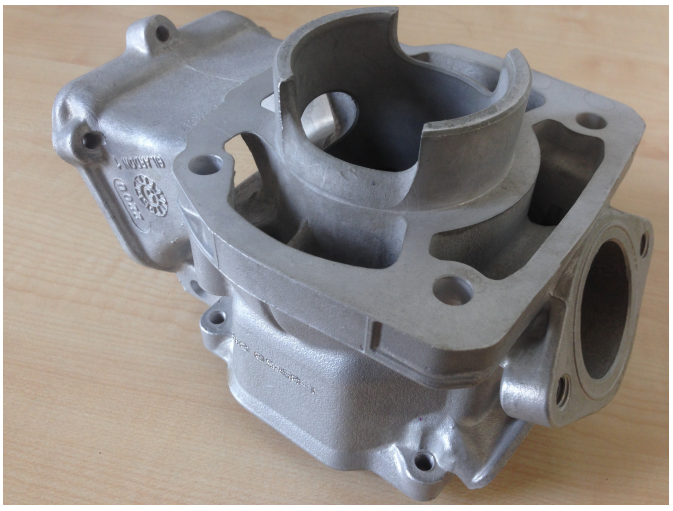
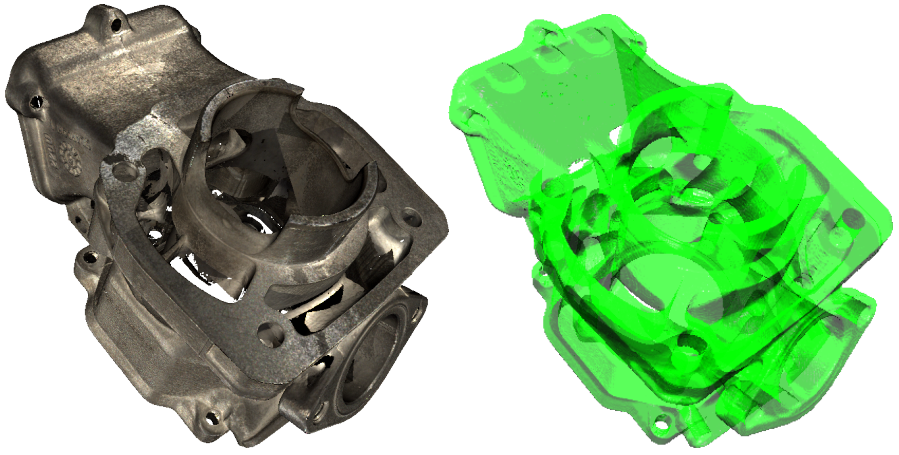


FIGURE 1.1: Real-world example for specimens in the industrial domain.

files. For non-destructive testing, Computed Tomography, as well as optical scans, are commonly used for quality inspection of industrial parts.

A typical example of a real-world industrial object is shown in Figure 1.1. It has been acquired via optical 3D imaging (Figure 1.2A) and Computed Tomography. Figure 1.2B shows an isosurface at the interface of the objects' material and the surrounding air. Both imaging techniques have their own strengths and weaknesses. A prerequisite for Data fusion is to align those representations, shown in Figure 1.3, which is in principle feasible through standard approaches. Unfortunately, due to the characteristics of the acquired data sets, alignment algorithms are prone to introduce errors, which we address in the following.



(A) Mesh of optical scan $\mathcal{M}_{\text{Opt}}$ (B) Isosurface mesh $\mathcal{M}_{\text{CT}}$

FIGURE 1.2: Industrial example of a cylinder cast from different acquisitions.

We use the notation $\mathcal{M}_{\mathrm{CT}}$ for any isosurface mesh generated from CT data and the notation $\mathcal{M}_{\mathrm{Opt}}$ for a surface acquired with an optical 3D scanner.

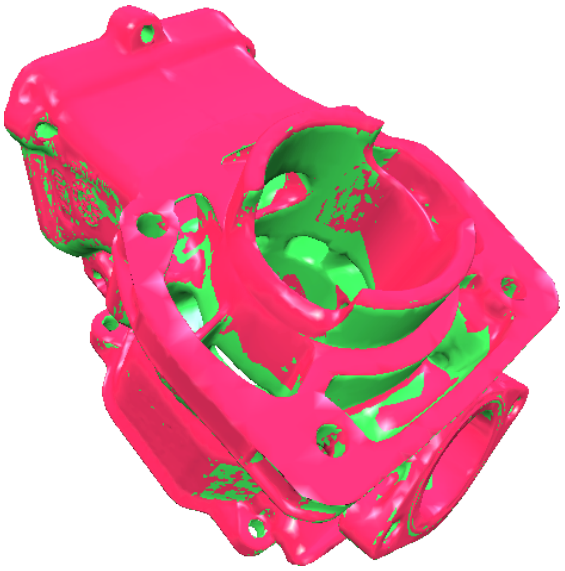


FIGURE 1.3: Super-imposed $\mathcal{M}_{\mathrm{Opt}}$ (red) and $\mathcal{M}_{\mathrm{CT}}$ (green).

1.1.1 Project ILATO

The ILATO project focuses on *Improving Limited Angle computed Tomography by Optical data integration* in order to enhance image quality and shorten acquisition times in X-ray based industrial quality inspection. Limited angle computed tomography is indicated whenever specimen dimensions exceed cone beam limits or the object is impenetrable from certain angles. Thus, acquiring only a subset of a full circle CT scan poses problems for reconstruction algorithms due to incomplete data which introduces blurred edges and other artifacts. To support volumetric data reconstruction algorithm a surface mesh of the object obtained via optical scanning acts as a mask defining boundaries of the reconstructed image. The registration of optically acquired surfaces with data acquired from computed tomography is our current challenge. This work presents our setup, the methods applied and discusses the problems arising from registration of data sets created with considerably different imaging techniques.

1.1.2 Cooperation with Empa

The ILATO project—a joint so-called DACH project—is funded by Deutsche Forschungsgemeinschaft (DFG) under grant number *BO 864/17-1* and by the Swiss National Science Foundation (SNF) under grant *200021L 141311*. It is carried out by the *Interdisciplinary Center for Scientific Computing* (IWR) of Heidelberg University and *Empa* -

Swiss Federal Laboratories for Materials Science and Technology. The task is to investigate opportunities to speed up acquisition times in limited-angle computed tomography, to optimize CT trajectories and to improve quality of volumetric representation. Empa[1], with a strong reputation in X-ray analysis and non-destructive testing, focuses on improving reconstruction algorithms to incorporate prior knowledge. All CT equipment is located in Dübendorf (Switzerland) and the related acquisitions, measurements and analysis are carried out there. IWR[2] and the associated Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences (HGS MathComp), with competences in numerical geometry, and the processing of optically acquired data, provide such prior knowledge in form of surface scans aligned with the volumetric representations. All optical acquisition systems are located in Heidelberg (Germany) and the related measurements, comparisons and processing steps are performed. Throughout the whole project, lively discussions and exchange happened in regular telephone conferences, biannual meetings and several week-long research stays.

1.2 Input Data

All developed methods are tested against a variety of objects. Some are designed by us to study certain artifacts and were manufactured under controlled conditions. This provides us with exact knowledge of tolerance and accuracy during production. Others objects are real-world examples of synthetic data.

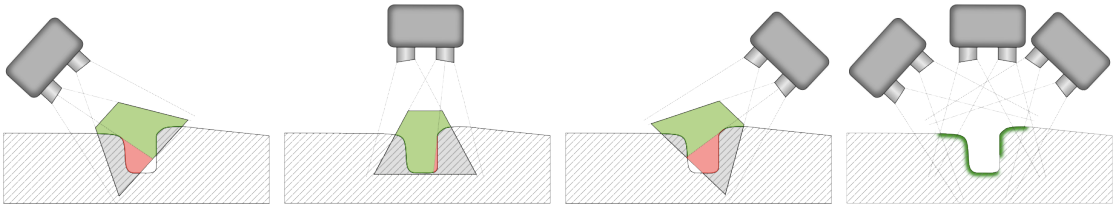
1.2.1 Optical Data Acquisition

Optical acquisition systems typically apply fringe pattern projection (active) and stereoscopic scanning (passive). The field of view in which data points are acquired, is restricted to the focal area of the camera system. Depth information per data point is computed by triangulation via disparity in camera views and displacement of the projected pattern. Thus, any data point acquired by optical systems must be visible either from both cameras or the projector and a camera (see Figure 1.4).

Stereoscopic Acquisition is a passive technique that typically uses two cameras with same focal length that are mounted parallel to each other. Both cameras view the same real-world point in a different location on the acquired 2D images. The projections of a real-world point in the left and right camera image have a distance which is known

[1] Empa—Eidgenössische Materialprüfungs- und Forschungsanstalt
[2] IWR—Interdisziplinäres Zentrum für wissenschaftliches Rechnen

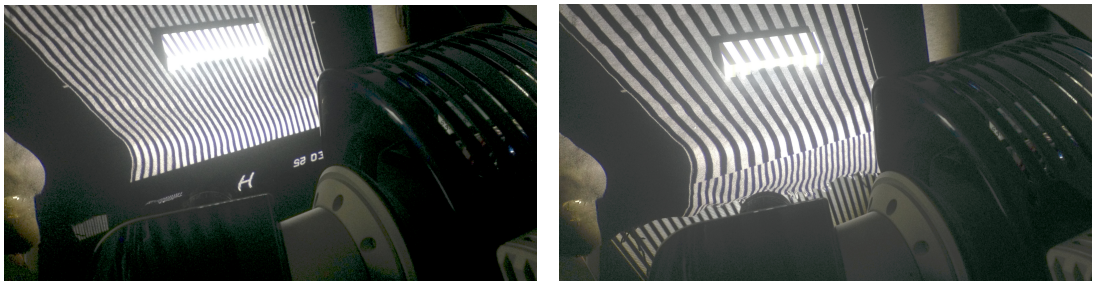
as disparity. This can be used to calculate depth information, which is the distance between the real-world point and the stereo vision system [Koc95].



(A) Scan 1 (B) Scan 2 (C) Scan 3 (D) Merge 1..3

FIGURE 1.4: Stereoscopic Acquisition.

Structured Light Scanners use fringe or Moiré projection and/or phase shift technology which is commonly referred as active scanning. Moiré patterns are a series of non-random linear projections onto the surface of the object. Multiple captures of the same pattern, slightly shifted, improve the measurement accuracy but increase acquisition times [BBS10]. By projecting a regular pattern onto an object (Figure 1.5) and recording several images, the resulting image depth information is obtained and a 3D model can be constructed as point cloud. Further information about the continuity of the surface is represented by connecting points and forming triangles.



(A) Pattern 1 (B) Pattern 2

FIGURE 1.5: Fringe pattern projected on specimen.

Limitations of optical scanning arise since optical surface scanners are unable to acquire data points in narrow cavities or deep trenches. Also, the surface mesh $\mathcal{M}_{\text{Opt}}$ cannot reveal any interior structure or undercuts. Therefore, $\mathcal{M}_{\text{Opt}}$ may have defects on the captured surface, which manifest as holes in the mesh. Other holes are caused by reflective, translucent or matte black surfaces that are very difficult to acquire due to physical limitations in the optics.

Conditions for representability of object features in the resulting mesh can be formulated as follows: Any vertex v in $\mathcal{M}_{\text{Opt}}$ satisfies

$$\mathcal{M}_{\text{Opt}} = \{v \,|\, \exists \triangle(v, d_1, d_2) \text{ with } (\measuredangle(v) > \phi \ \wedge \ \text{object }) \cap (\triangle(v, d_1, d_2) = \emptyset)\}\,. \tag{1.1}$$

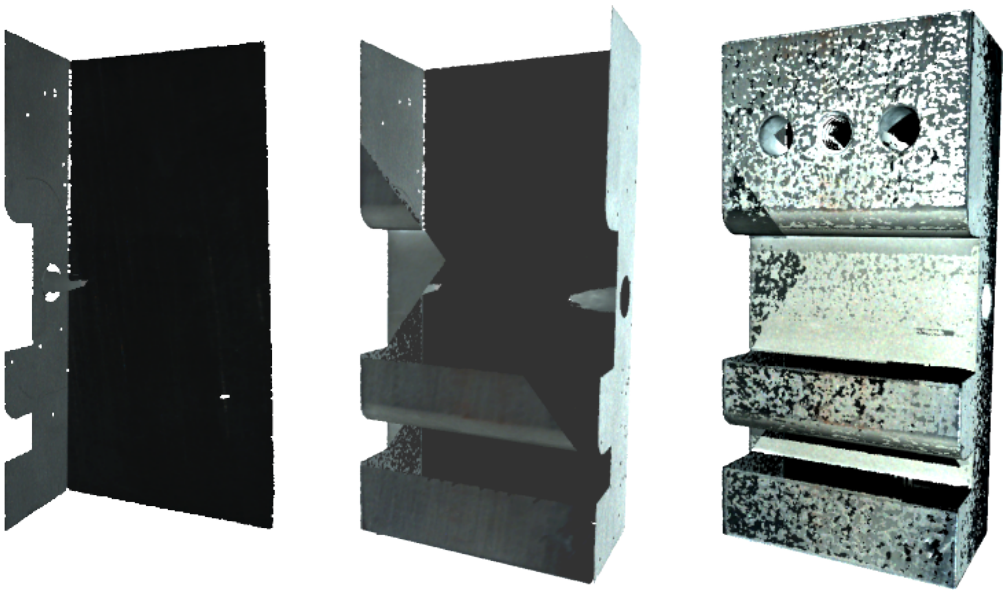


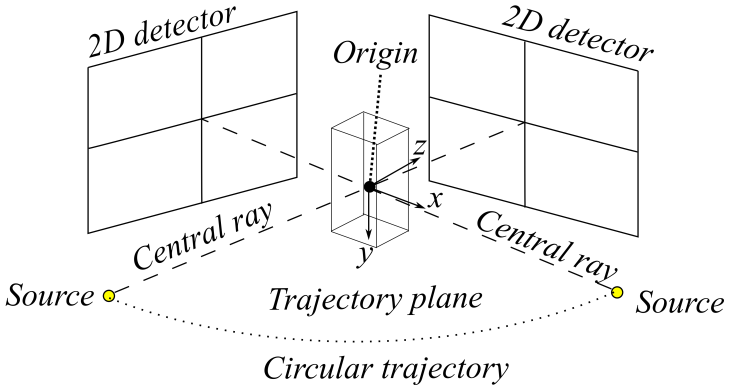
FIGURE 1.6: $\mathcal{M}_{\text{Opt}}$ of a sample object from 1, 4 and 8 partial scans (left to right). Narrow trenches and cavities are not captured. Surface: 117.89 cm^2; vertices: $\sim$ 1.6 million.

This implies the condition of an unblocked view from cameras d_1 and d_2 to any point on the object. The opening angle ϕ of the triangle $\triangle(v, d_1, d_2)$ depends on the specific setup of the optical scanning system and describes the disparity angle of either one camera and the projector in the active case, or both cameras in the passive case. Therefore, the minimal opening angle of any cavity of the object defines which data points can be acquired.

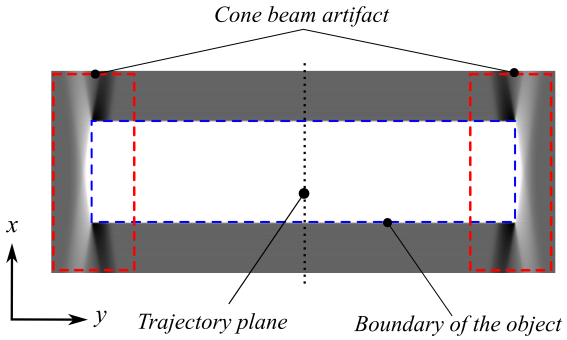
Optical 3D scanning augments a 2D image of a scene, acquired via visible light, with depth information per pixel by disparity measurement. The surface data derived by optical measurement typically consists of vertex-based data that includes geometric information, i.e., position, normal vectors, color, and to some extent also material information, e.g., from diffraction. These so-called point clouds are provided with a connectivity list to form a mesh, and most of the 3D-scanner software provides semi-automatic tools to generate meshed surface models from multiple scans. Regardless of post-processing steps it is a very detailed representation of the exterior features. But due to undercuts or blocked view, interior structures and shaded regions are not preserved. Several single acquisitions, each capturing a part of the total surface, need to be merged to obtain a complete representation of the objects' surface (Figure 1.6). For optical scans the accuracy depends on reflectiveness, texture and color of the material. Furthermore, merging all partial acquisitions to the final representation introduces alignment errors [BBS10]. Typically, all processing steps from disparity measure for partial scans to assembling the final mesh are encapsulated in the accompanying scanner software.

1.2.2 X-ray based Data Acquisition

Cone beam X-ray Computed Tomography (CBCT) is a cross sectional imaging technique derived from conventional X-ray imaging. X-rays emit from a point source, forming a cone shape, and interact with the object under investigation. The interaction follows the Beer-Lambert law according to which the transmission of the X-ray is related to the line integral of the attenuation coefficients of the object along a ray. A planar detector placed behind the specimen, perpendicular to the central ray, measures the intensity of each ray. The resulting 2D image corresponds to a conventional X-ray image and is referred to as a projection. In CBCT, a series of such projections are acquired while the source and detector pair is moving along a predefined trajectory with respect to the object, as in Figure 1.8. In a legacy CT, the trajectory is a full circle around the object while the center of the circle lies in the object.



(A) Schematic of the CT geometry.

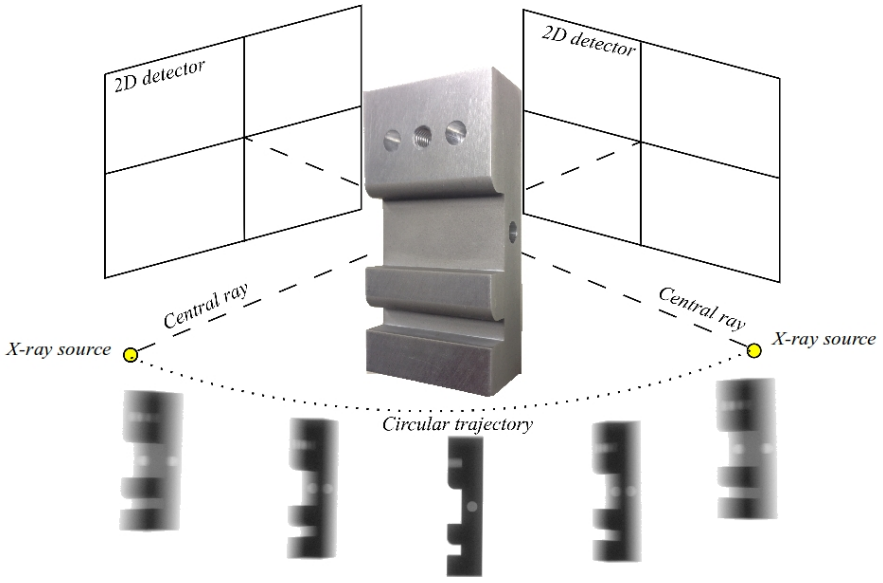


(B) Cone beam artifacts in red areas.

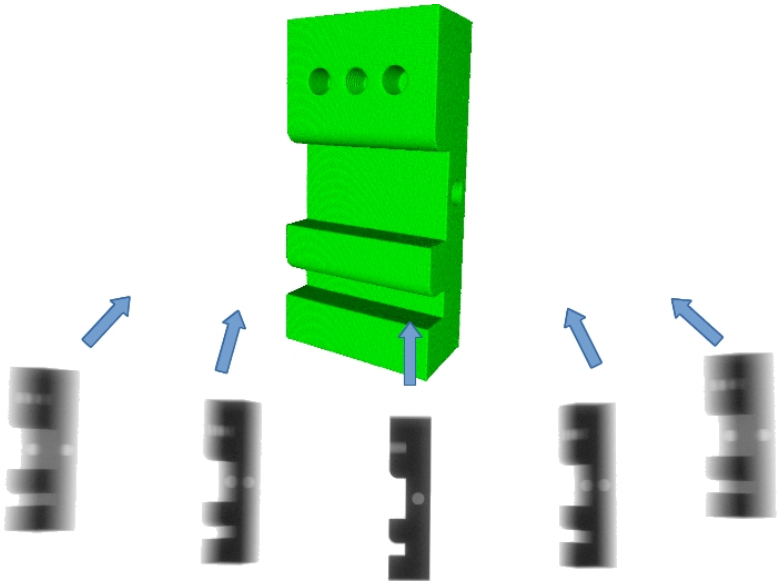
FIGURE 1.7: CT system arrangement and visualization of common defect.

With the projection images and a full circular trajectory, the attenuation coefficients in the illuminated area which contains the whole object can be computed using reconstruction algorithms such as the Feldkamp–Davis–Kress (FDK) method [FDK84]. Since the attenuation coefficients in the area are not homogeneous, the result is often represented

as a 3D grid of voxels, which leads to a problem when trying to fuse data from optical scanning represented as polygons. Either a Marching Cubes algorithm [LC87] or in our case Volume Enclosing Surface Extraction Algorithm (VESTA) [Sch12] is applied to generate a watertight surface, i.e., a mesh free of holes, $\mathcal{M}_{\mathrm{CT}}$ from the scalar data on the dense voxel grid as reconstructed from the CT scans. This $\mathcal{M}_{\mathrm{CT}}$ is an isosurface for a certain density value and usually represents the interface between object and surrounding air. It segments the volumetric data set according to the given threshold in areas below and above this density value.



(A) Example of projections along trajectory.



(B) Reconstruction from projections.

FIGURE 1.8: CT setup example for acquisition and reconstruction.

Limitations and artifacts of CT are related to Tuy's sufficiency condition [Tuy81], which suggests that only the attenuation coefficients in the circular trajectory plane can be

exactly reconstructed. In the rest of the volume, cone beam artifacts arise due to the uncertainty of the attenuation coefficients.

The surfaces that are parallel to the trajectory plane are blurred by this effect. This leads to a reduced spatial resolution in y-direction which further causes segmentation problems. As shown in Figure 1.7B, the boundary of the object (blue) is not properly reconstructed within red areas.

Limited Angle Computed Tomography (LACT) scan uses a trajectory that is less than a full circlem, thereby violating Tuy's condition. The reconstruction from limited-angle scans is an underdetermined problem which has non-unique solutions [Ram91]. LACT is used for specimens that are impenetrable from certain directions or substantially deviate from circular symmetry. As this provides incomplete data for the reconstruction algorithm, additional artifacts are introduced. In order to enhance the spatial resolution and the contrast of CT images, complementary information, e.g., optical or even tactile measurements, might contribute to all stages of a CT scan. As shown in Figure 1.9, a correct localization of attenuation coefficients in the volumetric representation is not possible. The leaking or fading surface parts are a direct result of the improper distribution of attenuated energy within the reconstruction volume.

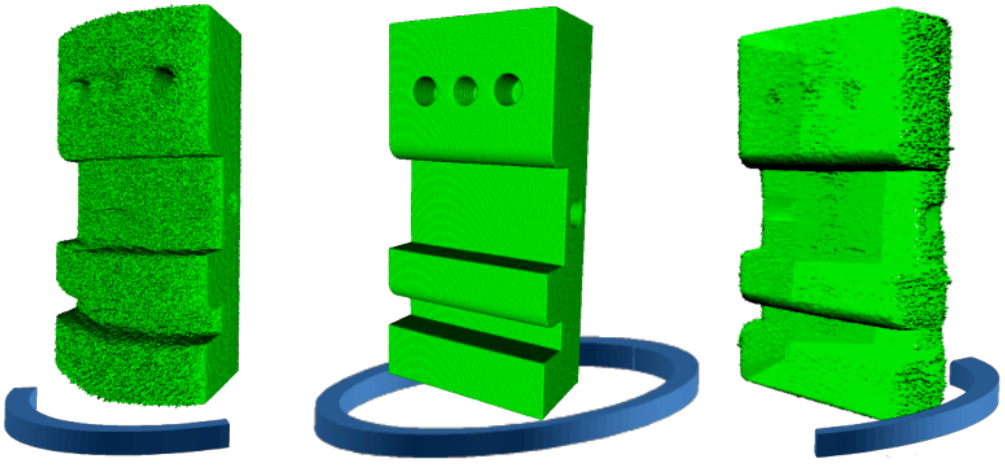


FIGURE 1.9: LACT scans with trajectory indicated by blue arcs (left and right). $\mathcal{M}_{\mathrm{CT}}$ from full angle CT (center, surface: 123.11 cm^2; vertices: 118,398).

1.2.3 Computer Aided Design (CAD) Files

Manufacturing of industrial samples is often based on CAD descriptions. They provide the complete geometry information of the object under investigation as polygon surfaces and can easily be transformed in triangular meshes (Figure 1.9). For quality assessment and indication of wearout, the CAD files can be compared to results from acquisitions of the specimen. Although, they do not provide the current state of the

object—as, e.g., optical scans do—but the original design blueprint it was manufactured from. Thus, interior structures are included, but manufacturing inaccuracies introduce a new source of errors.

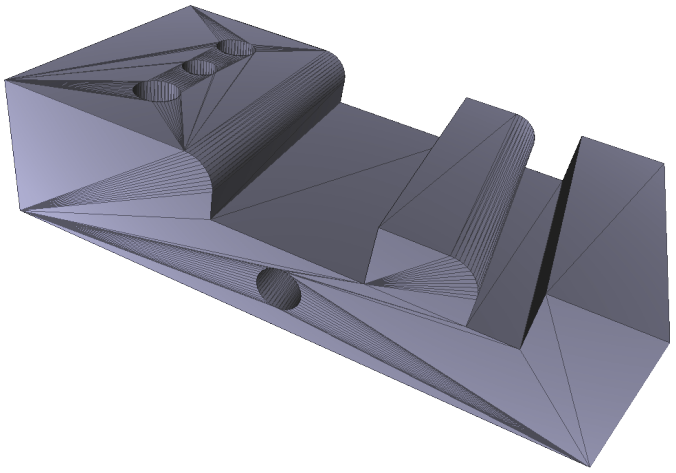


FIGURE 1.10: CAD of sample 1A (surface: 126.4 cm^2; vertices: 464).

1.2.4 Tactile Measurements

Tactile measurements are performed manually, by a computer numerical control (CNC) milling machine with specialized accessories or by a computer measuring machine (CMM). Tactile measurements can provide ground truth for the expected outer dimensions of the specimen. Similar to optical scans they usually are not able to reveal interior structures. Furthermore, free form surfaces are impossible to acquire since their parametric form is usually not recoverable from a few control points. The output consist of distance measures between reference points in the case of manual measurements or CNC measurements. Manual measurements with a caliper achieve an accuracy of 1 μm, while the CNC mills used to produce our sample objects have an accuracy of 15-25 μm. The output of CMM machines consists of parametric forms of cylinders and planes. For cylindrical surfaces several measurement points at the cylinder wall are acquired, and fitted to the parametric form of a cylinder. The result is a point on the rotation axis of the fitted cylinder, a normal vector along the rotation axis, the diameter of the fitted cylinder and a quality measure estimating the roundness of the cylinder, i.e., the deviation of measurement points and the fitted cylinder. For planar surfaces, several measurement points are acquired and fitted to a plane. The result comprises one point on the plane, a normal vector perpendicular to the plane and a quality measure estimating the flatness of the plane, i.e., the deviation of measurement points to the fitted plane. Typical CMM output is presented in Appendix A and visualized in Figure 1.11.

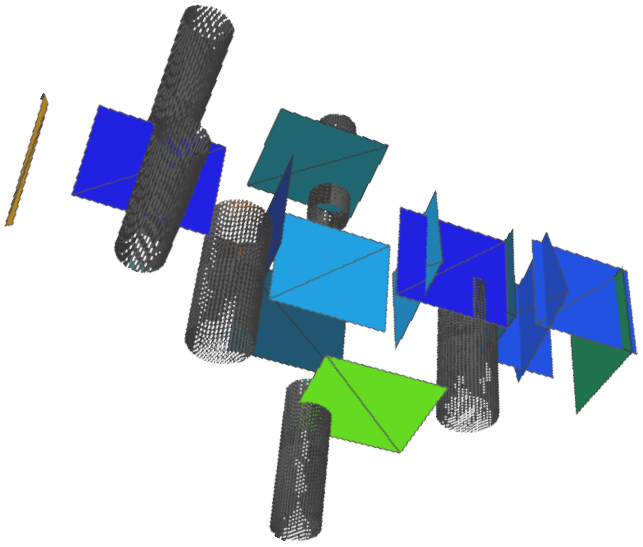


FIGURE 1.11: CMM visualization for sample object in Figure 1.8A.

1.3 Data Representation

The presented imaging techniques are fundamentally different and also the resulting data differs. Optical scans gather surface information which is represented as mesh while attenuation based measurements need to go through reconstruction before they are available as volumetric data set.

1.3.1 Surface as Mesh

Surface models from optical acquisition emerge from a series of pairs of 2D images captured from arbitrary positions around the specimen. Each pixel in the 2D image which can be tied to depth value is part of the data set. This happens either by identifying a corresponding pixel in the affiliated 2D image (passive acquisition) or by observing a known, projected pattern in the acquisition area (active case). The resulting representation is referred as point cloud. To form a surface mesh of planar polygons, the neighborhood relations or all points need to be resolved. The list of vertices with given x, y, z coordinates is enriched by a connectivity list describing edges between two vertices, triangles between three vertices or in general polygons formed by the referred vertices. Thus, a mesh consists of a set of data points and their connectivity. Other sources of surface meshes are parametric descriptions, e.g., in the case of CAD models, which need to be discretized to obtain vertices and their connectivity on the parametric

surface. Also volumetric representations usually provided as scalar field can be represented as surface mesh by identifying an interface by a given threshold and generating the corresponding isosurface.

1.3.2 Volume as Scalar Field

Reconstructing a series of X-ray projections along a known trajectory to a volumetric representation, typically in a three dimensional regular grid, is done by solving the inverse radon transformation [Rad17]. Each projection captures the attenuation of the specimen from a certain direction. Reconstruction assigns the voxels in the reconstruction volume local density values to obtain a configuration which provokes the observed attenuation measurements. The grid size corresponds to sensor pixels on the detector and the distances between source and specimen or detector and specimen. Typically, the resulting scalar field of densities, represented as grid, has identical spacing in x and y direction, but might have another spacing in z direction, i.e., along the rotation axis.

1.4 The Complete Data Fusion Pipeline

Enabling the reconstruction algorithm to consider prior knowledge, is subject of another PhD thesis within the ILATO project, carried out by Liu Yu at Empa in Dübendorf, Switzerland. The next section presents a brief summary of the data fusion approach to highlight the necessity of an accurate alignment and the benefits of including optical data as prior information.

1.4.1 Prerequisites

Improvements in spatial representation and new metrology tasks are expected from data fusion of $\mathcal{M}_{\mathrm{Opt}}$ and $\mathcal{M}_{\mathrm{CT}}$. A prerequisite is of course an acquisition of the specimen by both imaging techniques. For the CT acquisition, this includes an initial reconstruction from acquired projections. The reconstruction assigns each cell within the reconstruction volume an attenuation value to approximate the local density of the specimen within the region of this cell, resulting is a 3D scalar field of densities. The reconstruction concludes the standard CT workflow. Optical scanning provides the triangle mesh $\mathcal{M}_{\mathrm{Opt}}$ by merging all partial acquisitions of the objects' surface. To convert the CT scalar field in a comparable representation, a density threshold for the isosurface $\mathcal{M}_{\mathrm{CT}}$ needs to be determined and the isosurface extraction performed. This process is described in Section 3.2.2.

1.4.2 Alignment & Mask Generation

An alignment of optical acquisition $\mathcal{M}_{\text{Opt}}$ and isosurface $\mathcal{M}_{\text{CT}}$ is computed, providing a 4x4 affine transformation matrix T, as described in Sections 4.2 and 4.1. Fusing both data sets requires the generation of a mask from $\mathcal{M}_{\text{Opt}}$ to provide clipping information to a subsequent reconstruction of the original projections from CT acquisition. This mask, aligned to $\mathcal{M}_{\text{CT}}$ via T, formulates constraints on the distribution of attenuated energy within the reconstruction volume. A second reconstruction—we call it "augmented reconstruction"—incorporates the mask as prior knowledge to prevent distribution of attenuated energy in regions clearly marked as air during optical acquisition. Such clipping information is either formulated per ray, i.e., in each projection for each line from the X-ray source to each sensor pixel, or per cell within the reconstruction volume. A ray based mask requires—or at least benefits—from a preprocessing step for $\mathcal{M}_{\text{Opt}}$, in which small fragments are removed and holes filled to generate a watertight model. The geometry of the CT setup and the exact location of each pixel and the source are known from the trajectory file, defining the CT acquisition procedure. On the one hand, such a mask is computationally expensive since, depending on the CT resolution, billions of raycasts need to be performed. On the other hand, the intersection of each CT ray with $\mathcal{M}_{\text{Opt}}$ yields the locations of all intersections along the ray and in addition information about whether the mesh was entered or exited at the specific location. Thus, the ray-based mask implicitly answers the question in which direction clipping is performed. It constrains distribution of attenuated energy outside the mask, a description is given in Section 4.5.1.

A volumetric mask provides clipping information per cell and is computed from the intersection of each triangle of $\mathcal{M}_{\text{Opt}}$ with the reconstruction volume. Dimensions, grid size, stepping and spacing of this volume are known from the initial reconstruction, as the alignment T is provided. Each cell intersected by a triangle is tagged in the mask and in case hole-filling was applied to $\mathcal{M}_{\text{Opt}}$, the information if the cell is intersected by an artificial patch or real data is also included. During an augmented reconstruction this mask permit distribution of attenuated energy within the boundaries of the mask; a description is given in Section 4.5.2.

1.4.3 Data Fusion

As overview and to present the complete process, the following is a brief summary of our data fusion method published by Liu et al. [LBS+16], which integrates optical surface scans with CT projections. A prerequisite to the presented method is a successful alignment and the computation of a volumetric mask.

In an industrial cone beam CT scanner, the relative position of the source to the flat-panel detector is fixed, therefore the geometry of one projection can be uniquely defined by the source position $s(s_x, s_y, s_z)$, and the normal vector of the detector $n_s(n_x, n_y, n_z)$. The set $T = \{(s, n_s)\}$, of the projection geometry vector is defined as the scanning trajectory. The projection operator A maps from the object vector space x, to the detector vector space y, using a scanning trajectory.

$$Ax = y \tag{1.2}$$

In case of scanning with limited-angle circular trajectory, the projector operator contains a nontrivial null space. Let x_{null} denote the null space,

$$Ax_{\text{null}} = 0. \tag{1.3}$$

Therefore, the linear system has a non-unique solution $x + x_{\text{null}}$, which translates to possible volumetric artifacts in the reconstruction result.

For industrial applications it is save to assume that the image is sparse under some transformation, as most of the industrial parts consist of a limited number of piece-wise constant materials. Here, we formulate the reconstruction problem in the compressed sensing framework:

$$\mathbf{x} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\|\psi(\mathbf{x})\right\|_1, \quad s.t. \quad \mathbf{A}^{\Phi}\mathbf{x} = \mathbf{y}^{\Phi} \quad \text{with scanning angle} \quad [-\Phi, \Phi] \tag{1.4}$$

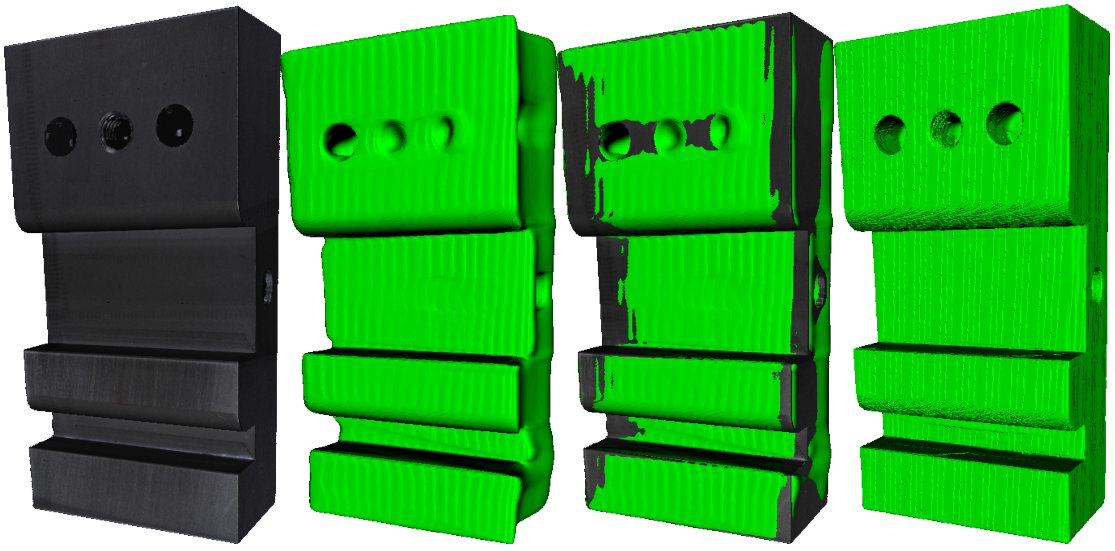
where ψ is some sparsifying operator transforming the image to sparse representations.

The pre-aligned mesh representing the optical surface scanning result is discretized into the reconstruction grid to generate vector B, which contains all voxels that intersect with the mesh as information about the reconstruction boundary. B represents the prior knowledge based on the aligned mesh $\mathcal{M}_{\text{Opt}}$ after generating the volumetric mask.

$$\mathbf{x} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\|\mathbf{D_B}\psi(\mathbf{x}) + \alpha\overline{\mathbf{D_B}}\psi(\mathbf{x})\right\|_1, \quad s.t. \quad \mathbf{A}^{\Phi}\mathbf{x} = \mathbf{y}^{\Phi} \tag{1.5}$$

where $\mathbf{D_B} = \text{diag}\{B_i\}$, $\overline{\mathbf{D_B}} = \text{diag}\{1 - B_i\}$ and $\alpha \in [0, 1)$.

We use the gradient operator as the sparsifying operator ψ and update Equation (1.4) such that the gradient calculated at the boundaries tagged in $\mathbf{B}$ are preserved because they have a smaller weight α in the cost function.



(A) $\mathcal{M}_{\text{Opt}}$ (B) $\mathcal{M}_{\text{CT}}$ (C) Overlap (D) Result

FIGURE 1.12: Fusion inputs from optical scan and CT, their overlap and the resulting data set.

After transforming the unconstrained optimization to a constrained optimization problem, the reconstruction result is calculated by minimizing the cost function:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\| \mathbf{A}^{\Phi}\mathbf{x} - \mathbf{y}^{\Phi} \right\|_2^2 + c \left\| \mathbf{D}_{\mathbf{B}}\psi(\mathbf{x}) + \alpha\overline{\mathbf{D}_{\mathbf{B}}}\psi(\mathbf{x}) \right\|_1 \tag{1.6}$$

where the first term is the data fidelity term. In the second term, c controls the smoothing strength and α the weight of the optical constraints. To search for the minimizer $\hat{\mathbf{x}}$, we use the Alternating Direction Method of Multipliers [BPC+].

1.4.4 Fusion Result

Clearly, an enhancement in object representation was achieved from CT input (Figure 1.12B) to fusion output (Figure 1.12D). The exterior object boundary provided as mask, computed from the optical scan in Figure 1.12A, enables a segmentation of the reconstruction volume and corrected blurring artifacts. The superposition of both input data sets in Figure 1.12C visualizes the defective areas. Given then fact that both acquisitions describe the very same object, their regional distinctions are remarkable. This also emphasizes the necessity for highly accurate alignment which takes typical CT artifacts into account. The three drilling holes in the top section of Figure 1.12D are exemplary for the achieved improvements. Neither one of the input data sets provide the correct representation of drilling hole bottoms. Yet, the combination of penalizing energy distribution outside the boundaries of $\mathcal{M}_{\text{Opt}}$ and ensuring smooth gradients within the scalar field permit their correct reconstruction.

1.5 Key Problem

The data fusion task requires a very accurate alignment of these object representations from fundamentally disparate imaging techniques. The problem arises from the imbalance of represented information and difference in spatial resolution. For full-angle CT data and accompanying optical scan data of the very same object, inaccuracies in alignment do occur [BMK14] as shown in Figure 1.13. Due to the corresponding artifacts, limited-angle data presents an even greater challenge. An accurate alignment is not feasible as long as internal structures contribute to the alignment error. A higher degree of accuracy can be achieved if only essential data points contribute to the alignment error. Consequently, aiming for the preservation of relevant parts and the omission of incomparable regions, we need an efficient data reduction. In our setup, a mesh $\mathcal{M}_{\text{Opt}}$ has a very high resolution up to $\sim$10 μm, but lacks all data from internal structures. Isosurfaces $\mathcal{M}_{\text{CT}}$ from volumetric data sets contain, in contrast, all interior and exterior structures, but generally have a lower accuracy of only $\sim$75 μm. So, for each data point on the exterior surface from CT, we have several data points describing the very same surface in the optical scan. However, the interior surface contained in CT data is not represented in optical data at all.

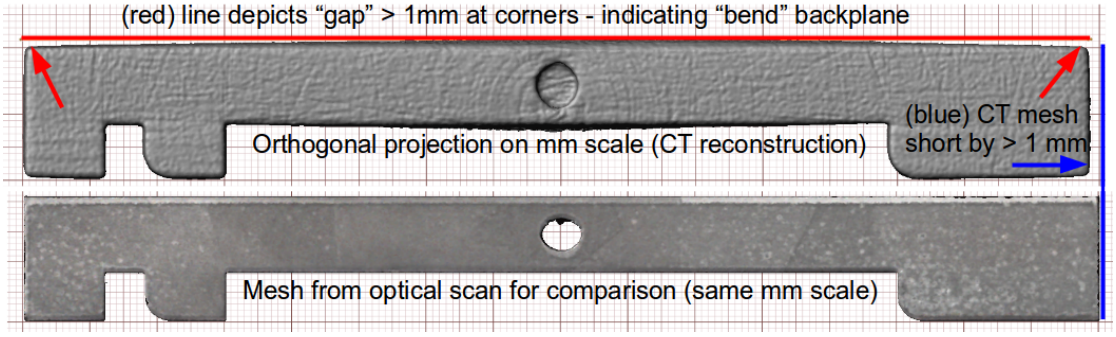
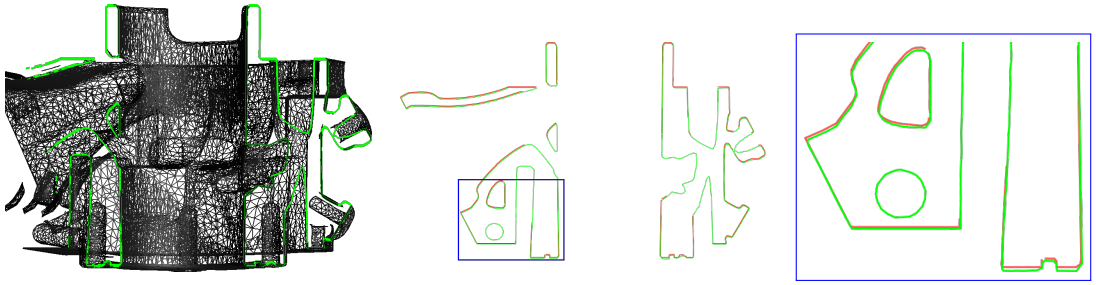


FIGURE 1.13: Elongation and blurring comparison.

The exterior surface of an object, in our context, includes all surface parts visible from the outside. According to Equation (1.1), $\mathcal{M}_{\text{Opt}}$ is only a fraction of the complete exterior surface, which in turn is a fraction of all the data included in $\mathcal{M}_{\text{CT}}$. $\mathcal{M}_{\text{Opt}}$ and $\mathcal{M}_{\text{CT}}$ provide different representations of the identical object, and to find a suitable alignment, we definitely have to take those differences into account.

Similar to acquisitions of the sample object in Figure 1.13 and the mismatches between both representations, the same observation can be made for the industrial cylinder part in Figure 1.2. Slight elongations and mismatch in scale hamper an accurate alignment, as shown in the cross-section in Figure 1.14.



(A) Cross-section of $\mathcal{M}_{\mathrm{CT}}$ with wireframe behind the cutting plane.

(B) Cross-sections of aligned meshes

(C) Zoom to blue box in (b)

FIGURE 1.14: Mismatch between $\mathcal{M}_{\mathrm{CT}}$ (green) and $\mathcal{M}_{\mathrm{Opt}}$ (red).

On the one hand, it is worth mentioning that both examples of $\mathcal{M}_{\mathrm{CT}}$ and $\mathcal{M}_{\mathrm{Opt}}$ contain artifacts and none of these data sets is assumed to be the more trustworthy or correct representation. On the other hand, both CT data sets are acquired with a full angle trajectory and therefore free of LACT artifacts which would of course increase the deviation. Thus, to achieve a proper alignment we will omit all vertices which only contribute to error. Areas without a suitable counterpart in the other mesh need to be neglected.

Chapter 2

Basic Principles and Sample Data

This chapter briefly explains details and characteristics of the single acquisition methods, mainly with focus on flaws and artifacts. In the following, established algorithms, 3rd party data formats, libraries and tools are presented and a description of all sample objects within this work is provided.

2.1 Volumetric Reconstruction from 2D Projections

Reconstruction algorithms can be divided in two groups, analytical reconstruction and iterative reconstruction. They serve the same goal of identifying a density value configuration within the reconstruction volume, responsible for the observed measurements, but follow different approaches. Also, each method allows for a different set of filters and correction methods or error detections. The aspect of introducing prior knowledge in form of a mask and also to apply regularization motivated the works on an analytical reconstruction algorithm within the context of the ILATO project.

2.1.1 Iterative Reconstruction

The principle of iterative algorithms is to find a solution by successive estimates. The current estimate and the projections it is based on are compared to all measured projections. The current estimate is then modified according the result of the comparison, thereby creating a new estimate [Bru02]. In the so-called Algebraic Reconstruction Technique (ART), projections are discretized, yielding a huge linear system of equations. Statistical reconstruction methods take into account the random nature of the measurements; they are based on the minimization of the distance between the measured

data and the estimations given by a statistical model. Iterative reconstruction is computationally expensive compared to analytical approach and tends to provide increased resolution. The drawback is that each iteration amplifies noise.

2.1.2 Analytical Reconstruction

This technique is based on the concept of backprojection. The accumulation of the ray-sums of all the rays that pass through any voxel in the reconstruction volume is processed. In practice this consists of two steps: At first, sinograms are run back through the image to obtain a rough approximation to the original. Since this causes some blurring in other parts of the reconstructed image the second step is to apply a deconvolution filter or ramp filter per projection to mitigate the blurring effects. Systems differ in choice of applied filter and order of steps but in general follow this idea. Analytical methods have the advantage to be fast and deliver good quality results under standard scanning conditions [ML07]. To enhance filtered backprojection approach a mask defining boundaries of the object can be included in reconstruction algorithm. This is called weighted filtered backprojection.

2.2 Defects in Data Representation

All imaging techniques produce output with artifacts, which either are *false negatives* or *false positives*. A *false negative* is the absence of data which is expected to be represented in the output. To compensate this and to enable further processing, the missing information is either added via data fusion or the defective region is corrected by adding synthetic data in a best-effort approach. *False positives* describe the presence of data which has no correspondence to the real-world object. These artifacts have to be detected and possibly deleted. The correct representation of real-world data points in generated output is called *true positive*. And the correct absence of data in the representation, e.g., no data points in regions, where no structures are to capture, is called *true negative*. It is important to notice that the interpretation of these terms sometimes differs according to the context. For surface data such as optical scans, absence of acquired data results in absence of represented data. For transmission-based measurements the absence of data can also be interpreted as *absence of observing attenuation*, e.g., caused by defective sensor pixels. In the resulting volumetric representation this is responsible for falsely high density values in some voxels. So in a CT workflow, depending on the chosen representation, *false negatives* in projections result in *false positives* after reconstruction.

2.2.1 False Data in $\mathcal{M}_{\text{Opt}}$

False negatives in optical scanning are very common and usually referred to as a hole. False positive are extremely rare and show up as surface fragments above the true surface. They only occur for highly reflective materials. In this case the fringe pattern is projected on one surface part, and its reflection captured from a different surface part. Or in general the mirroring of a surface part is falsely selected for disparity measurement. For the same reason, small surface fragments are identified as holes since both are enclosed by a mesh boundary. Determining if the current boundary encloses a hole or a small fragment is usually accomplished by calculating the surface area near the boundary, i.e., triangles without three adjacent triangles. If applying a region growth algorithm to one boundary vertex accumulates an area larger than a given threshold, before all triangles within the boundary are selected, the boundary is assumed to represent a hole. Likewise, boundaries that enclose triangles with a total area below a given threshold are assumed to represent small fragments which usually are deleted. In addition to defects related to the acquisition of data, the process of merging all single acquisitions to a final representation also introduces errors. The surface parts are aligned based on their overlap—usually by manually selecting tie points—and depending on the geometry of the object small errors can occur. An a priori estimation of accumulated errors through the whole workflow is not possible [BBS10]. Our a posteriori analysis of multiple measurements for a single specimen are listed in Appendix E.

2.2.2 False Data in $\mathcal{M}_{\text{CT}}$

For CT a variety of sources can introduce artifacts. This brief summary of common flaws [BF12, Hut02] is given to highlight the width of the spectrum of potential defects. As for optical acquisitions, this work does not study the contribution of single artifact sources. Our focus on limited-angle artifacts is justified by the given usecase and the possibility to correct those defects by providing prior information, yet this is not the only—and depending on the setup not the most prominent—source of artifacts.

Noise occurs due to statistical error of low photon counts. They result in random thin streaks in $\mathcal{M}_{\text{CT}}$. False photon counts are also caused by reflections or photons emitted by the walls of the CT chamber after long exposure. The effects of noise are increased during iterative reconstruction. Artifacts introduced by noise are so common that to synthetic data usually so-called Poisson noise is added to mimic the characteristics of a real acquisition.

Beam Hardening describes the effect of low energy X-ray photons being attenuated before high energy photons. Thus, attenuation does not follow exponential decay. This

leads to misinterpretations of projection data during reconstruction since the algorithm assumes exponential decay. As a result attenuation values along the ray appear lower than they are in reality, the further the ray traveled in dense material. This effect is only seen in polychromatic x-ray, not with a monochromatic x-ray setup. In other words, sources that guarantee to only emit photons with exactly the same energy do not suffer from beam hardening. For homogenous materials characteristic cupping artifacts are attributed to beam hardening. The effect is widely studied and well known, yet there seems to be no suitable detection or prevention mechanism available. Best effort correction attempts apply fitting of a polynomial on the relation between the logarithm of the transmission and the path length in the (homogeneous) material in order to adjust the transmission values. The effect of beam hardening is substantially reduced by a strong filtration of the source spectrum, e.g., by introducing Tungsten foils or blocks between source and specimen to achieve *almost monochromatic* beam characteristics.

Scatter causes more photons to be detected than expected. This results in dark streaks along the lines of greatest attenuation. The sensor areas behind dense material should not gather energy and therefore are most sensitive to falsely distributed photons. Like Noise, this is a statistical effect caused by miscounts, but in contrast to it, the geometry of the specimen influences the effects of Scatter.

Limited-Angle Artifacts occur in LACT scans and represent an under determined system for reconstruction as explained in Section 1.2.2. Since it does not cover a full circular trajectory, the lack of projections from certain angles does not permit to resolve all surface configurations. Optical scans resolve data points on surfaces parts orthogonal to the line of view, but never on surface parts parallel to it. For CT it is the other way round, interfaces between materials, e.g., aluminum and surrounding air, parallel to the central ray are resolved best. If full attenuation is measured in one pixel and zero attenuation in a neighboring pixel the maximum of information is collected for reconstructing the specimens volumetric representation. Two neighboring pixels collecting identical attenuation values—as it is the case for interfaces orthogonal the the central ray—do not provide any meaningful information. Thus, the lack of projections from certain angles usually implies the lack of segmentation information and therefore only permit a vague distinction between material interfaces after reconstruction. This results in leaking or blurring of material borders in volumetric representation and creates inconclusive isosurface parts in $\mathcal{M}_{\mathrm{CT}}$ as shown in Figure 2.1.

Ring artifacts can be observed as concentric circles after reconstruction and usually originate from a miscalibration of CT setup. If position or inclination of the rotation stage, carrying and moving the specimen, are not accurately specified, all projections are processed with a slight offset. This results in rings around the rotation axis and

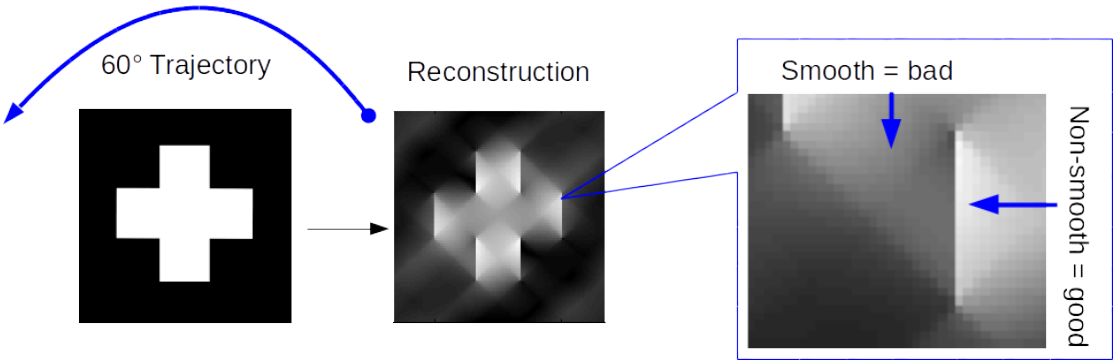


FIGURE 2.1: Effect of trajectory on reconstruction and subsequently on isosurface extraction.

may also be caused by very heavy specimens physically bending the mechanics of the rotation stage. No matter wether it is caused by miscalibration or by exceeding the payload, either objectively or virtually the rotation stage performs a wobbling motion instead of a rotation.

2.3 Quality Measure for Alignment

Any kind of manipulation needs a suitable verification to ensure the correct execution. Metrics and measurements to determine the correctness of a given alignment or at least allowing to assess and rank two different alignments are presented in the following.

2.3.1 Mean Square Error

The Mean Square Error (MSE) between two point sets $\mathcal{P}$ and $\mathcal{S}$ measures the average squared distance between neighboring point pairs $u_i \in \mathcal{S}$ and $v_i \in \mathcal{P}$.

$$\text{MSE} = \frac{1}{n}\sum_{i=0}^{n}(u_i - v_i)^2, \quad u_i \in \mathcal{S}, \quad v_i \in \mathcal{P}, \quad i = min(|S|, |P|) \tag{2.1}$$

As for sets of different sizes no set of unique pairs between a sample from one set and its neighbor in the other set covers the sum of all samples in both sets, the MSE evaluation is a directed measure. Thus, $\text{MSE}(\mathcal{S}, \mathcal{P})$ yields a different result than $\text{MSE}(\mathcal{P}, \mathcal{S})$. In addition, outliers have a mayor impact since error distances are squared. To address the latter, the Root Mean Square Error (RMSE) or Root Mean Square Deviation can be computed as

$$\text{RMSE}(\mathcal{S}, \mathcal{P}) = \sqrt{\text{MSE}(\mathcal{S}, \mathcal{P})}. \tag{2.2}$$

This quadratic mean is a common measure of the imperfection of the fit or alignment. The lower the result of RMSE($\mathcal{S}, \mathcal{P}$), the more data points from $\mathcal{S}$ are in close proximity to $\mathcal{P}$.

2.3.2 Hausdorff Distance

The Hausdorff distance computes the largest error distance between two sets of points. It can be used as a similarity measure for registration [ZHW05] or validation [KGA+10]. As it is based on the Euclidean norm it is dimension-independent. Other norms like, e.g., Manhattan distance are also possible but less common. The result is always equal or greater zero while equality is only reached if the compared sets are identical.

Let p be a point of $\mathbb{R}^3$ and $\mathcal{S}$ a triangulated 2-dimensional surface embedded in $\mathbb{R}^3$. The distance δ from p to $\mathcal{S}$ is defined as

$$\delta(p, \mathcal{S}) = \inf_{q \in S} ||p - q|| \tag{2.3}$$

where $||.||$ corresponds to the Euclidean norm and q is a point of $\mathcal{S}$—not necessarily a vertex in $\mathcal{S}$. Depending on the implementations q can be located in one of the triangle surfaces instead of being a corner point of any triangle.

Now, let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two 2-dimensional surfaces embedded in $\mathbb{R}^3$ and p_1 a point belonging to $\mathcal{S}_1$. Equation (2.3) allows to define a surface-to-surface relative distance $\Delta(\mathcal{S}_1, \mathcal{S}_2)$ as

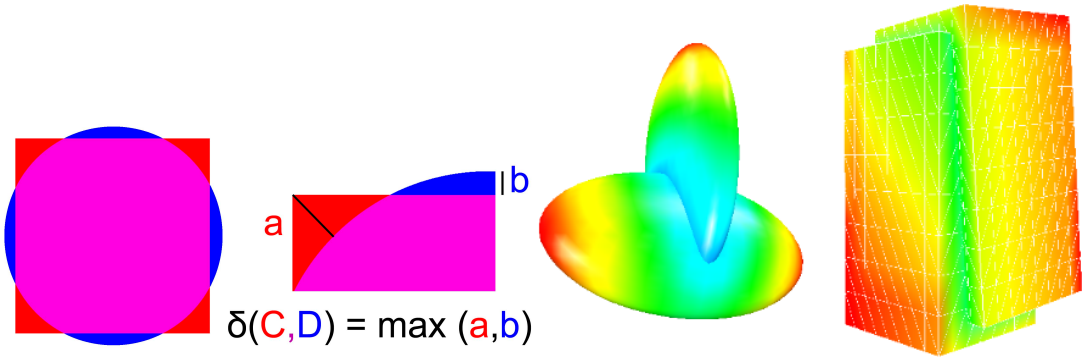
$$\Delta(\mathcal{S}_1, \mathcal{S}_2) = \sup_{p1 \in \mathcal{S}_1} \delta(p1, \mathcal{S}_2) \tag{2.4}$$

This distance is relative as it is not symmetrical, i.e., $\Delta(\mathcal{S}_1, \mathcal{S}_2) \neq \Delta(\mathcal{S}_2, \mathcal{S}_1)$. Finally, the Hausdorff distance d between two surfaces $\mathcal{S}_1, \mathcal{S}_2$ is defined as the maximum of the two relative distances:

$$d(\mathcal{S}_1, \mathcal{S}_2) = \max \{\Delta(\mathcal{S}_1, \mathcal{S}_2), \Delta(\mathcal{S}_2, \mathcal{S}_1)\} \tag{2.5}$$

In contrast to RMSE, the Hausdorff distance is the largest of all the distances from a point in one set to the closest point in the other set, not averaged over multiple error distances. It is a necessity that RMSE is computed while calculating the Hausdorff distance.

For the intersecting ellipsoids in Figure 2.2B the distance measures in each vertex were mapped to a color ramp. The values range from blue—which refers to minimal distance—over yellow to red, while red refers to the largest error distance measured.



(A) $\Delta(Circle, Square)$ as example for directed measure.

(B) Hausdorff error distances for visualizing mesh deviation or alignment error.

FIGURE 2.2: Minimal Hausdorff example and coloring based on error distances.

The intersecting boxes also feature a mapping from measured error distances to a color range—in this case green is set to correspond with zero error. Measuring the Hausdorff distance based on vertex to surface instead on vertex to vertex, allows to take the surface orientation into consideration and—considering the normal vectors—might result in negative error distances to indicate that the vertex is actually behind the surface. As the Hausdorff distance is a directed measure, in both cases—ellipsoids and boxes—each of the intersecting meshes was processed, and the calculate error values mapped to a color range.

2.4 Data Set Analysis

To grasp the global features of a data set and quantify attributes of the represented object we need to determine, e.g., surface area, volume or barycenter. Furthermore, they might support decision making on orientation and expansion or characterize the data set as being roughly cuboid or rather spheroidal.

2.4.1 Surface Area

Surface Area Calculation follows "Herons formula". Let a, b, c be the lengths of the sides of a triangle. The area is given by:

$$A = \sqrt{p(p-a)(p-b)(p-c)} \tag{2.6}$$

2.4.2 Box Volume

The volume V of a box is calculated as $V = whl$ where width w, height h and length l in this case correspond to the extents of the enclosed vertices. The corner points of the box around a subset of all data or the whole data set are in the simple case defined by the smallest and largest coordinates in xyz direction.

$$p_{min} = (\min(x), \min(y), \min(z)), \quad p_{max} = (\max(x), \max(y), \max(z)) \tag{2.7}$$

corresponds to corner points from all permutations of min and max for xyz coordinates and yields the volume of a so-called axis aligned bounding box.

2.4.3 Center of Gravity

Calculating th Center of Gravity $\bar{v}$ from all vertices v in the data set $\mathcal{P}$ is done by averaging all position vectors

$$\bar{v} = \frac{1}{n} \sum_{i=0}^{n-1} v_i, \quad n = |\mathcal{P}|, \quad v_i \in \mathcal{P}. \tag{2.8}$$

Data: Eigenvectors from E, point CoG
Result: Octant CoGs 1..8
set $CoG_q = (0,0,0)$ for $q \in (1..8)$;
set $nrCoG_q = 0$ for $q \in (1..8)$;
for $v_i \in \mathcal{P}$ **do**
  $q = \text{getOctant}(v_i)$;
  $CoG_q \mathrel{/}= v_i$;
  $nrCoG_q{+}{+}$;
end
for $i \in (1..8)$ **do**
  $CoG_q \mathrel{+}= nrCoG_q$;
end

Algorithm 1: OctaCoG update

2.4.4 Dimensionality Reduction

Principal Component Analysis (PCA) identifies the major axes of a data set as linearly independent components, i.e., the first component indicates the direction of largest variance. Succeeding components indicate the largest variance within the expansion, under the constraint that they are linearly independent of all components with higher variance. In the 3D case, up to three major axes can be identified which in turn represents an orthogonal basis.

PCA can be computed as the eigendecomposition of the sample covariance matrix A

$$A = \frac{1}{n}\sum_{i=0}^{n-1}(v_i - \bar{v})(v_i - \bar{v})^t, \quad v_i \in \mathcal{P}. \tag{2.9}$$

Thus, $A = EDE^t$ provides eigenvalues λ_j as diagonal entries in D and orthonormal eigenvectors $e_j, j \in 1, 2, 3$ as columns of E. This represents principal components in direction e_j with ordering given by the magnitude of the j^{th} eigenvalue.

Data: $\mathcal{P}$
Result: PCA and CoG
$A^{3,n}$ = Zeros(3,n);
for $v_i \in \mathcal{P}, i \in (0..n-1)$ **do**
  A.row(i) = v_i ;
  CoG += v_i;
end
CoG /= n;
A.rowwise() - CoG;
E = A.eigenvectors().sorted();

Algorithm 2: PCA and CoG

A similar method and generalization of the eigendecomposition is singular value decomposition (SVD). It also provides an orthogonal transform that decorrelates the variables and keeps those with the largest variance. Compared to PCA, SVD is known to be numerically more stable [KL80]. Assuming the same centered $m \times n$ data matrix P as in Equation (2.9), where $A = PP^T$ the SVD of P is calculated as

$$P = U\Sigma V^T \tag{2.10}$$

with a unitary $m \times m$ matrix U, a non-negative $m \times n$ diagonal matrix Σ and a unitary $n \times n$ matrix V. Decomposition provides $P\vec{v} = \sigma\vec{u}$ and $P^*\vec{u} = \sigma\vec{v}$ where vectors $\vec{u}$ and $\vec{v}$ are called left-singular and right-singular vectors for σ, respectively.

In analogy to the covariance matrix $A = PP^T = EDE^T$ from PCA, the similarity of both approaches can be seen with

$$PP^T = (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V\Sigma U^T) = U\Sigma^2 U^T \tag{2.11}$$

since V is an orthogonal matrix $V^TV = I$. Thus, the square roots of the eigenvalues of A correspond to the singular values of P and eigenvectors in E correspond to left-singular vectors u.

2.4.5 Local Linearity

To calculate the local linearity [RL15], we compute the three eigenvalues λ_i of the PCA of the point set in 3D space

$$l = \frac{\max(\lambda_i)}{\sum(\lambda_i)}, \quad i \in 1..n. \tag{2.12}$$

For the n dimensional case, l is in the range of $\frac{1}{n} \leq l \leq 1$. The unit vectors are uniformly distributed for $l = \frac{1}{n}$, i.e., all eigenvalues are equal. Likewise, $l = 1$ indicates that all vertices of the point cloud are collinear, thus all but one eigenvalues are zero since all points contribute exclusively to the first principal component. In the 3D case, a linearity measure of $l = \frac{1}{3}$ corresponds to uniformly distributed points on a sphere.

2.5 Example Data

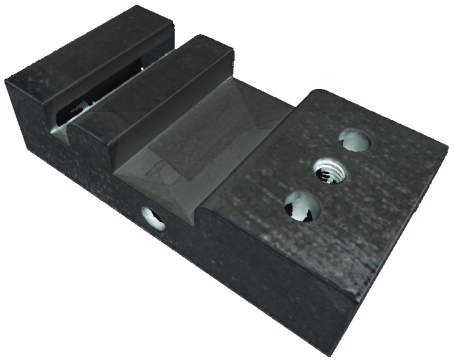
Our research aims at improving the understanding of the intrinsic resolution limits of CT, allowing maximum spatial resolution and contrast with a minimum of artifacts.

To enable the comparison of meshes generated from volumetric data and meshes generated from optical scans several sample objects have been selected. Some have been designed by us and produced at Empa, others are real-world samples in mint condition or with wear traces. Our students finished the task of scanning those sample objects with Breuckmann scanners *SmartScan 3D SM2069-HE5* and *SmartScan 3D SM152712-HEC8LW* featuring 5 MP and 8 MP cameras and offering a resolution of $18\mu m$ and $20\mu m$ at IWR.

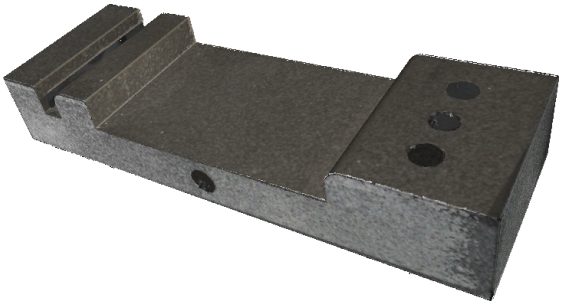
2.5.1 ILATO Samples

We designed test objects to provoke and evaluate typical artifacts related to the imaging techniques. The resulting representations, the amount of data points and the captured surfac are shown in Figure 2.3 for the optical measurement system and in Figure 1.9 for processing in a CT workflow. The objects are made from aluminum blocks with brushed surface finish, and were manufactured in different sizes. The dimensions of the presented object are 8x4x2 cm, elongations of the long edge to 12,16 and 20 cm were applied for the other objects. They feature drilling holes in front and side, milled trenches of different size in the front and one screw thread applied to the center drilling hold on the front. Two identical sets of four samples each were manufactured at Empa according to CAD specifications and labeled 1A..4A and 1B..4B. For all samples we performed

optical 3D scan at IWR, CT scans at Empa, manual measurements via Caliper by IWR Feinwerkstatt, and in addition the B set was measured via CMM machine by Empa.



Our smallest sample 1B



ILATO sample 2A

3B—wedge shaped due to flaws in production

The largest sample 4A

FIGURE 2.3: ILATO samples from optical scanning.

As indicated in Figure 2.4 we experimented with different coatings to resolve issues with highly reflective surfaces, namely developer spray from photography domain and Cyclododecan spray from the conservator domain. Although both were able to effectively reduce the reflectiveness or the surface, the thickness of the additional coating layer rendered the scan result useless. Also, the advertised effect of evaporation without residue could not be observed.

2.5.2 An Industrial Real-World Object

We also processed industrial real-world objects to demonstrate practicability and to further evaluate our method. It is—not disclosed for public presentation due to an

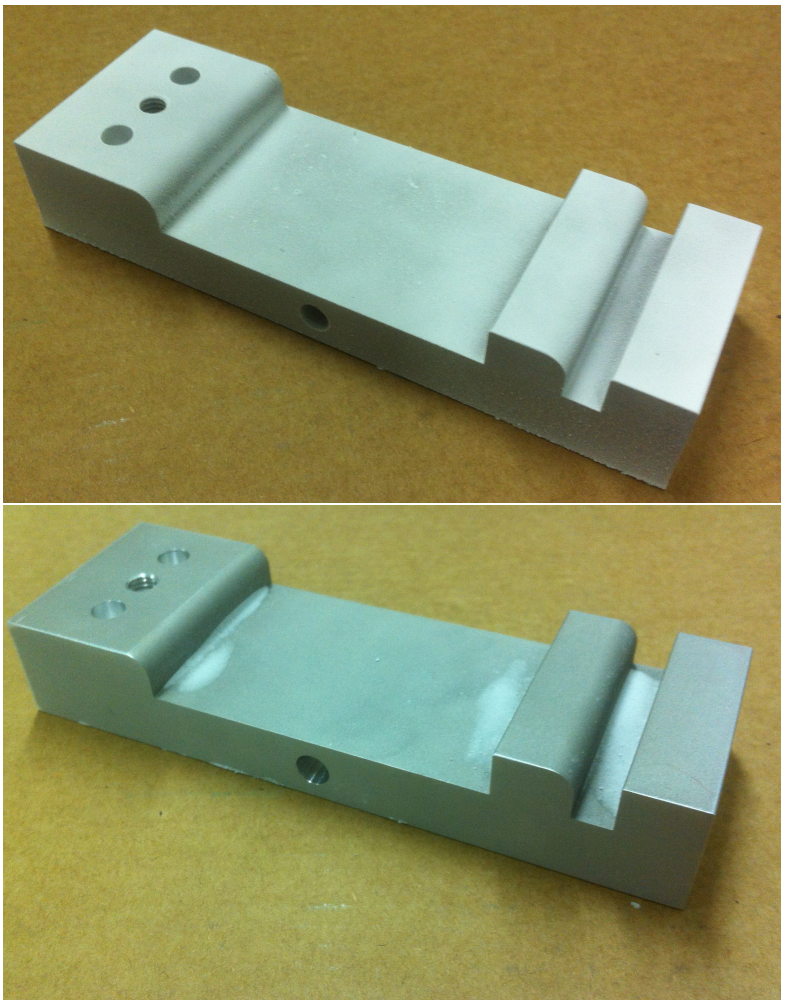


FIGURE 2.4: ILATO sample 2B with freshly applied coating (top) and residue after evaporation (bottom).

NDA—included in the results Section 5.2 as "industrial object". As an alternative specimen featuring double walled asymmetric machinery parts, the synthetic data set called "wing model" was created and evaluated.

2.5.3 Wing Model

Synthetic Data of a hollow object is presented in Figure 2.5. The surface model of a wing has no interior structures—as an optical scanner would capture it. To reflect the characteristics of real acquisitions, the original CAD sample was subdivided [CC78] to increase the number of vertices and a random displacement was added as noise. This serves as mesh $\mathcal{M}_{\mathrm{Opt}}$ from optical scanning.

In Figure 2.6 the same wing model has a downscaled version nested as inner structure to serve as lined sample, as double-walled specimen are very common in real world applications. To create $\mathcal{M}_{\mathrm{CT}}$, the vertex count was again reduced—after nesting the

TABLE 2.1: Measurements for ILATO samples.

Sample	W	H	L	Surface (cm^2)	Vertices
1A60	40.20	20.72	80.13	116.0	1666236
1A	40.28	20.33	80.15	117.9	281812
1B	40.02	20.63	80.01	145.6	355635
2A	40.52	21.00	120.05	154.2	413894
2B	40.14	20.41	120.07	155.7	327752
3A	40.06	20.05	160.30	198.5	418383
3B	40.20	20.84	160.08	200.3	436105
4A	40.16	21.04	200.13	237.6	502275
4B	40.33	20.58	200.19	241.2	498702

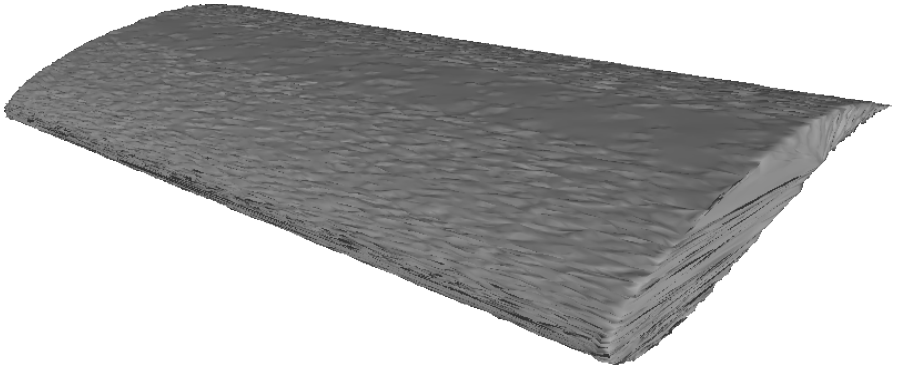


FIGURE 2.5: $\mathcal{M}_{\text{Opt}}$ of synthetic wing data with detailed surface but no interior structures. surface: 303 cm^2; vertices: 19,298.

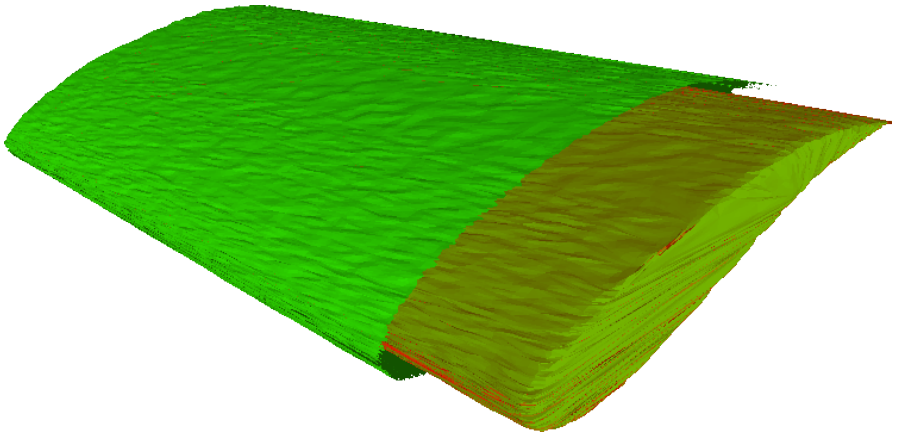
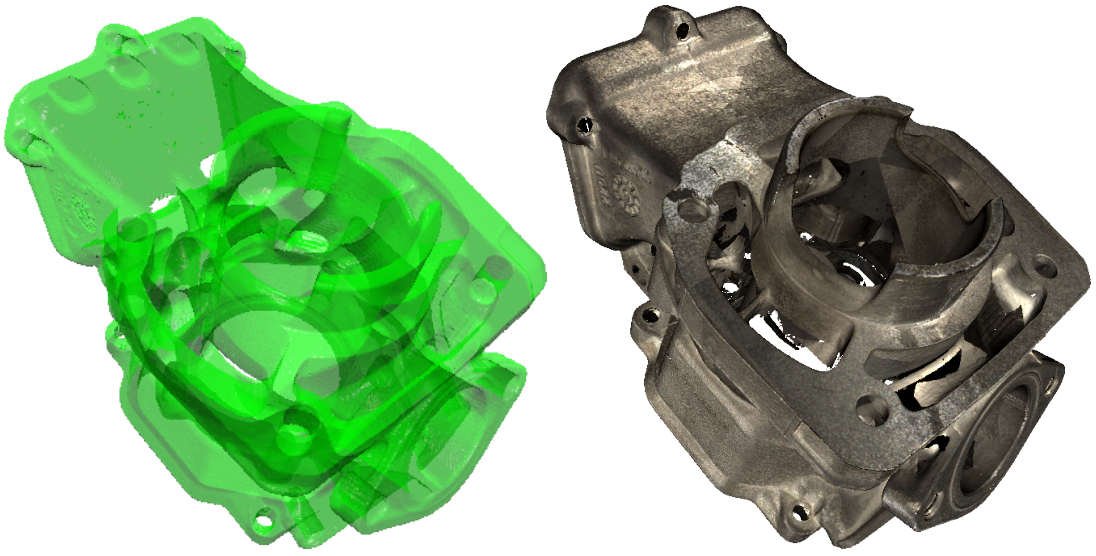


FIGURE 2.6: $\mathcal{M}_{\text{CT}}$ of synthetic wing data with full coverage but lower level of detail. surface: 579 cm^2; vertices: 28,187.

interior structures—via quadric-edge collapse based mesh decimation, as typically a CT acquisition has lower resolution than a comparable optical scan.

2.5.4 Cylinder Cast

As presented in Section 1.1, this is a real-world object (Figure 2.7) from the industrial domain. It poses a challenge to optical acquisition and CT alike. Since complex interior structures include many undercuts, the surface in mesh $\mathcal{M}_{\text{Opt}}$ from optical scanning only comprises only a fraction of the CT result. On the one hand, embossments of serial numbers and logos are filigree surface structures well-preserved in optical scans but easily blurred or smoothed in CT data sets. On the other hand, screw threads interior structures are correctly represented only in the isosurface extracted from $\mathcal{M}_{\text{CT}}$. The cylinder cast object is made from aluminum and measures $18.3x11.9x10.9cm$. $\mathcal{M}_{\text{CT}}$ comprises $1491cm^2$ mesh surface in a $423cm^3$ volume with 1,352,529 vertices while $\mathcal{M}_{\text{Opt}}$ covers a surface of $1047.5cm^2$ with 9,519,090 vertices.

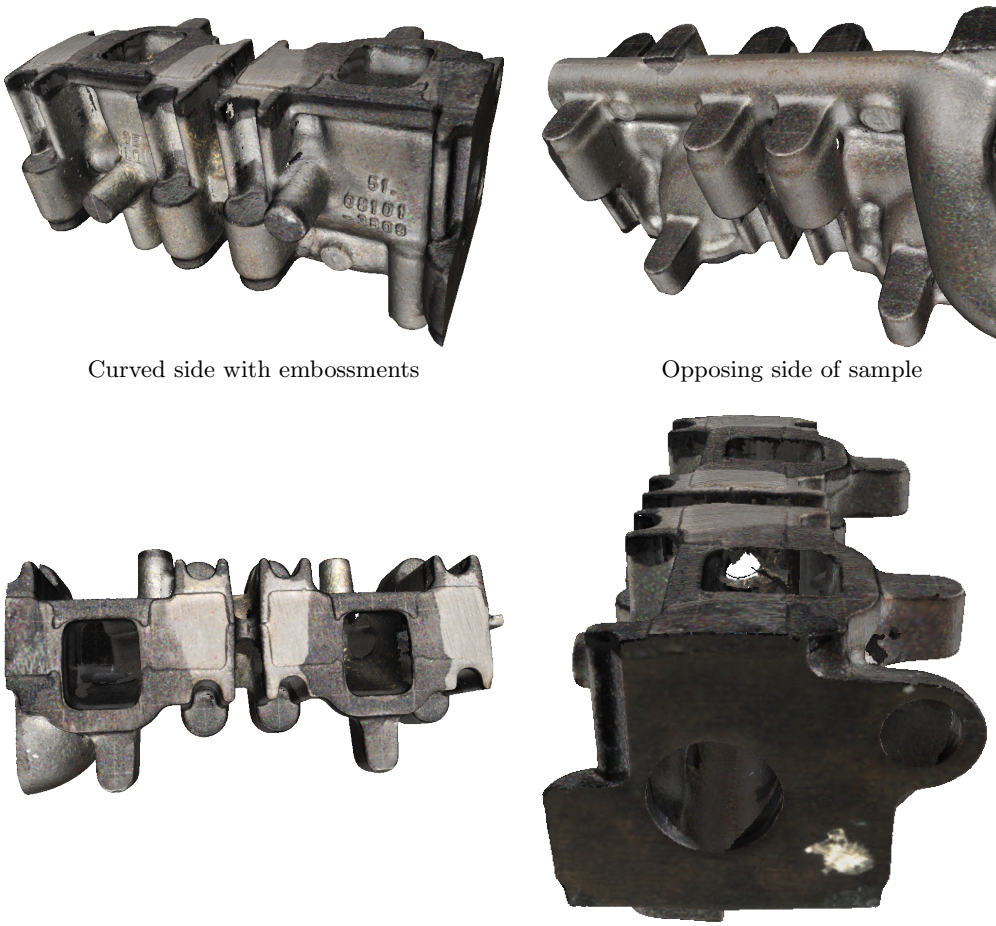


(A) Isosurface mesh $\mathcal{M}_{\text{CT}}$ (B) Mesh of optical scan $\mathcal{M}_{\text{Opt}}$

FIGURE 2.7: Industrial example of a cylinder cast as acquired by CT and optical scan.

2.5.5 Steel Sample

Certain artifacts in CT scanning only occur for very dense materials—or at least are more prominent there. We present a cast iron sample which measures 234.9x105.4x120.0cm, $\mathcal{M}_{\text{Opt}}$ comprises $1183cm^2$ mesh surface in a $881cm^3$ volume with 788560 vertices. Several processing steps were performed, each contributing a different surface texture. The surface is rough and matte as typical for cast products in original surface parts. Polished surfaces in the processed outlets are highly reflective (Figure 2.8D) and grinding marks—presumably caused by dismantling the sample with a metal saw—feature filigree streaks (Figure 2.8C).



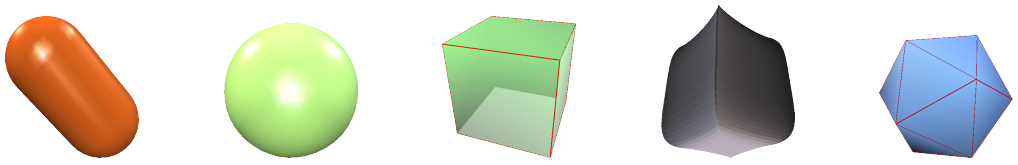
Curved side with embossments | Opposing side of sample

Bottom with saw marks | Polished inlet

FIGURE 2.8: Cast iron sample with 1184 cm^2 surface in a volume of 879.5 cm^3 represented by 788560 vertices.

2.5.6 Primitives for Testing Purposes

Several primitives were generated to evaluate algorithms and highlight results of operations such as computing the enclosing ellipsoid, the Minimum Bounding Boxes or a space partitioning. Furthermore, information obtained from trajectory files or similar is indicated. Super-Ellipsoid [Bar81] also served as random sample object featuring random configurations of convex/concave synthetic test data.



Capsule | Sphere | Cube | Super-Ellipsoid | Icosahedron

FIGURE 2.9: Primitives for testing purposes.

2.6 3rd Party Libraries and Formats

The successful implementation of concepts and algorithms presented within this work benefited from a variety of excellent open-source projects. They provide fundamental building blocks, such as versatile data structures, efficient algebra systems, numerical solvers and proper visualization as reliable well documented libraries and are presented in the following.

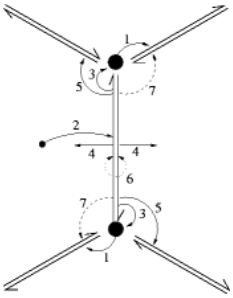
2.6.1 Data Structure

The mesh structure all operations are performed on is a half-edge data structure provided by OpenMesh [BSBK02], which also handles most of the I/O and data conversion in writing and reading the meshes from and to disk. OpenMesh implements a half-edge data structure, also known as doubly-connected edge list. For an embedding of polytopes in 3D, each vertex is stored with coordinates and other attributes such as color or normal vector. In addition, the connections to previous and next vertices are stored as links and referred as half-edges. Per vertex, an arbitrary number of connected vertices are stored by their outgoing half-edge, optionally also all incoming half-edges, i.e., references from other vertices to a single vertex can be stored. Besides edges and vertices, the data structure also holds faces. In general these are polytopes and in the case of surface meshes more commonly quads or triangles. Faces are implicitly defined by a closed cycle of connected half-edges, circling the face clockwise with respect to the normal vector of the face, i.e., the front side. As two adjacent faces contribute one half-edge each to their connecting edge, the opposing half-edges form a full edge.

The data structure provides iterators and circulators to exploit to connectivity information. Hence, one can iterate over all vertices, edges, half-edges and faces in a mesh. Circulators over all vertices of a face, all half-edges around one face or all faces around one vertex are provided, which are calles *one-ring-circulator*. Furthermore, the outgoing half-edges of a given vertex in clockwise or counter-clockwise order or combinations of the aforementioned connections can be traversed.

2.6.2 Linear Algebra

For linear algebra, vector and matrix operations the Eigen Lib [GJ+10] is employed. This also includes geometrical transformations and numerical solvers, e.g., for eigende-composition as in PCA and SVD.



- Each vertex references an outgoing halfedge that starts at this vertex (1).
- Each face references one of the halfedges bounding it (2).
- Each halfedge provides a handle to
  - the vertex it points to (3),
  - the face it belongs to (4)
  - the next halfedge inside the face (ordered counter-clockwise) (5),
  - the opposite halfedge (6),
  - (optionally: the previous halfedge in the face (7)).

TABLE 2.2: OpenMesh documentation of half-edge data structure.

2.6.3 Data Types

To extend the support of data types, mesh formats and to enable the loading of volumetric data, additional modules have been implemented. OpenMesh already supports common formats such as the *obj* format developed by Wavefront Technologies, STereoLithography or Standard Tessellation Language as *stl* format, the Polygon File Format or the Stanford Triangle Format as *ply* file and and *off* files for the Object file format. To benefit from existing implementations such as GigaMesh and the related MSII analysis (see Section 3.4), an importer for *mat* files storing the resulting feature vectors were implemented. Being able to import *mpl* MeshLab project files including the stored transformations and chosen perspectives per mesh has proven to be very convenient. By far more important is the interface to load CT geometry files.

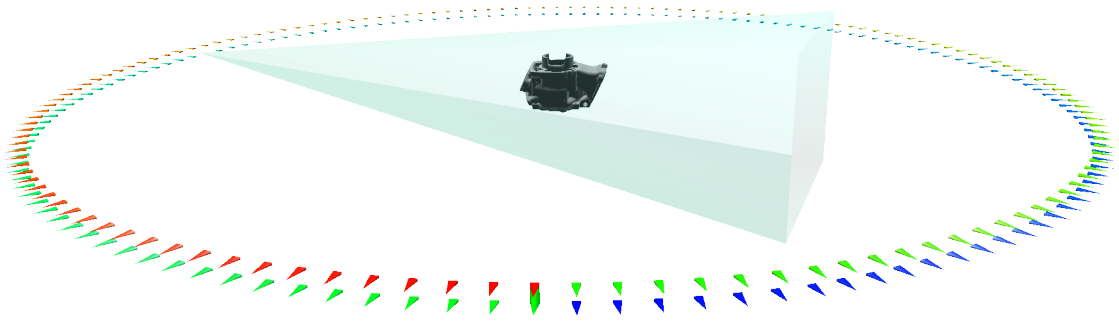
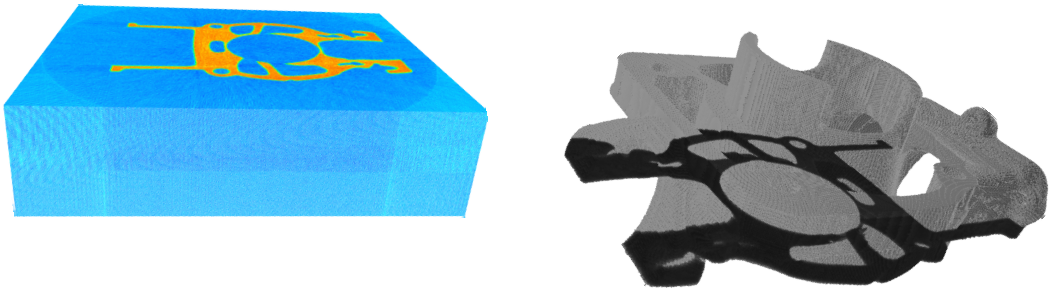


FIGURE 2.10: Visualized projections

They contain the geometric properties of the CT machinery such as pixel size, spacing and layout of the detector. Furthermore, the distances between source and detector and the center of the rotation stage are contained, the rotation axis around which the rotation stage turns the object. Also the trajectory is stored along with the geometric properties, describing the perspectives of all projections the CT performs. As visualized in Figure 2.10, small tetrahedrons indicate source positions along—in this example—a so-called Circle-Line-Circle (CLC) trajectory. The semi-transparent pyramid represents

the cone beam of a single projection from source (tetrahedron) to detector (bottom of pyramid). The geometry file is essential for operating the CT machine, and also for the subsequent reconstruction. An overview on the CT setup is given in Section 1.2.2, the reconstruction step is described in Section 2.1 and the geometry file as basis for generating a mask as prior knowledge to an enhanced reconstruction is explained in Section 4.5.1.

As for surface meshes, a variety of data formats exist to describe volumetric data sets. The most simple case is storing the 2D representation of a single layer of the reconstructed volume *raw* file. Several of those files or slices cat be stacked according to manually specified parameters like width and height of slice, pixel spacing, endianess and data type. After loading a series of *raw* files, each density value is represented as property of one voxel in volumetric representation—or vertex when represented as point cloud. The resulting structure forms a homogenous 3D grid according to width and height of each slice and total number of slices. An example of loading a the upper third of all stacked *raw* slices of the Cylinder Cast object is given in Figure 2.11, the color are mapped according to density per vertex using a standard *hot-to-cold* color ramp[1].



(A) Scalar field with vertices colored by density

(B) Upper 3^{rd} of slices (ony vertices above threshold)

FIGURE 2.11: Volumetric data loaded from *raw* slices.

A more common format for volumetric data is *mhd*, a MetaImage from MetaIO implementation (Figure 2.12) within the Insight Segmentation and Registration Toolkit (ITK) by Kitware which also provides an implementation of an alignment algorithm as

[1]Color ramp: `http://paulbourke.net/texture_colour/colourspace/`

presented in Section 3.5.2. Internally *mhd* relies on the *raw* file data type for representation but accompanies this with a meta file containing all information about cell size, dimensions, spacing and enclosed formats.

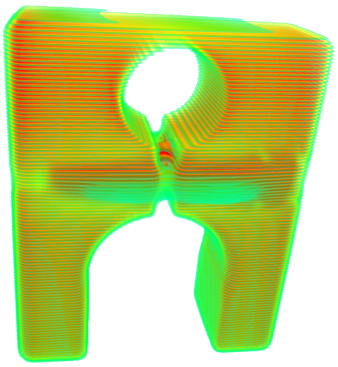


FIGURE 2.12: Object with different densities from MHD.

Chapter 3

Related Work

The following chapter describes existing concepts that were adopted in our project or included in the evaluation of own methods. This comprises post-processing steps to provide watertight surfaces from optical acquisitions, strategies to extract a mesh surface from volumetric data, alignment algorithms and intersection tests.

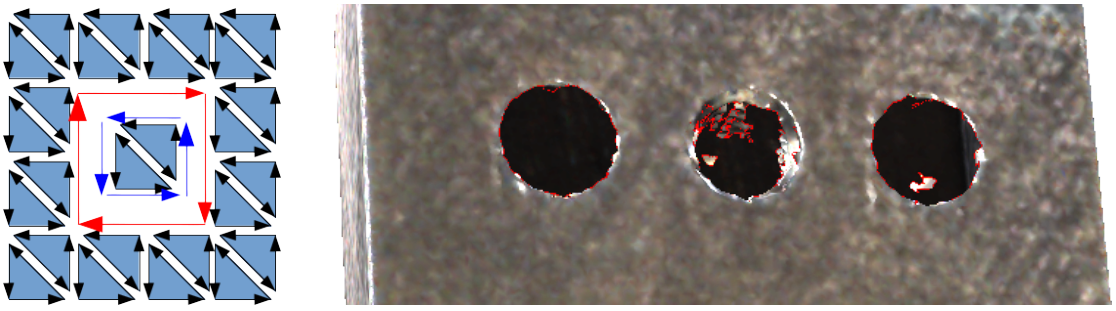
3.1 Holes and Fragments

As optical scans are not able to capture interior structures and also might not acquire all exterior surface parts, the defective areas have to be assessed and eventually corrected. The most prominent defects are holes and fragments.

3.1.1 Finding Holes in a Mesh

In the half-edge data structure, introduced in Section 2.6.1, each partial edge is represented as an oriented half-edge circling the corresponding face clockwise. Thus, an edge between two faces consists of two half-edges with opposite direction. A boundary is considered to be a single half-edge without a corresponding opposing half-edge. This indicates the presence of a face lacking a neighboring face as illustrated in Figure 3.1A.

To identify such boundaries as a cycle of connected half-edges, Algorithm 3 is employed. Border vertices can be identified in the data structure as being connected to border edges. Until any border vertex v_i is assigned to a closed cycle, the border half-edges connected to v are followed to neighboring border vertices v_j. Vertices already known to be part of a cycle are neglected; identifying the start vertex as next vertex indicates a closed cycle.



(A) Border edges (B) Borders of holes and patches

FIGURE 3.1: Indication of half-edges without opposing half-edge.

Data: $\mathcal{M}$
Result: allClosedCycles
allClosedCycles={};
for $v_i \in verticiesAtBorder$ **do**
  if $v_i \in allClosedCycles$ **then**
    skip this iteration;
  end
  while *current cycle not closed* **do**
    for $v_j \in outgoingBorderHalfedges(v_i)$ **do**
      if $v_j isknownincurrentcycle$ **then**
        skip this iteration;
      end
      Add v_j to current cycle;
      if v_j *identical with first vertex in current cycle* **then**
        add current cycle to allClosedCycles;
        break loop;
      end
    end
    if $currentcyclewasnotappended$ **then**
      break loop;
    end
    $v_i = v_j$;
  end
end

Algorithm 3: Finding closed cycles of half-edges in $\mathcal{M}$

3.1.2 Closing Holes

Two approaches to close holes are implemented. The first—a subdivision patch as depicted in green in Figure 3.2—is preferred since its patches are from triangles of similar shape and size. Also the produced patches respect the local curvature in the vicinity of each border vertex. The latter—a triangle fan patch shown in red—provides a fall-back solution for very large cycles or border cycles that deviate to a large extent from a circular shape.

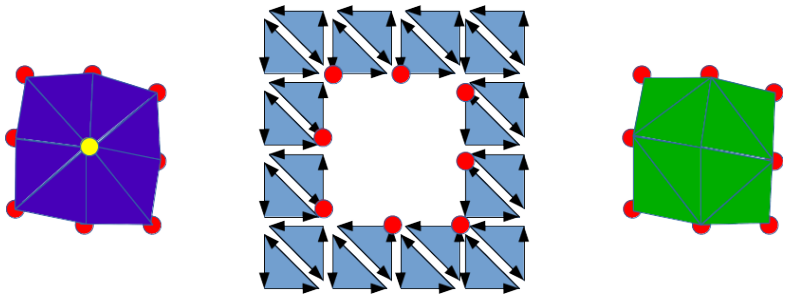


FIGURE 3.2: Example mesh (center) with border vertices indicated in red. Corresponding Triangle fan patch (left) with barycenter (yellow) and Subdivision patch (right).

Subdivision Patches provide a surface fragment resulting from recursively applying a refinement step to an existing surface part, which is called subdivision [Lie03]. According to a given set of rules the triangles of the existing surface part are successively split into smaller primitives, to replace the original surface part during each iteration. The subdivision patch not only considers the border vertices of a given hole in $\mathcal{M}$, but also the normal direction and average edge length of each border vertex. The result, provided as list of all new vertices and their connectivity, is a smooth surface part for the irregular triangle mesh $\mathcal{M}$. This approach works well for border cycles not largely deviating from circular shape but is computationally expensive for border cycles of several hundred edges. And as this approach does not provide intermediate results, the whole border cycle is either covered by the resulting patch, or no patch is provided at all. Figure 3.3 shows a subdivision patch applied to the right drilling hole from Figure 3.1B.

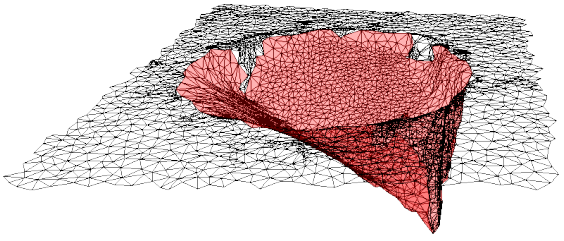


FIGURE 3.3: Partially acquired screw thread closed with subdivision patch.

The Triangle Fan Patch Algorithm 4 provides a very robust and fast solution to close a given hole in $\mathcal{M}$, indicated by its n border vertices. It generates n triangles from any two border vertices v_i and v_j that share a common border edge and the common barycenter bc of the whole cycle. The resulting surface part—in contrast to Subdivision patches—does not estimate the local curvature in the border region and most likely does not correspond at all to the missing surface part. In this approach all triangles have one edge defined by one arbitrarily small border edge and the opposing vertex defined by the barycenter bc. Thus, all triangles share one single vertex bc and tend to have a very unfavorable edge length ratio. For these almost-degenerate triangles, the processing is

prone to being numerically unstable. Due to intersections with existing surface parts in $\mathcal{M}$, the patch might not be fully applicable.

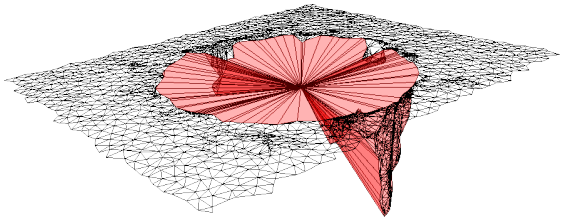


FIGURE 3.4: Partially acquired screw thread closed with triangle fan patch.

Intended as fallback solution when the subdivision patche construction fails, this is acceptable. As an advantage, the triangle fan patch can be partially applied and it provides solutions for border cycles of arbitrary length. Partial application refers to leaving out triangles intersecting existing surface parts. The resulting holes from applying a partial patch can be closed in consecutive runs. Figure 3.4 shows a triangle fan patch applied to the left drilling hole from Figure 3.1B. Note that all triangles of the patch intersect in a single point, the barycenter of the border cycle.

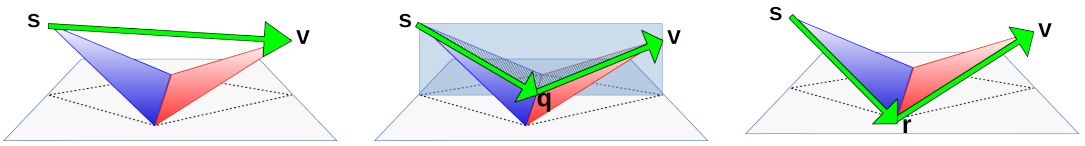
Data: closedCycle
Result: fanPatch
fanPatch={};
bc=barycenter(closedCycle);
v_{tmp}=last known vertex in closedCycle;
for $v_i \in closedCycle$ **do**
  add triangle (bc, v_i, v_{tmp}) to fanPatch;
  v_{tmp}=v_i;
end

Algorithm 4: Composing triangle fan patch from closed cycle of vertices.

3.1.3 Stencil Selection & Region Growth

Figure 3.1B depicts all half-edges without opposing half-edge in red. They either revolve around a hole in their center or circle the boundary of a small fragment. The distinction between fragment and hole cannot be made from the data structure itself, but needs to take certain aspects of the local neighborhood into consideration. For example whether a connected set of faces with a surrounding border is considered a fragment or whether missing faces manifest as a hole, is a subjective perception and only depends on the relation of face surface and boundary length.

A possible solution to distinguish hole from fragment is the application of a region growth algorithm with a certain threshold to any of the seed vertices. Region growth describes a selection scheme where connectivity or proximity are considered in order to compose



(A) Eucledian distance (B) Geodesic distance (C) Edge trail

FIGURE 3.5: Distance measures from a in blue triangle to v in red triangle.

a subset of all available data. From a given start or seed vertex s, all components are selected which either are located in a given distance d or are reachable via a set of connected edges. The distance measures from the seed vertex s to any selected vertex v either consider the Euclidean distance $\overrightarrow{sv}$, as in Figure 3.5A, or the geodesic distance, as in Figure 3.5B. For the latter, an intermediate point q denotes the intersection of the adjacent edge and the projection of $\overrightarrow{sv}$ onto the connecting surface. Thus, the geodesic distance of s and v is $|\overrightarrow{sq} + \overrightarrow{qv}|$ and follows the intrinsic geometry between s and v. This is computationally expensive and introduces many new intermediate points which usually neither correspond to vertices in $\mathcal{M}$, nor reoccur for multiple independent distance measures. The summation of existing edges in $\mathcal{M}$ on the path from s to v as shown in Figure 3.5C, is a good trade off between simple Euclidean measurements and geodesic ones.

Region growth based on Euclidean distance and based on edge sum have been implemented. The concatenation of existing edges was also chosen over the correct topological measure via geodesic distance since it does not need a second processing step to identify traversed triangles and associated edges in $\mathcal{M}$.

Data: Mesh $\mathcal{M}$, seed vertex s, distance *maxLength*
Result: patch $\mathcal{P}$ with vertices connected to s in distance $\leq$ *maxLength*
growthEuclidean *(v,d,$\mathcal{P}$)*

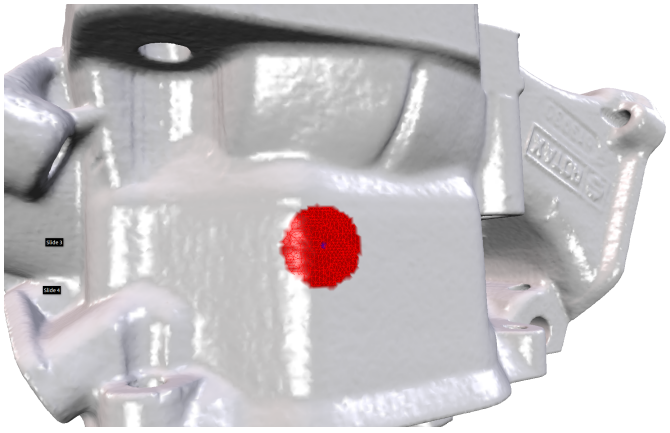
```
for edge e ∈ edgeConnectedTo(currentSeed) do
    vertex v = toVertex(e);
    distance d = getDist(v,currentSeed);
    if P.contains(v) then
        if d ≤ maxLength then
            P.add(v);
            growthEuclidean(v,maxLength − d,P)
        end
    end
end
```

Algorithm 5: Recursive region growth with euclidean distance measure.

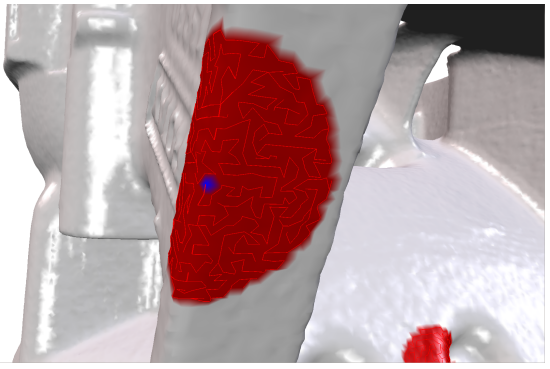
So for distinguishing holes from fragments, a threshold is set as targetted surface area, and the resulting selection is checked if the target was met. Starting at a vertex of a border cycle, the patch either stops growing because the threshold is reached, meaning

that the border belongs to a sufficiently large surface part. Or if all connected components to the border vertex were included in the patch before the threshold was reached, the patch is considered to represent a small fragment and usually deleted.

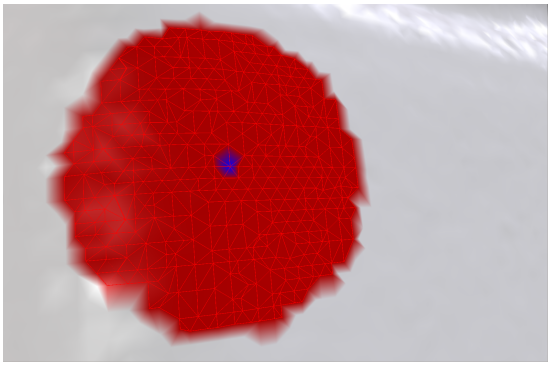
Of course this approach can be applied to any vertex in the mesh, if one wants to select connected components of a given size. We refer to this as *Stencil selection*. The expected output is that neighborhood relations and connectivity within the patch are preserved and that the growth region is expanded equidistantly from the seed vertex. Thus, the growth follows a circular or a spherical shape as shown in Figure 3.6, instead of randomly selecting any connected triangle until the given threshold is reached.



(A) Selected patch of mesh depicted in red.



(B) Patch with highlighted edges that form a tree



(C) Patch with highlighted edges as full subgraph

FIGURE 3.6: Several stencil selection patches with blue seed vertex in the center and red growth area around it.

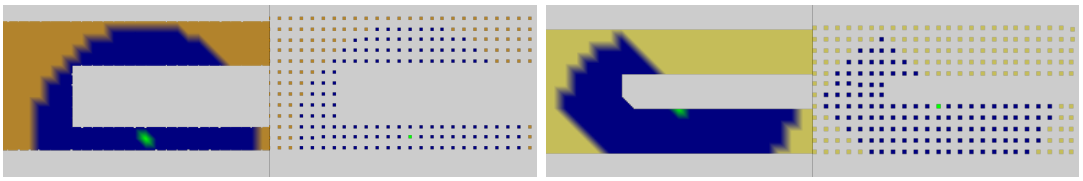
Figure 3.6C shows the result of recursively circulating the one-ring-neighborhood around the borders or the current patch. For each added vertex, all its connected vertices are tested if they are within the distance as defined by the threshold. In this case they become the newly added vertices and their neighborhood is circulated. This approach provides—besides all vertices in the patch—the complete subgraph of edges in mesh $\mathcal{M}$ which correspond the the selected surface part as all traversed edges are added.

Algorithm 5 describes the implementation of this variant, which is much faster than computing the edge-tree shown in Figure 3.6B.

Data: Mesh $\mathcal{M}$, seed vertex s, distance *maxLength*
Result: patch $\mathcal{P}$ with vertices connected to s in distance $\leq$ *maxLength*, edge set $\mathcal{E}$
growthEdgeSum *(v,d,$\mathcal{P}$,$\mathcal{E}$)*
  for *edge e ∈ edgeConnectedTo(currentSeed)* **do**
    vertex v = toVertex(e);
    distance d = getDist(v,*originalSeed*);
    if $\mathcal{P}$.*contains(e)* **then**
      if $d \leq$ *maxLength* **then**
        $\mathcal{E}$.add(e);
        $\mathcal{P}$.add(v);
        `growthEdgeSum(`v,*maxLength*,$\mathcal{P}$,$\mathcal{E}$`)`
      end
    end
  end

Algorithm 6: Recursive region growth with edge summation as distance measure.

For the edge-tree, no valid edge in the one-ring-neighborhood is added. Instead each recursion updates a map of already included edges and allows only those to be added, which contribute a new vertex to the patch. While the distance measure in Algorithm 5 compares the distance from the original seed vertex s to the newly added vertex v, the distance measure via edge summation reduces the threshold with each recursion by the length of the last edge and therefore respects local topology. The implementation of region growth via edge summation is listed as Algorithm 6. The difference in distance measures is especially noticed near elongated gaps or trenches in the mesh, as seen in Figure 3.7. For Euclidean measure, the growth expands across the gap if only a single edge allows circumventing it within the radius of the patch. For edge summation, the distance to circumvent the gap is taken into account and not all vertices within the radius of s are selected. Thus, the selection with Euclidean distance preserves circular shape in



(A) Triangles Euclidean (B) Vertices Euclidean (C) Triangles EdgeSum (D) Vertices EdgeSum

FIGURE 3.7: Stencil selection with seed vertex labelled in green and growth area indicated in blue on an orange 2D surface.

Figure 3.7A (or spherical in 3D) where each vertex v is at max in distance d of the seed vertex s and has a known path connecting s and v within the patch. The selection via edge summation also has roughly circular shape, as shown in Figure 3.7C, but each vertex

v has a known path of length $\leq d$ within the patch to the seed vertex s. Of course an alternative solution to obtain a similar result is to implement Dijkstra's algorithm [Dij59] for finding the shortest paths starting from s to all vertices v. This would require the identification of a suitable subgraph since applying Dijkstra to $\mathcal{M}$ just to analyze a $5mm$ patch in a $1500cm^2$ surface requires considerable extra effort. Furthermore, our main motivation for geodesic region growth as presented in Section F.5.2 would have implied hundreds of thousands of Stencil selections but turned out to be a dead end. Also, the applications *small particle identification* and *border trimming* as presented in Section 3.2.5 are already work sufficiently well with Euclidean region growth.

3.2 Isosurface Extraction

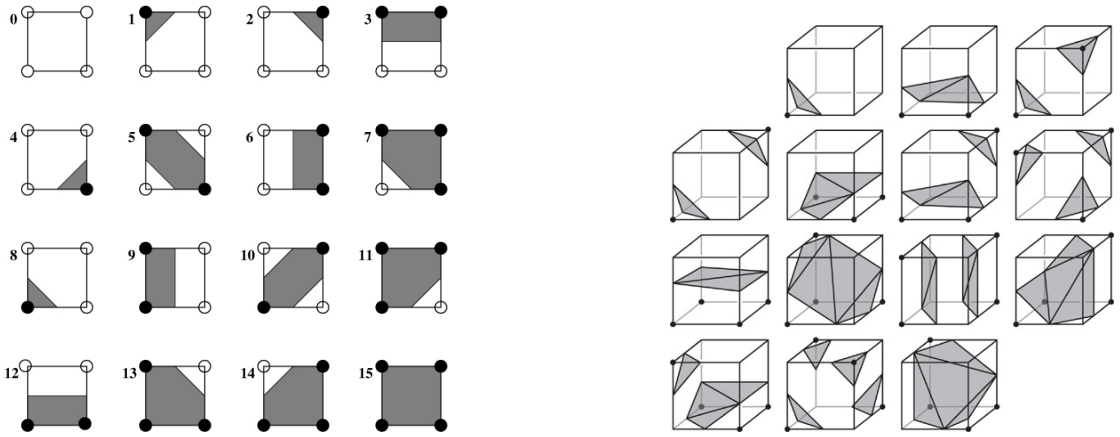
In general, an isosurface is the pre-image of a given threshold in a volume data set. It serves as boundary or segmentation to separate those components where values satisfy the threshold criteria from the rest. In our context, isosurfaces describe the interface between two density regions in the volumetric presentation from CT reconstruction. Usually the threshold is set such that is separates material from air since all our samples are made from homogenous material. At least in theory the scalar field should only contain values corresponding to air and values corresponding to the specimen. In practice this is not the case since the CT does not provide infinite resolution and therefore some voxels in the reconstruction volume contain material and air alike. Also, support material, the rotation stage, deposits and residue are included in the acquisition. Furthermore, all kinds of defects and artifacts may alter the represented attenuation values from the real values.

3.2.1 Marching Cubes

The Marching Cubes algorithm (MC) [LC87] triangulates a scalar field. A cube in the case of 3D scalar fields (or a square in 2D case) is shifted over the volume under investigation. The size of the cube commonly matches the grid size such that any corner corresponds to the center of a voxel in the reconstruction volume. According to the density values on all corners of the cube a template for triangulation of the respective area is applied. The template is chosen based on the state of the corner, which is defined by the given threshold. For the 2D case, if all corners are below the threshold, this corresponds to template (0) in Figure 3.8A, and if all are above this corresponds to template (15). Each template represents a surface configuration corresponding to the observations on the corners of the square.

In principle, the cube or square moves through the scalar field and locally applies the appropriate template to generate local surface fragments. This provides a closed surface after all positions have been triangulated. As the density values are discrete, the location matching the threshold between two scalar values is interpolated and the coordinates of the applied triangles adjusted to seamlessly adjoin to a watertight surface. Eventually several unconnected surface parts are generated, but each surface part represents a watertight mesh, i.e., a manifold of lower dimensionality than the processed field.

Many implementations of the marching cubes algorithm exist, also variants such as marching tetrahedra or marching diamonds. They mainly differ in choice of templates, the treatment of special cases or the ability to evaluate multiple locations in parallel.



(A) 16 templates for 2D (B) 3D templates (plus empty)

FIGURE 3.8: Marching Cubes templates.

The marching cubes approach is very intuitive and great to explain the concept of isosurface generation. Yet, for a practically applicable algorithm the special case treatment and variety of templates makes it a very demanding implementation. For the 2D case 16 templates were listed. For the 3D case $2^8 = 256$ templates exist which due to rotation or inversion can be reduced to the 15 effectively distinguishable patterns as shown in Figure 3.8B. Special cases and further considerations require again to add certain patterns, e.g., to be applied at the external borders of the scalar field, so common MC implementations rely on 27 templates.

3.2.2 Volume-Enclosing Surface Extraction

Volume-Enclosing Surface Extraction (VESTA) [Sch12] presents an alternative approach to Marching Cubes algorithms. In short, this is isosurface identification without templates or 2x2x2 voxel comparison.

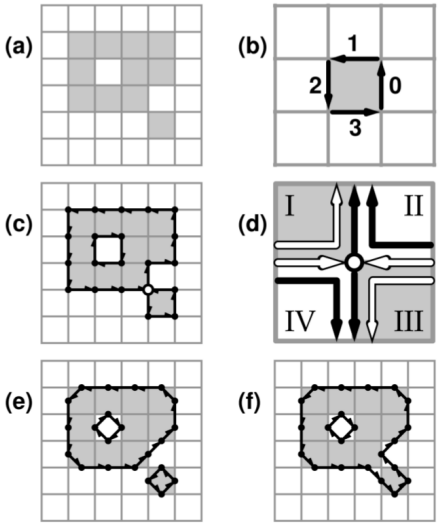


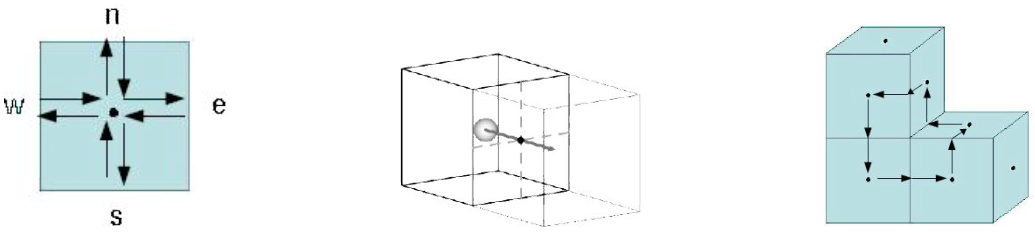
FIGURE 3.9: Cells with a density value above threshold in grey and with lower density in white.

The VESTA approach—in the simpler 2D case, shown in Figure 3.9—assigns active and inactive cells as shown in example (a) of Figure 3.9. If adjacent cells have same configuration, e.g., both are inactive or both are active, no action is taken. If they have different configuration, the joint border is marked as a directed edge, such that the active cell is on the left side of the border in direction of the edge. In the next step those vectors are connected to paths, as in (c), which in 2D is trivial. If in one intersection more than two edges meet, then there exist multiple solutions to form cycles as in (d). Disambiguation for those Points Of Ambiguity (POA) is either achieved by separating the intersecting paths as in (e) or by joining them as in (f). In general all adjacent vectors with a 90° turn are replaced by connecting their centers such that two 45° turn provide smoother edges. The separation or union of cells at POAs is referred to as operating in *disconnected mode* or *connected mode*.

The VESTA algorithm triangulates a 3D scalar field of density values in a similar fashion. As an additional step, the calculating appropriate Face Center Points (FCP) needs to be performed. Those represent the same interpolation step as for the Marching Cubes approach and mark the approximate location the threshold between two voxel centers. Following this, cycles are formed following a simple set of rules and finally the cycles are triangulated. This approach does not rely on templates since they are implicitly generated during creation and triangulation of the cycles. Also the treatment of POAs—which refers to special cases in MC—is intuitive and allows for further customization.

In 3D, the FCP identification and the forming of directed edges are separate steps. The cubes in Figure 3.10 represent the voxels of the scalar field and simplified FCPs are denoted by the black dots. The movement pattern can be roughly described as *take*

a left but depends on the orientation of the cell side it is applied to as illustrated in Figure 3.10A. The complete cycle of five directed edges is shown in Figure 3.10C.



(A) Directions - *take a left*

(B) Face Center Point between voxel centers

(C) Cycle from directed edges

FIGURE 3.10: Vesta - 3D

3.2.3 Triangulation

Starting from original FCP the remaining directions need to be processed. All resulting cycles are shown in Figure 3.11A including each resulting triangulation. During the forming of cycles the information about which direction was already processed is stored for each FCP. If all directions for all FCP were processed the triangulation is finished as shown in Figure 3.11B.

(A) All cycles at FCP

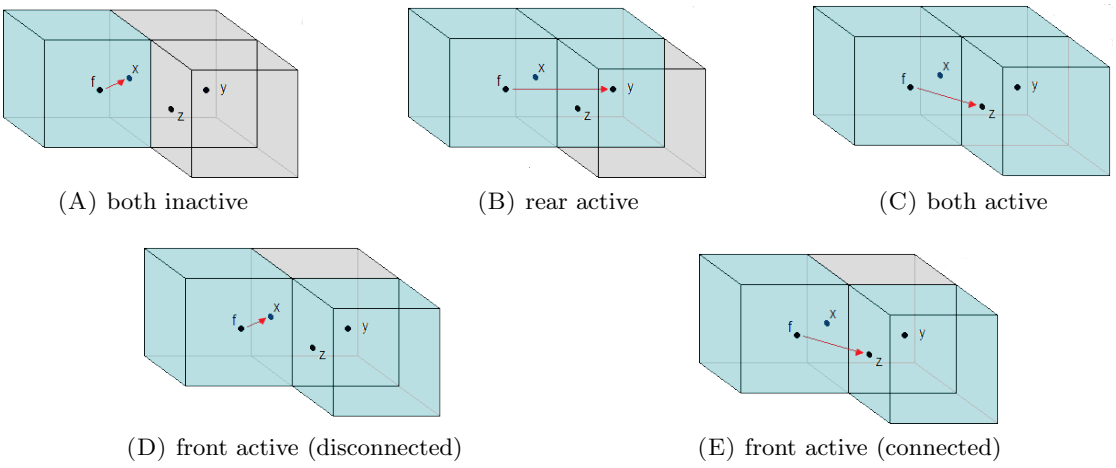
(B) Resulting surface parts

FIGURE 3.11: Vesta cycles (left) and resulting triangulation (right).

3.2.4 (Dis)connected Mode

In contrast to MC, special cases are handled by a simple set of rules for the treatment of POAs. A configuration of three voxels is shown in Figure 3.12 with an ambiguous case in the bottom row where two FCPs are valid targets.

Both solutions in the bottom row of Figure 3.12 are valid and called *connected mode* or *disconnected mode*. The preferred strategy can be applied by default to all POAs or an evaluation of density values in the surrounding is made for case-by-case decisions. The resulting surface for a similar case is shown in Figure 3.13.



(A) both inactive (B) rear active (C) both active

(D) front active (disconnected) (E) front active (connected)

FIGURE 3.12: Starting from FCP f seletion of next FCP x, y or z based on status (red=inactive) of next voxels. Standard cases on top row, ambiguous case in bottom row.

(A) Connected mode (B) Disconnected mode

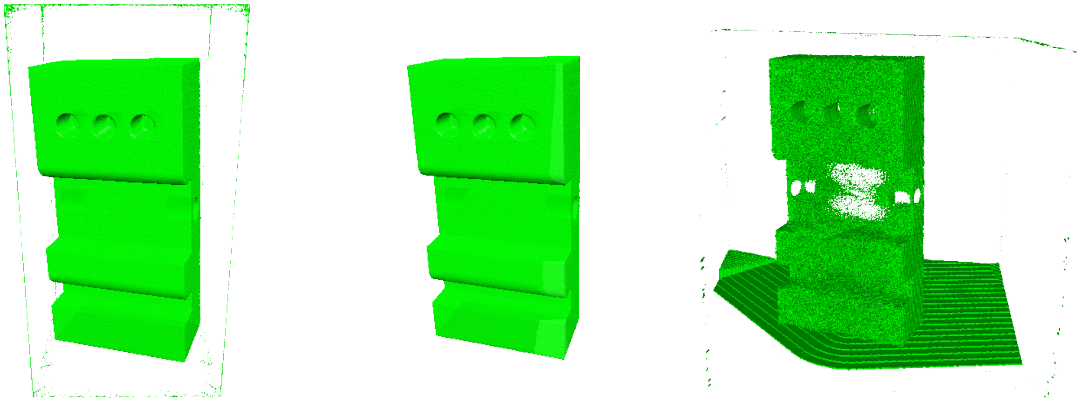
FIGURE 3.13: Resulting surface for connected and disconnected mode for an identical voxel configuration.

To decide between disconnected mode (a) and connected mode (b) the average value of FCPs surrounding the POA is compared to the threshold.

3.2.5 Border Trimming

Applying isosurface extraction to volumetric representations requires setting a suitable threshold, which can easily achieved by identifying the appropriate interface in the histogram. As described in Section 2.2.2 various sources of artifacts contribute defects to the volumetric data and therefore to the extracted isosurface. An example of reconstructed artifacts is shown in Figure 3.14. The tiny fragments that occur at the outer borders of the reconstruction volume are present in all volumetric data sets presented in Section 2.5.1. They can be removed either by clipping a thin layer from each side of the volume, which is referred as *border trimming* or by applying a region growth algorithm as in Section 3.1.3 to crop the largest connected component. While border trimming can always be applied but rarely removes all particles, cropping the largest fragment is not suitable for all cases. A successful removal of all small particles as in Figure 3.14B

does not work for isosurfaces that contain multiple small components, which obviously belong to the object or isosurfaces in which the largest component represents the rotation stage or support material. Both is the case in Figure 3.14C which prohibits region growth and rather requires a more sophisticated treatment like introducing a second density threshold, e.g., to neglect air and steel while preserving voxels that correspond to aluminum.



(A) Full volume (B) Trimmed border (C) LACT volume

FIGURE 3.14: Isosurfaces before and after post-processing.

3.3 Strategies for Estimating the Outer Dimensions of an Object

Many approaches are known to estimate the outer dimensions of the mesh representation of an object; we present them according to the level of detail they provide. All of them have been implemented and the relevant ones are investigated in Section 5.1.4.

3.3.1 Axis-Aligned Bounding Box

The axis-aligned bounding box (AABB), the simplest estimation of object dimensions, is computed from component wise comparison of all vertex coordinates. The two points p_{min} and p_{max} defining the enclosing cuboid are calculated as

$$p_{min} = (\min(x), \min(y), \min(z)), \quad p_{max} = (\max(x), \max(y), \max(z)) \tag{3.1}$$

In the 2D case, the resulting axis-aligned box has the corner vertices $(\min(x), \min(y))$, $(\min(x), \max(y))$, $(\max(x), \min(y))$ and $(\max(x), \max(y))$.

3.3.2 Minimum Volume Bounding Box

The Minimal Volume Bounding Box (MVBB) algorithm [HP01] calculates the exact diameter of a point set. All points—or a subset—are ordered in a fair-split tree. From the two vertices responsible for the largest extent and their location the tree structure, an arbitrarily oriented box can be computed. The accuracy or tightness of the resulting MVBB depends on the size of the subset.

3.3.3 Convex Hull

The Convex Hull (CH) is the smallest convex set of vertices of an object that contains the object itself. It is an even better estimation of the object's dimensions than MVEE and usually is the basis of calculating MVEE, since it reduces the problem size drastically. However, it lacks a parametric form (Figure 3.15B).

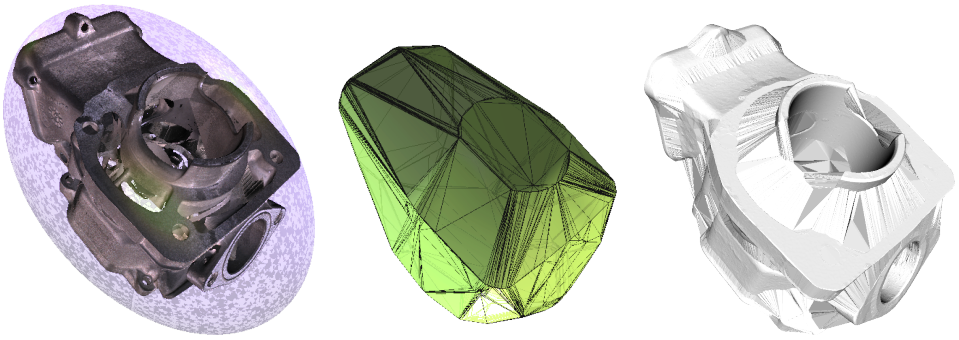
3.3.4 Minimum Volume Enclosing Ellipsoid

The Minimal Volume Enclosing Ellipsoid (MVEE) is an oriented ellipsoid with nine degrees of freedom, i.e., xyz-position of the center, orientation of the three perpendicular axes and the three radii along these axes. The implementation based on Todd et al. [TY07] computes a parametric form of an enclosing primitive around the object. If the object is not already known to be roughly cuboid, this presents a better estimation of the dimensions in the general case (Figure 3.15A). As the computation optimizes the parameters such that volume and outlier distance are minimal, the convex hull is computed beforehand. Of course calculating the enclosed volume is trivial, but checking for each vertex in $\mathcal{M}$ if it is outside the ellipsoid is very costly. Reducing $\mathcal{M}$ to the more relevant set $\mathcal{M}_{\mathrm{CH}}$ decreases processing speed drasticly [Mos05].

3.3.5 Alpha Shapes

Alpha Shapes (AS) define a shape around the object, but this shape does not need to be convex. So far, it is the best approximation of the object's dimensions and commonly compared to shrink-wrapping or gift-wrapping an object. The Delaunay triangulation of all object vertices [Joe91] provides a basis to compute the α-complex [EM94] and in turn the α-shape as shown in Figure 3.15C. Depending on the chosen α-value, the surface varies, i.e., the value defines how tight $\mathcal{M}_{\mathrm{AS}}$ approximates the input mesh. We choose α such that the tightest hull that still produces a single connected component

is computed. Any deviation results in $\mathcal{M}_{\text{AS}}$ either loosely fitting the input mesh, or containing several unconnected surface parts.



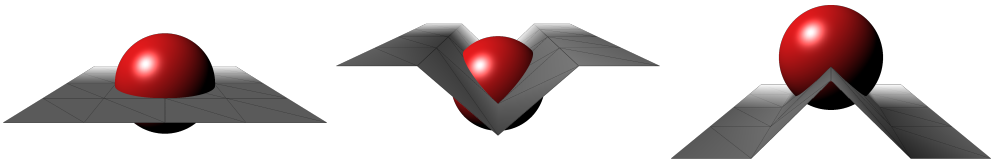
(A) MVEE (B) Convex Hull (C) Alpha Shape

FIGURE 3.15: Different hulls around $\mathcal{M}_{\text{Opt}}$.

3.4 Curvature Analysis via Multi-Scale Integral Invariants

Drift errors in alignments were presented in Section 1.5 and an approach for mitigating them is introduced in Section 4.1. For the purpose of evaluating that approach, we need to analyze the curvature of different samples and shapes. A suitable method to do so, is an analysis based on Multi-Scale Integral Invariants. An algorithm implementing this concept was developed by our colleague Hubert Mara [Mar12].

Multi-Scale Integral Invariants (MSII) are computed from the intersection of the surface $\mathcal{M}$ and a set of n isocentric spheres with different radii, i.e., scales. The analysis is performed for each vertex v of the mesh, i.e., while each v defines the center of the nested spheres. The largest sphere S_0 has a radius r_0 depending on the size of the desired features. For the nested spheres $S_1, \ldots, S_{n-1}$ the radii are equidistantly chosen such that radius r_x of each sphere equals $r_x = r_0 - x\frac{r_0}{n}$.

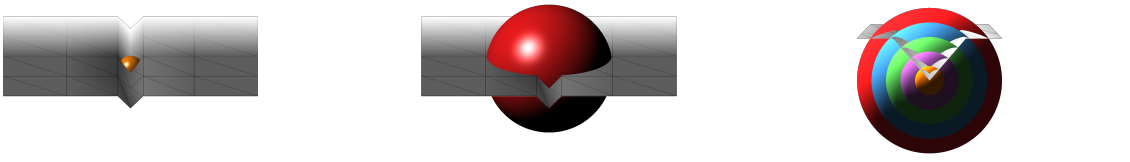


50% → flat 75% → concave 25% → convex

FIGURE 3.16: Sphere volume below surface and implication.

A 16D feature vector $\mathcal{F}_v$ holding the results per sphere is computed per vertex $v \in \mathcal{M}$. The output of these computations are in the range $\left]0, \frac{4}{3}\pi r^3\right[$ for the analysis based on

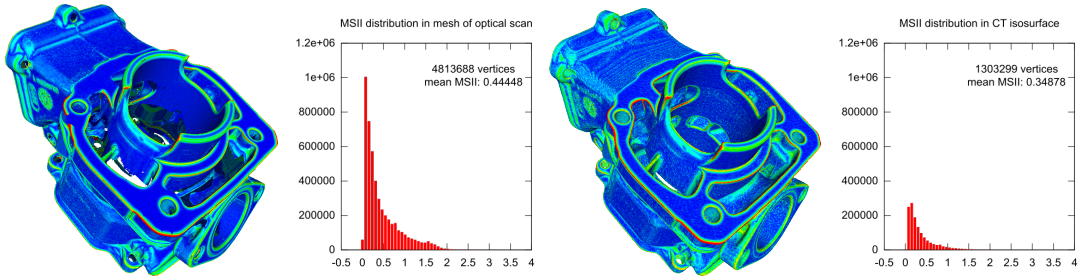
enclosed volume and $]0, +\infty[$ for the analysis based on enclosed surface. After normalization for each radius, the feature vector contains entries in the range $]0, +\infty[$ for the enclosed surface and $]0, 1[$ for the enclosed volume. MSII provides invariant curvature information on various scales, i.e., it provides robust results for different resolution levels of the mesh. Therefore, it is highly suitable for analyzing the very same object represented as $\mathcal{M}_{\text{CT}}$ in lower resolution and $\mathcal{M}_{\text{Opt}}$ in higher resolution while computing comparable feature vectors as shown in Figures 3.18A and 3.18C.



55% → concave 75% → concave $\mathcal{F}_v = \{75\%, 75\%, 75\%, 74\%, 72\%\}$

FIGURE 3.17: Intersection of surface with small, large and nested spheres.

In our case, $n = 16$ spheres are computed, which is heuristically a good trade-off between accuracy and performance. Two variants are implemented for the analysis, computing either (a) the fraction of the volume of S_x and the enclosed volume as intersection of S_x and the volume below the intersected surface area of the mesh, or (b) the fraction of the surface of a disc with the radius r_x and the surface area of the intersection of mesh and sphere.



(A) MSII for $\mathcal{M}_{\text{Opt}}$ (B) Histogram for $\mathcal{M}_{\text{Opt}}$ (C) MSII for $\mathcal{M}_{\text{CT}}$ (D) Histogram for $\mathcal{M}_{\text{CT}}$

FIGURE 3.18: Curvature visualized via Euclidean distance of MSII feature vectors. Histograms list MSII score per vertex with a high fraction of scores near zero indicating flat regions.

The evaluation of surface extraction methods uses the MSII analysis based on intersected volume since it is closer related to Gaussian curvature and conveniently provides results in the range $]0, 1[$. The intersected surface parts estimate mean curvature and provide results in a range not suitable for our analysis in Section 5.1.

3.5 Alignment Algorithms

A common problem when working with data acquired from different sources or during independent acquisitions is to establish a relation between them. The partial acquisitions from optical scans as described in Section 1.2.1 for example do not contain the information how the single surface fragments need to be joined to form the full scan result. In general, alignment algorithms solve the problem of finding a suitable change of basis in the given vector space. Related to 3D data, the transformation from one Cartesian coordinate system to another is provided. The acquisition system denotes measurements according to a Cartesian coordinate system relative to the camera position. Altering the camera position or the specimen orientation then requires the computation for the corresponding coordinate transformation which is represented as a 4x4 affine transformation matrix—or in short referred as *the transformation*. This transformation is what alignment algorithms provide by *point-to-point* comparisons between the respective meshes.

3.5.1 RANSAC

RANdom Sample And Consensus (RANSAC) schemes generate various hypotheses and verify or falsify those hypotheses based on random sample surveys. The RANSAC concept is not limited to alignment algorithms and can be applied to a variety of fields. For our case the implementation follows Winkelbach et al. [WMW06] and associates certain properties to a given pair of points from one mesh. Those properties are supposed to characterize the relation of both points. Identifying a pair of points with similar properties in a different mesh, serves as basis for a hypothesis. Thus, the hypothesis is that the pairs correspond to each other. Testing the hypothesis is provided by computing a transformation matrix from one pair to the other. Evaluation of this test is achieved by applying the resulting matrix to a random set of points from the first mesh and measuring the distance from their new location to the destination mesh. Proximity to the destination mesh indicates a good hypothesis, large distances suggest rejection and motivate the generation of a new hypothesis. Point pairs are selected randomly within a mesh and after every chosen pair—including computation of the properties—all properties ever evaluated in the other mesh are checked for correspondence. This cycle of selecting points, computing properties and checking for correspondence is continued in an alternating manner until a hypothesis can be formulated and continues after a hypotheses was rejected. The acceptance of a hypothesis ends this cycle as a transformation could be validated and is considered to be correct. Also exceeding a given maximum of allowed cycles might end the algorithm in which case the best known, yet no sufficient transformation is returned.

To give an example, let us assume we want to align mesh $\mathcal{M}_{\text{Opt}}$ and isosurface $\mathcal{M}_{\text{CT}}$. We select a vertex pair $(o_1, o_2) \in \mathcal{M}_{\text{Opt}}$. A 4D-vector p_{Opt} characterizes those vertices by their properties. The properties are computed from the vector $\overrightarrow{o_1 o_2}$, the normal vector n_1 of vertex o_1, and n_2 of vertex o_2. The four components of $p_{\text{Opt}} = \text{prop}(o_1, o_2)$ are:

1. the length $||\overrightarrow{o_1 o_2}||$,
2. the rotation angle between n_1 and n_2 around $\overrightarrow{o_1 o_2}$,
3. the inclination angle between $\overrightarrow{o_1 o_2}$ and n_1 around $\overrightarrow{o_1 o_2} \times n_1$,
4. the inclination angle between $\overrightarrow{o_1 o_2}$ and n_2 around $\overrightarrow{o_1 o_2} \times n_2$.

For each iteration a vertex pair $(o_1, o_2) \in \mathcal{M}_{\text{Opt}}$ is selected, $p_{\text{Opt}} = \text{prop}(o_1, o_2)$ searched in all previously processed p_{CT}. If no match is found, a new iteration starts with randomly selecting a vertex pair $(c_1, c_2) \in \mathcal{M}_{\text{CT}}$, searching $p_{\text{CT}} = \text{prop}(c_1, c_2)$ in all previously processed p_{Opt} and continues alternatingly. If the current property vector matches any previously computed vector from the other mesh, a transformation is computed. The resulting transformation T from $\mathcal{M}_{\text{Opt}}$ to $\mathcal{M}_{\text{CT}}$ can be computed by centering $\mathcal{M}_{\text{Opt}}$ in o_1 and $\mathcal{M}_{\text{CT}}$ in c_1 before rotating both such that the normal or the first point is at $(1, 0, 0)$ and the second point of the pair is on the y-axes. Then the transformation matrix T is

$$T = T_{\text{Opt}} T_{\text{CT}}^{-1}. \tag{3.2}$$

3.5.2 Iterative Closest Point

Alignment of meshes representing complete surfaces or surface fragments can be achieved by Iterative Closest Point (ICP) algorithms. This method continuously adjusts a rotation and a translation such that the root mean square error (RMSE) distance between point clouds is minimized [BM92]. The approach converges to a given threshold of largest acceptable error distance or breaks after a number of maximal iterations. However, it might converge in an erroneous local minimum, especially for noisy data. To ensure that the solution found by the ICP algorithm is a global minimum, a simulated annealing algorithm can be applied [PEK+01]. As a special case, ICP can solve certain alignments in a single iteration. This usually holds for perfect matches, e.g., identical meshes, or for incorporating prior knowledge to narrow down the solution space [NDF14]. Our implementation of the ICP algorithm is from ITK[1] [YAL+02] framework with a Levenberg-Marquardt solver to register the meshes fully automatically. A more common scenario is

[1] ITK - Insight Segmentation and Registration Toolkit - `www.itk.org`

to provide an initial setup by manually selecting tie points as user interface interaction, e.g., in MeshLab[2] [CCC+08].

3.6 Providing Prior Knowledge to CT Reconstruction

In order to compute a mask, i.e., provide prior knowledge to the CT reconstruction algorithm, the information contained in the optical mesh $\mathcal{M}_{\text{Opt}}$ needs to be converted in a suitable format. This either happens via Raycasting, by intersecting the aligned $\mathcal{M}_{\text{Opt}}$ with each single ray as it occurs during CT acquisition or by intersecting all voxels in the reconstruction volume with the triangles in $\mathcal{M}_{\text{Opt}}$. Intersection tests answer the question if two given primitives such as two lines in 2D or a line and a plane in 3D overlap. Furthermore, the exact location of intersection or extent of overlap can be provided. Several of those tests are performed to generate a mask.

The Ray-based Mask as the name suggests, involves a lot of intersection tests with rays. Any CT setup has a specific configuration file holding all geometry information such as distances between photon source, sensor array and rotation axis. It lists the elevation of the rotation stage, the direction, inclination and possible offsets of all components. In short, it provides a full specification of the machinery and in addition the trajectory as a sequence of all single projections is described. Given an alignment of $\mathcal{M}_{\text{Opt}}$ to $\mathcal{M}_{\text{CT}}$, the projection positions in the trajectory can be associated with both meshes and it is feasible to repeat the CT scan procedure virtually with $\mathcal{M}_{\text{Opt}}$ as specimen. Each ray as it occurs in the real CT scan is reiterated as a virtual intersection of a line from the source to any of the pixels on the detector plane, and in turn every single projection along the trajectory is processed. In addition to the reconstruction from real projections, distance measurements from virtual intersections describe at which position in the reconstruction volume the object—according to optical scans—begins and where it ends. This information is the prior knowledge a subsequent reconstruction can leverage to avoid errors and misinterpretations of the projection data which led to artifacts in the original reconstruction.

3.6.1 Ray versus AABB Test

Axis Aligned Bounding Box intersections are faster to calculate than all exact ray-vs-triangle intersections. The SLAB Method [KK86] efficiently answers the question as to whether a given ray intersects with an axis-aligned box. “Slabs” in this context refer to the space between two parallel planes. Since the AABB of $\mathcal{M}_{\text{Opt}}$ is known, this allows

[2]MeshLab - `http://www.meshlab.org`

to reject all rays not passing this test as they cannot intersect with the object inside the AABB. The SLAB method considers projections of the 3D intersection problem on the xy, xz and yz planes. So each projection can be treated as three 2D problems and the test checks whether it is possible to fully clip the ray in each of these planes by the four lines representing the AABB min/max extent in this plane. If this succeeds, the 3D ray misses the AABB. Else it needs to be further evaluated since it potentially interacts with $\mathcal{M}_{\text{Opt}}$. The 2D example in Figure 3.19 shows a 2D scenario with one ray intersecting and one ray passing the box.

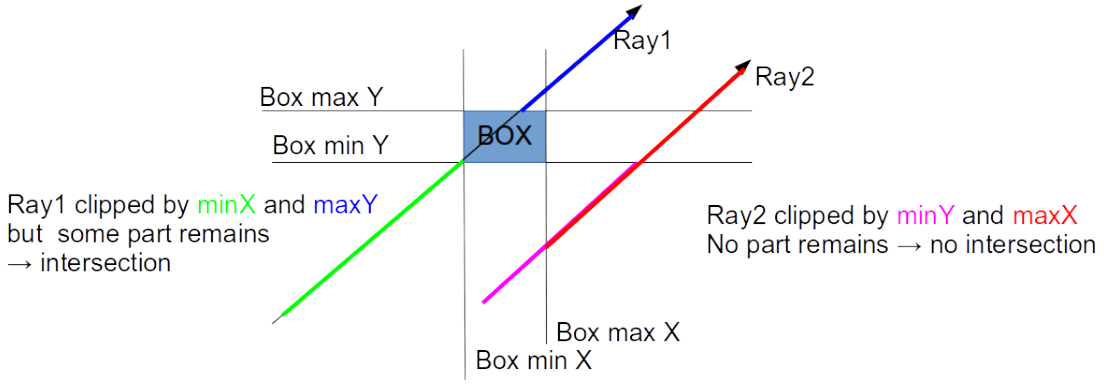


FIGURE 3.19: SLAB method

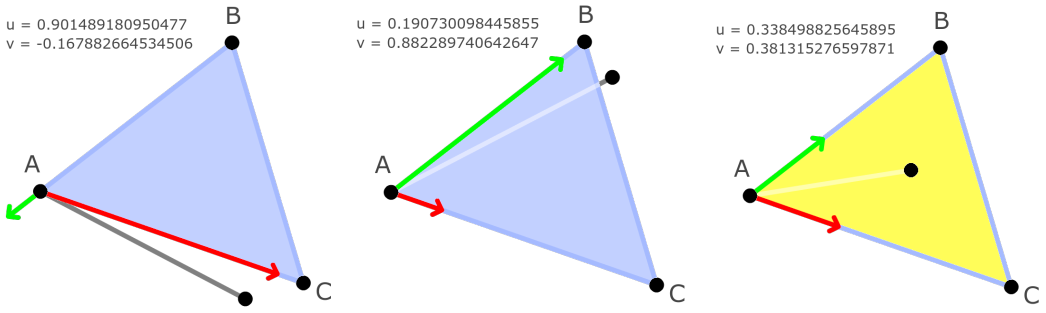
3.6.2 Barycentric Coordinates

If the previous test suggests that a given ray might intersect $\mathcal{M}_{\text{Opt}}$, each triangle in the mesh needs to be evaluated. A common approach is to solve the problem with barycentric coordinates where two edges of the triangle serve as coordinate system. In Figure 3.20 edges $u = \overline{AB}$ and $v = \overline{AC}$ serve as coordinate system. Any point in the triangle can be expressed as a linear combination of both vectors. In fact any point in the triangle plane can be described, but only points inside the triangle area that satisfy $u \geq 0, \quad v \geq 0, \quad u + v \leq 1.$

Barycentric coordinates are widely used in intersection tests since they provide accurate calculation of intersection points within the triangle. They also allow for precomputation and reuse of coordinate axes and determining the direction of the hit by the signed determinant during coordinate transfer to Cartesian coordinates.

3.6.3 Plücker Coordinates

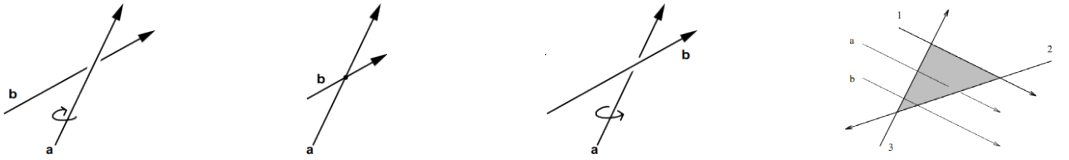
Determining if a ray intersects a triangle without locating the exact penetration point can be achieved via edge wise relative orientation check. Plücker coordinates [TH99] describe each edge—or half-edge—of a triangle and the intersecting ray as a 6D vector.



(A) $(u + v \leq 1$ && $u \geq 0$ && $\mathbf{v \geq 0})$ (B) $(\mathbf{u + v \leq 1}$ && $u \geq 0$ && $v \geq 0)$ (C) $(u + v \leq 1$ && $u \geq 0$ && $v \geq 0)$

FIGURE 3.20: Intersection test with barycentric coordinates. Edges $\overline{AB}$ and $\overline{AC}$ serve as the coordinate system for the triangle $\triangle(ABC)$. Red condition fails intersection test.

So, given two points $P = (P_x, P_y, P_z)$ and $Q = (Q_x, Q_y, Q_z)$ on a line L. Line L from Q to P is expressed in Plücker Coordinates as the pair $L = \{P - Q, P \times Q\} = \{U, V\}$. For two lines $L_1 = \{U_1, V_1\}$ and $L_2 = \{U_2, V_2\}$ the sign of the expression $\text{side}(L_1, L_2) = U_1 \cdot V_2 + U_2 \cdot V_1$ determines relative orientation. In Figure 3.21 the relations of the $\text{side}(L_1, L_2)$ are shown.



$\text{side}(a, b) < 0$ clockwise $\text{side}(a, b) = 0$ intersection $\text{side}(a, b) > 0$ counter clockwise For line b $\text{side}(1, 2) \neq \text{side}(1, 3)$

FIGURE 3.21: Intersection conditions for Plücker coordinates.

Since U and V can be precomputed and stored per edge efficient intersection tests using two to three dot products can be performed. As soon as two dot products return different signs the test fails and does not require further evaluation.

3.6.4 Hardware Accelerated Intersection Tests

All the above concepts for computing intersections have been widely studied and successfully implemented and improved by Möller-Trumbore [MT05], Wald [WPS+03], Badoul [Bad90], Shevtsov [SSKN07], Kensler-Shirley [KS06] and Herout/Have [HH10]. The latter leaverage the "Streaming SIMD Extensions 4" (SSE4) instruction set for calculations and combine advances in reordering the evaluations and precomputing of four dot products and an inverse determinant. Figure 3.22 depicts the SSE4 specialized

variant described by halfplanes N_1 an N_2 in addition to the triangle plane N as

$$\vec{N_1} = \frac{\vec{AC} \times \vec{N}}{|\vec{N}|^2}, \quad d_1 = -\vec{N_1} \cdot A \quad \text{and} \quad \vec{N_2} = \frac{\vec{N} \times \vec{AB}}{|\vec{N}|^2}, \quad d_2 = -\vec{N_2} \cdot A \tag{3.3}$$

The barycentric coordinates of the intersection point are

$$P = O + t\vec{D}, \quad u = \vec{N_1}, \cdot P + d_1 \quad v = \vec{N_2} \cdot P + d_2 \tag{3.4}$$

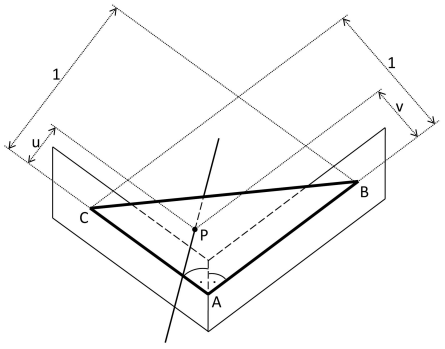


FIGURE 3.22: Triangle planes for SSE4 processing of intersections.

The Volumetric Mask follows a different approach than the ray-based mask. It provides similar information but is much easier to compute. The configuration file of the CT setup is not needed, instead all parameters can be extracted from the volumetric representation. As $\mathcal{M}_{\text{Opt}}$ is aligned to $\mathcal{M}_{\text{CT}}$, the same alignment holds for the volumetric presentation, and therefore the reconstruction volume for which dimensions and spacing are known. Intersecting all voxels of the reconstruction volume with the triangles in $\mathcal{M}_{\text{Opt}}$ tags them as boundary voxels. For a subsequent reconstruction the constraint based on $\mathcal{M}_{\text{Opt}}$ is that attenuation values outside tagged areas must not exceed the density of air. Of course this approach can take into account that $\mathcal{M}_{\text{Opt}}$ might have been post-processed to provide a watertight mesh. Therefore, the mask can also contain which voxels are tagged by measured triangles and which are tagged by artificial patches.

3.6.5 AABB versus Triangle Test

Algorithms to evaluate box-vs-triangle intersection tests[AM05] usually assume an AABB as box. To generate such for any triangle is trivial, also to expand it to the spacing of the volumetric representation. If the box containing the triangle now fits exactly one voxel, no test needs to be performed since the voxel can directly be tagged. In case the

AABB spans over several voxels, each voxel needs to be tested for intersection with the triangle.

Chapter 4

Developed Methods and Implementations

This chapter explains the concepts and methods developed in the context of this work. The main focus is on solving the alignment problem as stated in Section 1.5 since traditional schemes as in Section 3.5 do not provide a suitable solution. Mitigation of their alignment errors can be achieved by reducing the input meshes to the corresponding (exterior) surface parts. The following introduces *RanCEAF - Random Convex-Edge Affine Features* extraction as proposed method to achieve this. An evaluation against the methods presented in Section 3.3 is provided in the next chapter. Furthermore, the alignment scheme *OctaCog*, tailored for processing those surface subsets and omitting the errors of traditional schemes, is presented.

In principal, the data fusion of optical scans and Computed Tomography can be described as providing prior knowledge to the CT reconstruction algorithm. Artifacts occur during reconstruction from distributing attenuated energy outside the object itself, which we call "false positives". The measurement from optical scanning permits segmentation of the reconstruction volume such that the volume between object and optical acquisition system is considered air and communicated to the reconstruction algorithm as "true negative". Provided the alignment of both imaging systems is accurate, this enhances the reconstruction result by suppressing those artifacts. Of course the opposite holds if the alignment is flawed. In this case distribution of attenuated energy within the object would be restricted and distribution outside the object permitted. Thus, a proper alignment of both data sets is crucial for data fusion.

4.1 Random Convex-Edge Affine Feature

Random Convex-Edge Affine Feature (RanCEAF) selection of surface points is performed via nearest neighbor search. The seeds of those queries are randomly distributed on an enclosing ellipsoid around the mesh $\mathcal{M}$ as described in Section 3.3.4. Thus, the seeds are guaranteed to be above the mesh itself and unrelated to the resolution of the underlying mesh.

4.1.1 Regional Queries

Nearest neighbor searches (NNS) can be efficiently carried out by a suitable data structure, e.g., a k-d tree storing all vertices of the mesh under investigation. The seed vertex s of our query is above the exterior surface and the nearest neighbor v is chosen as:

$$v \in \mathcal{M} \quad \text{s.t.} \quad \|v - s\|_2 = \min_{p \in \mathcal{M}} \left(\|p - s\|_2 \right). \tag{4.1}$$

Thus, v is the one vertex from the mesh, which is closest to the seed vertex s, and it is also ensured that v is not below the exterior surface.

4.1.2 Randomized Seed Distribution

A randomized distribution for seeds s is generated via spherical coordinates θ and ϕ. A Mersenne Twister [MN98] pseudo-random generator of 32-bit numbers with a state size of 19937 bits is employed to provide a uniform distribution of $u, v \in [0, 1]$, with

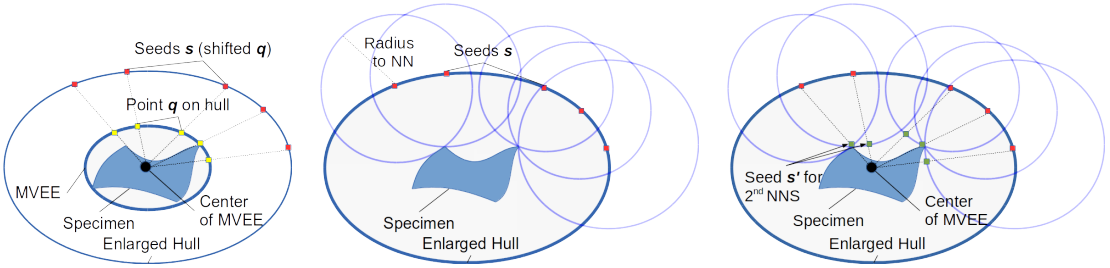
$$\theta = 2\pi u \quad \text{and} \quad \phi = \cos^{-1}(2v - 1). \tag{4.2}$$

In combination with a given radius r, the relation of spherical coordinates and Cartesian coordinates is established. In case of $r = 1$, the distribution contains points on a unit sphere such that any small area on the sphere is expected to hold the same number of points.[2] Let MVEE be described by its center c_{MVEE}, perpendicular axes a_1, a_2, a_3, and the respective radii r_1, r_2, r_3, which are derived from an eigenvalue decomposition to get a parametric form [TY07]. The xyz-coordinates of a point $q' = (q'_x, q'_y, q'_z)$ are based on θ and ϕ as follows:

$$q'_x = r_1 \sin(\theta) \cos(\phi), \quad q'_y = r_2 \sin(\theta) \sin(\phi), \quad q'_z = r_3 \cos(\theta). \tag{4.3}$$

[2] Eric W. Weisstein, MathWorld: `http://mathworld.wolfram.com/SpherePointPicking.html`

This formulation respects the radii of the ellipsoid but not its orientation and location, all points q' in Equation (4.3) are located on an axis-aligned ellipsoid centered at the origin of the Cartesian coordinate system. A transformation t given by a 4×4 matrix A_t is computed from a rotation to axes a_1, a_2, a_3 and the translation to the center c_{MVEE} of the MVEE. Thus, after applying Equation (4.3), any point q' is transformed by A_t to its final position q on the surface of the MVEE around $\mathcal{M}$.



(A) Elevation of q to s

(B) NNS from seeds s (red)

(C) NNS from seed s' (green)

FIGURE 4.1: NNS without and with shifted seed vertices (2D example).

4.1.3 Random Points on Hull

For RanCEAF approach, a random distribution of seed vertices across the whole surface of the ellipsoid is computed. For simplicity, random unit vectors are generated in the origin and transformed to the surface of the ellipsoid. The transformation is according to length and direction of the principal axes of the ellipsoid and its center.

4.1.4 Random Points on a Unit Sphere

For the random distribution of unit vectors several methods [Mar72], [Mul59], [Coo57] have been evaluated. The implemented method generates a uniform real distribution of double precision values in the range $(0.0, 1.0)$ via a Mersenne Twister. We chose u and v to be random variates on this interval for calculating spherical coordinates

$$\phi = 2\pi u \quad \text{and} \quad \theta = cos^{-1}(2v - 1). \tag{4.4}$$

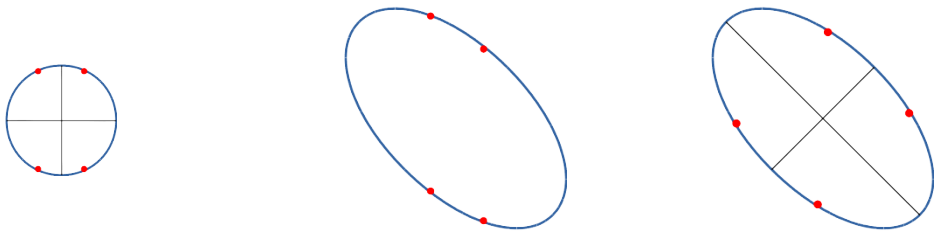
Conversion from spherical coordinates to Cartesian coordinates of point $\mathbf{p} = (x, y, z)$ on the unit sphere follows as usual

$$x = \sin(\theta)\cos(\phi), \quad y = \sin(\theta)\sin(\phi), \quad z = \cos(\theta). \tag{4.5}$$

The randomness of the generated unit vectors is analyzed via local linearity measure l of the resulting point cloud. Local linearity, as presented in Section 2.4.5, measures how well the points are distributed on the sphere. In contrast to choosing $u, v \in (0, 1)$, as suggested in literature, we included the boundaries since this yields better local linearity measures. For a detailed evaluation of boundaries, see Section B.

4.1.5 Symmetrical Distribution of Seed Vertices

In combination with OctaCoG alignment, presented in the next Section, the RanCEAF subsets must obey certain rules to preserve symmetry. The presented generation of seed vertices ensures uniform distribution, but this only holds if the number of seeds is sufficiently large. To avoid introducing any imbalance or misrepresentation, an OctaCoG specific RanCEAF implementation preserves the center of gravity and the symmetry of octant barycenters during seed generation. For each generated point (p_x, p_y, p_z) on the unit sphere, seven additional vertices are added in the other octants as reflections across all coordinate axes: (p_x, p_y, p_z), $(-p_x, p_y, p_z)$, $(-p_x, -p_y, p_z)$, $(-p_x, -p_y, -p_z)$, $(p_x, -p_y, -p_z)$, $(p_x, p_y, -p_z)$, $(-p_x, p_y, -p_z)$, $(p_x, -p_y, p_z)$.



(A) seeds on unit sphere (B) asymmetric seeds (C) correct distribution

FIGURE 4.2: Seeds from unit sphere on ellipsoid(B). Principal axis for seed distribution corrected to preserve symmetry(C).

This is suitable to ensure symmetry between octants on the unit sphere, but it is not preserved on the ellipsoid after applying transformation T to all $\mathbf{p}$. For this, the coordinate axes of the ellipsoid and the unit sphere need to be aligned, such that the largest ellipsoid axis matches the y axis of the unit sphere. The additional transformation is applied to all generated and reflected points $\mathbf{p}$ before the transformation T is applied. As U holds the affine part of T, the alignment can be computed, e.g., if λ_2 is the largest eigenvalue, as follows

$$T' = TT_{\text{rot}}^{-1}, \quad \text{with} \quad T_{\text{rot}} = U \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{4.6}$$

The resulting seeds $\mathbf{s}$ on the ellipsoid $\mathcal{E}$ preserve symmetry between the octants.

4.1.6 Mapping Seed Vertices on an Ellipsoid

Given a suitable set of vertices $\mathbf{v}$ on a unit sphere $\mathcal{S}$, the transformation to the surface of the ellipsoid $\mathcal{E}$ is calculated from matrix $A \in \mathbb{R}^{3\text{x}3}$ and ellipsoid center $\mathbf{c}$ as provided by the Khachiyan algorithm [TY07]

$$\forall \mathbf{v} \in \mathcal{E} : (\mathbf{v} - \mathbf{c})' A (\mathbf{v} - \mathbf{c}) = 1 \tag{4.7}$$

We compute the LL^T Cholesky decomposition of the symmetric, positive definite matrix A such that $A = LL^* = U^*U$, where L is a lower triangular matrix and U is an upper triangular matrix. The affine transformation from $\mathbf{v} \in \mathcal{E}$ to $\mathbf{y} \in \mathcal{S}$ is given by

$$\mathbf{y} = U(\mathbf{v} - \mathbf{c}) \tag{4.8}$$

Thus, the desired transformation T from the unit sphere, centered in the origin, to the surface of the ellipsoid, with center $\mathbf{c}$, consists of a rotation according to U and a translation according to $\mathbf{c}$

$$T = \begin{bmatrix} 1 & 0 & 0 & c_x \\ 0 & 1 & 0 & c_y \\ 0 & 0 & 1 & c_z \\ 0 & 0 & 0 & 1 \end{bmatrix} U^{-1}. \tag{4.9}$$

Seed vertices $\mathbf{s} \in \mathcal{E}$ for RanCEAF are generated from randomized points on the unit sphere and mapped on the surface of an ellipsoid enclosing the object under investigation via $\mathbf{s} = T\mathbf{p}$. Elevation of seed vertices $\mathbf{s}$ is achieved by multiplication of A with a scaling factor $e \geq 1$, $\quad A_{\text{elevated}} = eA_{\text{MVEE}}$.

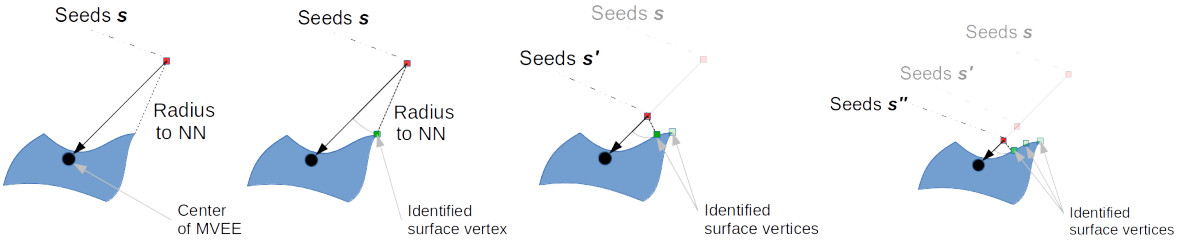
4.1.7 The Exterior Surface

Locally convex regions in the underlying mesh serve as attractors for NNS if they represent a protruding structure on the exterior surface. To expand their scope in answering NNS queries, each generated point q on the MVEE is shifted for simplicity by a factor $e = 2$, such that its distance to c_{MVEE} is doubled. This finally represents the seed location s as shown in Figure 4.1B. The last step is necessary to prevent local maxima of the mesh, contributing to the CH and defining the size of the MVEE, from only being

selected by an NNS query in case the randomized seed vertex s is identical to this extreme point of $\mathcal{M}$. Any factor $e > 1$ is sufficient, since the chosen value only effects the initial query, and is already compensated after the first seed-shift operation.

4.1.8 Expanding the Selection

Seed-shift operations allow for the extraction of larger surface parts, gradually relaxing the constraints on proximity to the MVEE and therefore the original seed vertex s. As shown in Figure 4.1C, subsequent NNS with shifted seed vertices s' allow for bypassing the most prominent and most protruding structures and expanding the selected exterior surface parts. In this case any seed vertex s is shifted towards the center c_{MVEE} by the distance $\|v - s\|_2$, which equals the distance to its nearest neighbor as it was returned from the initial query. It is still not possible to penetrate the exterior surface since the only vertex $p \in \mathcal{M}$ which can be reached from s by shifting it to position s' is v itself—and therefore a vertex on the exterior surface. The benefit of this operation is, that less prominent but still salient, locally convex regions on the exterior surface can be added to the extracted subset.



(A) start (B) first p (C) 1^{st} shift (D) 2^{nd} shift

FIGURE 4.3: Seed-Shift example for single s and multiple shifts.

4.1.9 The Resulting Subset

The attributes of the extracted data are that both reduced meshes

- only contain those parts visible from the outside, i.e., the exterior surface,
- exclude narrow cavities and covered regions behind obstacles,
- include samples distributed over the whole object, preferably from salient regions,
- only contain measurement results, and no smoothing, collapsing or averaging.

With this Random Convex-Edge Affine Feature (RanCEAF) selection, we present an approach to extract almost the same meaningful subset from each of the meshes $\mathcal{M}_{\text{Opt}}$ and $\mathcal{M}_{\text{CT}}$ as a pre-processing step to allow for efficient and robust alignment.

4.2 Alignment via OctaCoG

As the root cause of the observed alignment errors lies in point-to-point comparisons with inadequate comparison partners, our *RanCEAF* method is suitable solution to convert them to adequate partners. Still, point-to-point comparisons are not the preferred approach as data sets from optical scanning and Computed Tomography differ in resolution, coverage, topology and geometry. The second method we propose is *OctaCog* as alignment scheme not based on point-to-point comparisons, but on global features such as splitting the data set in *octants* and operating on their respective *Center of Gravity* (CoG).

4.2.1 Approach

A surface reduction to exterior features via RanCEAF is provided, preserving the better part of the mesh from optical scanning $\mathcal{M}_{\text{Opt}}$ and stripping all interior structures of the mesh $\mathcal{M}_{\text{CT}}$ from the CT isosurface extraction. Intended for the compensation of ICP drift errors, this reduces both meshes to point clouds $\mathcal{P}_{\text{OPT}}$ and $\mathcal{P}_{\text{CT}}$ representing similar surface parts. Instead of employing ICP to find a suitable alignment of both data sets, we can leverage the fact that both representations are from the very same object. Thus, they have similar geometric measures and corresponding analysis yields similar output in both cases.

The analysis we apply in order to directly find a suitable transformation are principal component analysis (PCA) and the computation of the barycenter, which in the case of a three dimensional body is called Center of Gravity (CoG). The name OctaCoG refers to segmenting the point cloud in *eight octants* according to PCA and computing an alignment based on their *CoG*s, as in Figure 4.4.

4.2.2 Center of Gravity

We compute the barycenter as presented in Section 2.4.3. As object, resolution and coverage are identical, we also assume comparable $\bar{v}$ results for both point clouds.

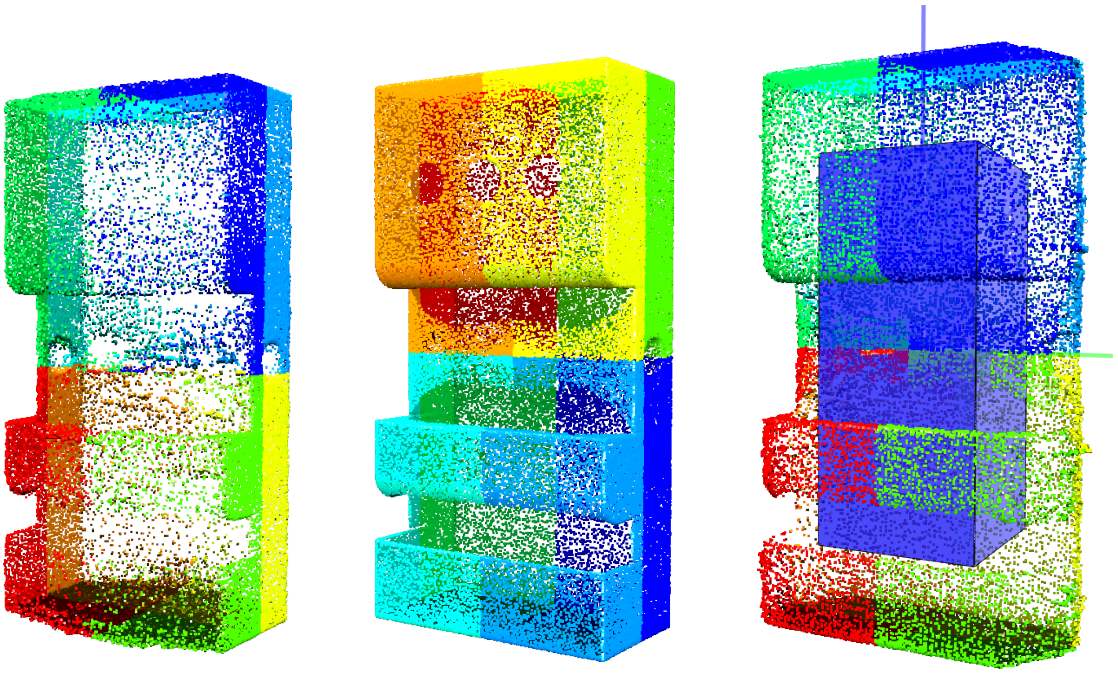


FIGURE 4.4: Surface subsets $\mathcal{P}_{\mathrm{CT}}$ of isosurfaces presented in Figure 1.9. Coloring indicates vertex assignment to octants. Right-hand figure features OctaCoG shape and PCA axes.

4.2.3 Principal Component Analysis

A PCA analysis as presented in Section 2.4.4 of each point cloud $\mathcal{P}_{\mathrm{OPT}}$ and $\mathcal{P}_{\mathrm{CT}}$ provides PCA axis $e_j, j \in (1, 2, 3)$. We will segment both point clouds by the three planes each two of the eigenvectors define.

4.2.4 Combining Octants and CoG

As for the calculation of the CoG, the computed PCA axes are at least similar for the two point clouds, but a direct alignment based on those is fragile and ambiguous. Slight deviations within the point cloud have a direct impact on CoG and PCA axes computation. Given the fact that the subsets are influenced by randomly generated seed vertices, artifacts and measurement uncertainty, a robust and reproducible alignment by PCA axes cannot be expected. To overcome the sensitivity against slightly differing point clouds, the computation of alignment parameters needs to be averaged.

Furthermore the disambiguation of a PCA based alignment can only be achieved by a trail and error approach. The three linear independent eigenvectors e_j are suitable to serve as a Cartesian coordinate system for the respective point cloud. But they are not a unique solution of the PCA analysis. In fact every scaling, except to zero, of each vector represents a similarly valid Cartesian coordinate system for the point cloud with

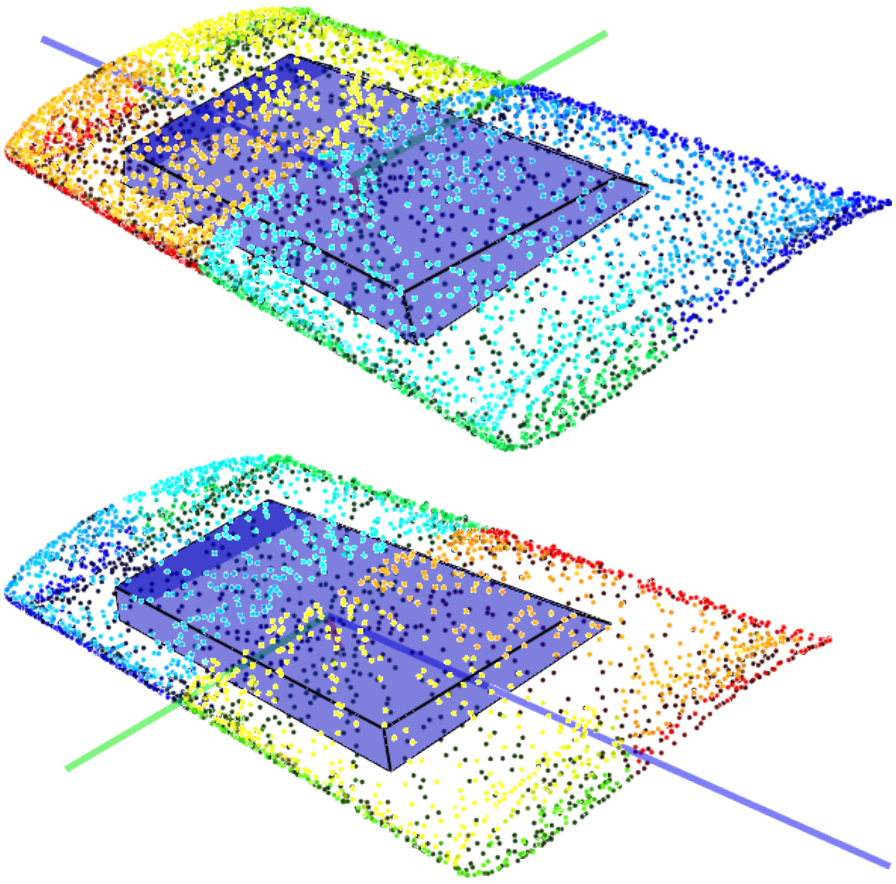


FIGURE 4.5: RanCEAF subsets $\mathcal{P}_{\text{OPT}}$ (top, 3168 vertices) and $\mathcal{P}_{\text{CT}}$ (bottom, 5006 vertices) of synthetic data with blue OctaCoG shape and PCA axes.

eventually inverted axes. For any given alignment of two sets of PCA axes, originating from the respective representations of both objects, three alternative alignment configurations exist. To explain this ambiguity, we assume two sets of PCA axes denoted as $A \perp B \perp C$ defined by vectors $\vec{a}, \vec{b}, \vec{c}$ and axes $K \perp L \perp M$, defined as $\vec{k}, \vec{l}, \vec{m}$. Both sets of axes are ordered by the corresponding λ values and the axes are pairwise parallel with $A \| K$ and $B \| L$ and $C \| M$. Besides the given alignment, inverting each two vectors produces similarly valid alignments based on the same PCA outputs. For $-\vec{a}, -\vec{b}, \vec{c}$ and $-\vec{a}, \vec{b}, -\vec{c}$ and $\vec{a}, -\vec{b}, -\vec{c}$ the resulting axes ABC are still pairwise parallel to KLM and fulfill the constraint that components match according to magnitude.

4.2.5 Robustness without Loss of Sensitivity

The averaging in our approach, intended to mitigate the sensitivity of PCA to slight deviations in $\mathcal{P}$, also solves the disambiguation. We segment each point cloud in octants along the PCA axes with the CoG as intersection point of PCA axes and origin of the octants. Now the barycenter of each octant is computed from all points within these boundaries. As shown in Figure 4.6 for the eight corner points of a cube, the resulting

segmentation assigns each corner point to one octant. It is easy to see, that small deviations in the corner points will alter the computed PCA axes, but their assignment to octants is not effected by this. Given a larger number of vertices in the point cloud under investigation, this implements the intended averaging. Although this reduces the influence of noise, artifacts or randomness in RanCEAF sampling to the octant assignment, each barycenter is still sensitive for small deviations in the input data. This ensures, that the computed alignment respects local features instead of smoothing the intrinsic differences in the measurement results.

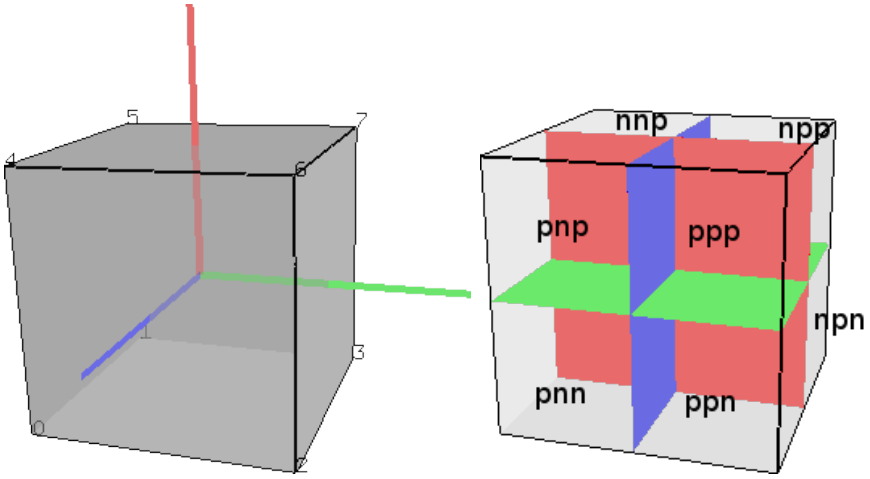


FIGURE 4.6: PCA axes for eight corners of a cube (left) and resulting octants with positive (p) or negative (n) sign.

A common nomenclature to address the eight segments of on octant is top-front-right as *ppp*, top-back-right as *npp*, top-back-left as *nnp*, and so on. According to this nomenclature the PCA axes in Figure 4.6 (left) correspond to x-axis as blue component, y-axis as green component and z-axis as red component.

4.2.6 Orientation based on OctaCog Shape

Disambiguation is achieved by computing an aligning based on the octant barycenters, instead of the PCA axes. They can be represented as quadrilaterally-faced hexahedron by connecting all barycenters from octants that share a common edge, which corresponds to octants adjacent to each of the six half-axes of the PCA. It is important to notice that in the general case, this shape is neither cubic, nor symmetric. Therefore, a unique alignment exists no matter if the two input meshes have identical content, as being from the same measurement device, or slightly different content, as being acquired via different imaging techniques. Cubic shapes can only occur for highly symmetric input data, but due to numerical limits are unlikely and usually produced by synthetic inputs. Figure 4.5 shows the resulting shape for $\mathcal{P}_{\mathrm{OPT}}$ of the synthetic data set from Figures 2.5

and 2.6. The intersection point of the PCA axes denotes the CoG and the OctaCoG shape is super imposed in blue. The coloring refers to the octant a vertex is assigned to.

4.2.7 Implementation

Given the surface subsets described in Section 4.1, all aspects of the presented approach are implemented. For efficiency, we reorder the steps such, that the barycenter computed for PCA in Algorithm 2 is reused in octant assignment in Algorithm 1.

Furthermore, in contrast to the described approach, octant assignment is not done after transforming the mesh to align PCA axes to coordinate axes. This would mean to multiply each point with a 4x4 transformation matrix, compare each coordinate per point to zero, assign the point to an octant based on the signs (*ppp*, *ppn*, ..., *nnn*) and transform the result back according to the inverse of the original transformation. Instead, we compute the distance of each point to three planes as defined by PCA vectors, and assign octants based on positive or negative distance. Assignment based on the Hesse normal form representation of segmentation planes is calculated from three dot products per point and much faster to compute. The outcome of both methods is identical.

4.2.8 Resulting Transformation

The alignment of two such shapes is computed by concatenating the shape edges in x/y/z-direction separately. This includes the accumulated data of all points, as the calculation is based on sector barycenters. We explicitly avoided to reflect small deviations in the point cloud during segmentation, but of course those deviations need to be considered at some point, since we do not want to present two identical alignment matrices for two similar, but not identical, sets of point clouds. The four edges connecting adjacent sectors in x-direction—sectors on the left to their right hand neighbors, separated by the blue plane in Figure 4.6—are $ppn - -pnn$, $npn - -nnn$, $npp - -nnp$, $ppp - -pnp$. These edges connect the octant barycenters, their sum is denoted as $\vec{o_x}$ and will serve us as one component for alignment. Respectively, in y-direction and z-direction a similar summation over vectors crossing the green and red plane are computed as $\vec{o_y}$ and $\vec{o_z}$.The transformation matrix is computed as concatenation of a translation and rotation. The translation moves the CoG of one point cloud to the CoG of the other point cloud. For rotation, we select the side of the OctaCoG shape with the largest surface and set the corresponding $\vec{o}$ vector in direction of the opposing side as first component for alignment. For the smallest of the remaining sides, the corresponding $\vec{o}$ vector from direction of the opposing side is set as second component for alignment. The relation of surface area

between the different planes of the OctaCoG shape has proven as reliable indicator for disambiguation of object orientation [BLK16].

For alignment, we compute the transformation T matrix of each subset $\mathcal{P}_a$ and $\mathcal{P}_b$ such, that we translate the subsets' CoG to the origin and align the selected axes to the coordinate axes. The final alignment matrix is a concatenation of $T_a \times T_b.inverse()$ and can conveniently be formulated via quaternions. The evaluation of the OctaCog alignment is presented in Section 5 and a detailed comparison table provided in Appendix D.

4.2.9 Key Contribution

OctaCoG takes prior knowledge about the representation into account and circumvents to iteratively converge to a solution. As described in Section 3.5 point-to-point comparisons are not suitable to align $\mathcal{M}_{\text{Opt}}$ and $\mathcal{M}_{\text{CT}}$. Instead, the presented method operates fully resolution independent and only considers global features. It allows for highly accurate and fast alignment based on point clouds of the complete exterior surface. The solution is found in $\mathcal{O}(n)$, without manual tie point identification or any other user interaction.

4.3 CMM to Mesh

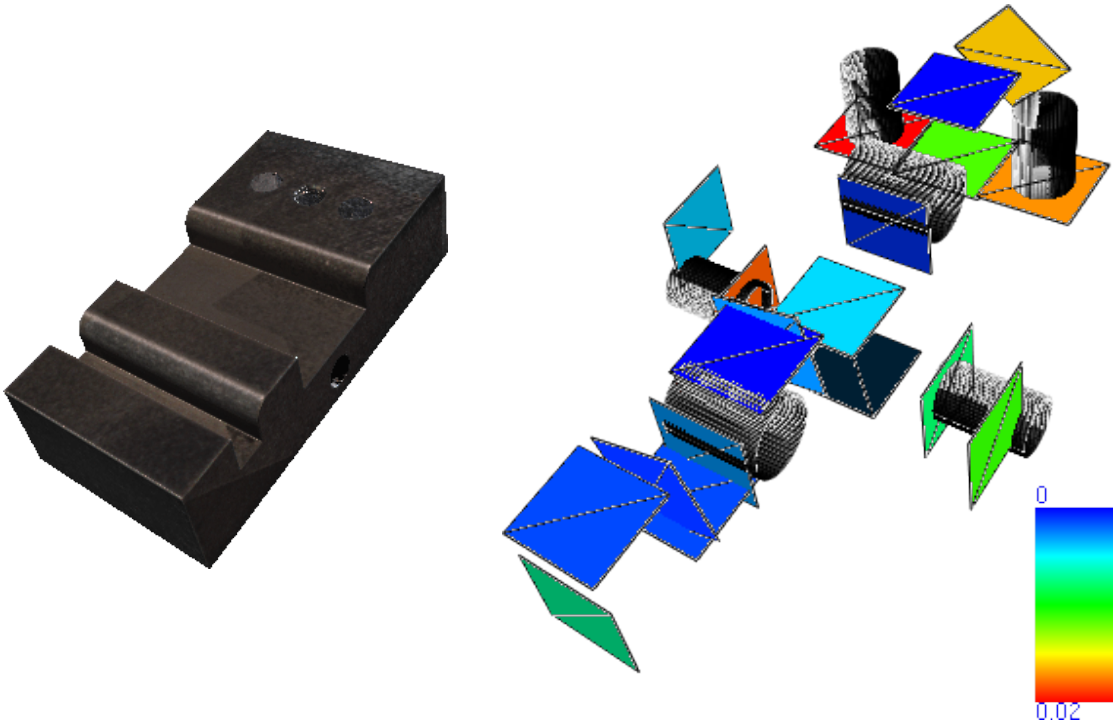
During the ILATO project several tactile measurements were performed to validate the optical scans and CT acquisitions, yet a compatible representation to process those results does not exist. With *CMM to Mesh* it is possible to generate mesh $\mathcal{M}_{\text{CMM}}$ from corresponding measurements.

It is impossible to provide ground truth for our approach since all measurement techniques and manufacturing processes have certain inaccuracies—and even those values are questionable. The milling process to manufacture the samples lists an uncertainty of 15 μm, yet some samples have a deviation of several mm from the CAD specification. The optical scanner is calibrated to provide accuracy of 12 μm for high resolution optics, yet we encounter deviations of 0.5 mm to reference measurements due to misalignments in the merging process. CT, depending on the combination of source and sensor, also claims to provide 10-15μm, which in practice is not reached due to miscalibrated voxel sizes, improper alignment of rotation stage or the inability to segment support material and specimen. In general, all those uncertainty values might very well hold for a single drilling hole in the case of milling machines, a single acquisition in the case of optical scans or a single projection in case of CT. For sure, they do not apply for manufacturing

a whole object, the full surface mesh from optical scan or a volumetric reconstruction from CT. As tactile measurements via Caliper or CMM only consist of independent single acquisitions, their uncertainty of $\sim 1\mu m$ each holds and is consistent if compared to each other. However, tactile measurements cannot describe free form surfaces, are difficult to match against meshes, and more difficult to convert to mesh representations.

4.3.1 CMM Measurement Output

An example of converting CMM results—surface descriptions parametrized as cylinders and planes—in a surface mesh is given in the following. To enable a comparison of meshes from optical scan, CAD or volumetric data with the tactile measurement represented as CMM, primitives are generated to suit the KNN query and allow for calculating the Hausdorff distance or RMSE. Coordinates and normals are read from CMM output and represented as planar objects and cylindrical objects. A comparison of CMM representation and surface mesh is shown in Figure 4.7. The presented visualization consists of a square with fixed edge length depicting each planar measurement result from CMM and the lateral area from cylinder measurements also with fixed height. The coloring of squares corresponds to the measured deviation given as *flatness* and *roundness* in the CMM results, the color scale is included in Figure 4.10.



(A) Sample 1A (B) CMM representation

FIGURE 4.7: Comparison of $\mathcal{P}_{\mathrm{OPT}}$ and corresponding CMM primitives (central drilling hole with screw thread not measured).

4.3.2 Intersection Point of Three Planes

We want to adjust the surface parts from CMM in edge length and eventually shape in order to form a watertight mesh. Thus, we need to intersect all planes and find a sane configuration in which adjacent surfaces have plausible orientation and all CMM results are included. As initial step, we generate a wireframe model depicting all possible intersection points of each triplet of planes as shown in Figure 4.10. The three planes intersect in one point if for their normal vectors holds $n_1(n_2 \times n_3) = 0$. If this is not the case at least two are parallel, coplanar or form a prismatic surface and in any case can be neglected for our purpose. If the condition is matched, the intersection point x from position vectors p_i and associated normal vectors n_i of planes $A_i, i \in (1..3)$ is calculated as

$$x = (|n_1 n_2 n_3|)^{-1} \left[(p_1 n_1)(n_2 \times n_3) + (p_2 n_2)(n_1 \times n_3) + (p_3 n_3)(n_1 \times n_2) \right]. \tag{4.10}$$

For each point x originating from the intersection of three planes, graph knots (red edges in Figure 4.10) indicate all other points sharing two of the three planes and therefore potential adjacent edges of x for forming the Jordan curve.

4.3.3 Finding Boundaries for Planes

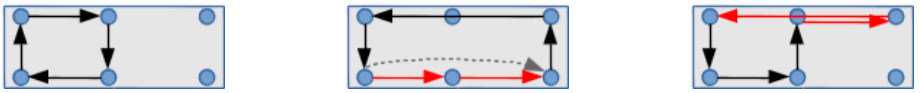
A Jordan curve divides a plane in *interior* and *exterior* region. The requirements for such a curve are that it (a) forms a closed cycle and (b) does not contain any self intersections (Figure 4.8).

(A) Simple cycle

(B) Cycle with self intersection

FIGURE 4.8: Graphical example of malformed cycle with faulty edges in red.

Each plane in the CMM data set has one correct Jordan curve representing its boundaries. Yet, several edge compositions from intersections with other planes can provide this configuration. In order to limit the amount of redundant solutions, further restrictions have to be formulated. As shown in Figure 4.9 we want to exclude the reverse cycle of each accepted edge sequence. Furthermore, containing edge pairs that have parallel consecutive edges need to be neglected since a shorter cycle describing the same Jordan curve must exist. In practice, it is also appropriate to demand the inclusion of the CMM measurement point which defines the current plane to be circumnavigated by the edge sequence, although theoretical examples exist to contradict this requirement.



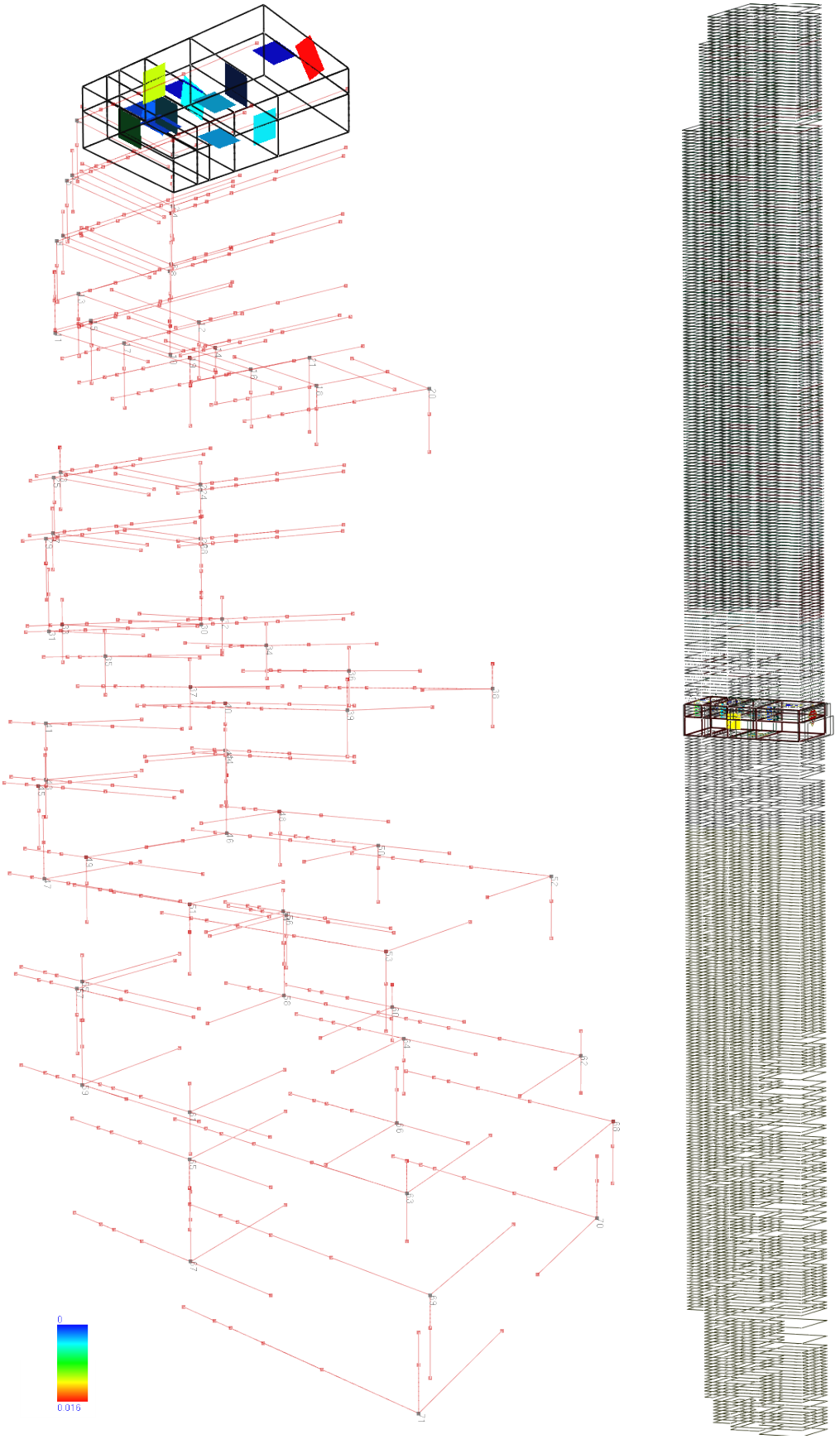
(A) reverse (B) parallel in same direction (C) parallel in opposite direction

FIGURE 4.9: Additional requirements for edge cycles forming the boundary of a plane.

We follow a greedy approach to generate all possible cycles within one plane. For each vertex all valid edge sequences are formed by exploiting the neighborhood relations stored from plane intersections and visualized in Figure 4.10A. After all paths starting at the current vertex are found only those with matching end and start vertices, i.e., the cycles, are kept. Also all cycles already known from previous vertices are deleted. Now all conditions (Figures 4.8 and 4.9) are checked and the remaining cycles considered as possible Jordan curve of the plane corresponding to the CMM measurement.

4.3.4 Selecting Final Set of Loops

A valid solution to transfer the CMM results in a mesh $\mathcal{M}_{\text{CMM}}$ must include exactly one cycle per CMM measurement as Jordan curve and provide a closed surface. We implement this by maintaining a *mandatory* set of vertices, and an *optional* set of vertices. A recursive function to identify a valid solution starts at one plane and processes each cycle. For each processed cycle, the respective vertices are set as *mandatory* and removed from set *optional*. Furthermore, the current plane is added to a set *processed planes*. As each vertex in each plane originated from the intersection of three planes and holds information to identify those, we refer to those planes as neighbors. Therefore the function is executed for each neighboring plane not included in the set *processed planes* and in turn processes all cycles which can be formed by including at least one mandatory vertex and apart from that only optional vertices. As the function is successively executed for the next neighbors, more planes are added to the processed set and less vertices remain in the optional set. If a function is called with no neighbors left to process, the function either returns all remaining valid cycles in the current plane or indicates a failure. In case no cycles can be formed for the current plane, the function also returns a failure status. A function receiving a failure state from any recursively called function also returns failure. After all configurations have been evaluated the valid solution is returned, identifying one specific Jordan curve per plane which in total represent the watertight mesh $\mathcal{M}_{\text{CMM}}$.



(A) Intersection edges for all CMM planes (black) and stacked graphs depicting neighborhood relations of each vertex (red).

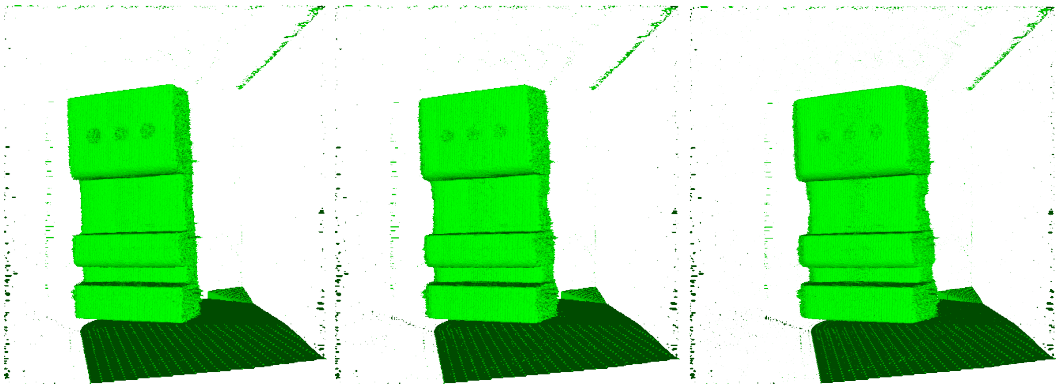
(B) ∼ 1000 resulting Jordan curves stacked by plane

FIGURE 4.10: Results of intersection tests (left) and cycle creation (right).

4.4 Gradient Assessment

The differences in data representation between $\mathcal{M}_{\text{Opt}}$ and $\mathcal{M}_{\text{CT}}$ are at least partially caused by limited-angle CT artifacts as described in Section 2.2.2. Above methods would both benefit from a proper localization of mentioned artifacts. Also the isosurface extraction itself could provide more meaningful results. With *Gradient Assessment* [Kle16] we present a method intended to tag the voxels in volumetric representation as being *reliable* or *LACT compromised*. And to a certain degree this even allows to correct of the volumetric data set itself.

The observation motivating the works on *Gradient Assessment* and shown in Figure 4.11 is that considerably small changes in the chosen threshold for isosurface extraction have a tremendous effect on certain areas of the resulting mesh, e.g., the sides and drilling holes in the current example, while other areas like front and back are preserved quite well. Compared to the trajectory indicated in Figure 1.9 and as explained in Section 2.2.2 only those surface part of the specimen, parallel to the central ray of any projection do not suffer from LACT artifacts. Clearly this is the case for the front in Figure 4.11 as those areas of $\mathcal{M}_{\text{CT}}$ remain unchanges while the specimen seems to grow in width for decreasing thresholds. The effect can be explained by inspecting the scalar field from



(A) threshold 0.013 (B) threshold 0.010 (C) threshold 0.008

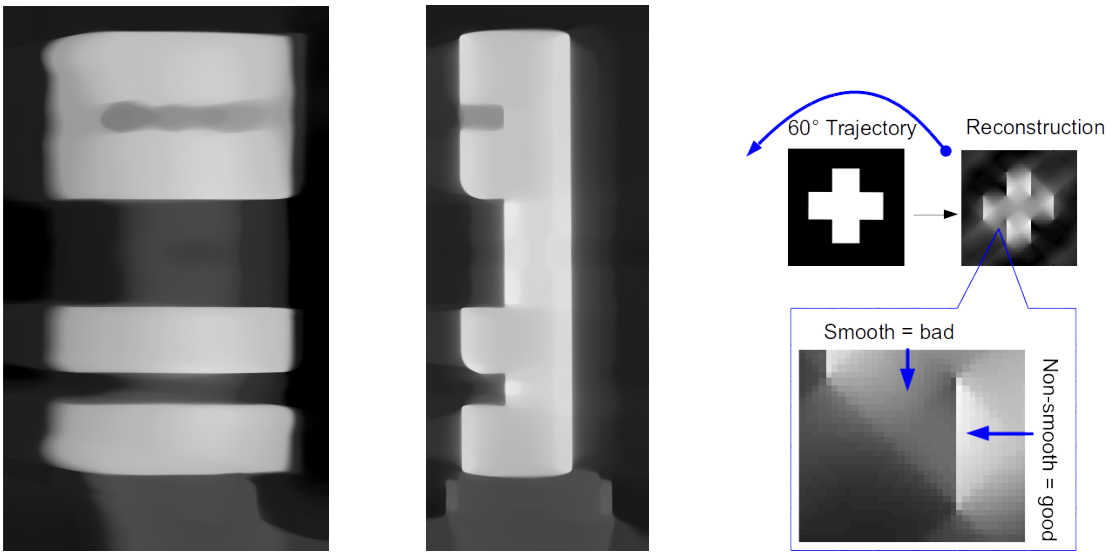
FIGURE 4.11: Influence of threshold variations on LACT artifacts in $\mathcal{M}_{\text{CT}}$.

which the isosurface is extracted.

4.4.1 Visualization of Limited-Angle Artifacts

A slice through the volumetric data set is presented in Figure 4.12 where (A) shows a frontal view at the specimen. The grey values are fading out at the left and right side with a smooth gradient. SubFig. (B) shows the lateral view depicting a comparably sharp transition from dense material (white) to the surrounding air (black). The effect

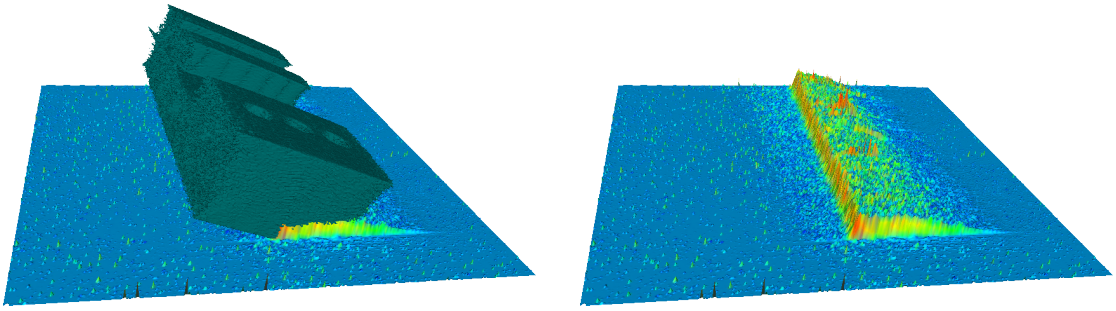
of LACT artifacts as faded and smoothed regions in the reconstructed scalar field is also shown in SubFigure 4.12C on the example of a 2D cross phantom. The blue arc above the phantom depicts the trajectory. Next to it, the resulting density distribution with well preserved boundaries for surface parts perpendicular to the trajectory.



(A) Slice front view (B) Slice side view (C) Gradient example

FIGURE 4.12: LACT artifacts in slices of volumetric representation.

Thus, the robustness of the extracted surface against small changes in the chosen threshold is not equal for all parts. In regions with a steep gradient in the volumetric representation, robust surface parts are extracted. Smooth gradients result in volatile surface parts, their location varies a lot even for small changes in the threshold. We want to preserve the information which parts are reliable and which parts are volatile since reliable surface parts are excellent for alignment and also should be excluded from any correction attempts.



(A) Slice superimposed on $\mathcal{M}_{\mathrm{CT}}$ (B) Slice with density as z component

FIGURE 4.13: Desity values in one slice of the scalar field superimposed on $\mathcal{M}_{\mathrm{CT}}$.

A similar observation can be made on the extracted surface when visualizing a slice of the scalar field along the coordinate axes. In Figure 4.13 one layer of voxels with fixed z

coordinate is shown, the z component depicts the density value of the respective voxel. It is easy to see that the top plane of the specimen located at a very steep gradient and would not change position for slight variations of the threshold. The distorted side plane of the specimen on the other hand is in a ramp shaped region of smoothly decreasing density values which we identify as limited-angle artifacts. For a CT scan with full circular trajectory, we would expect an area of elevated density with steep ridges at all sides instead of a ramp.

4.4.2 Local Gradient

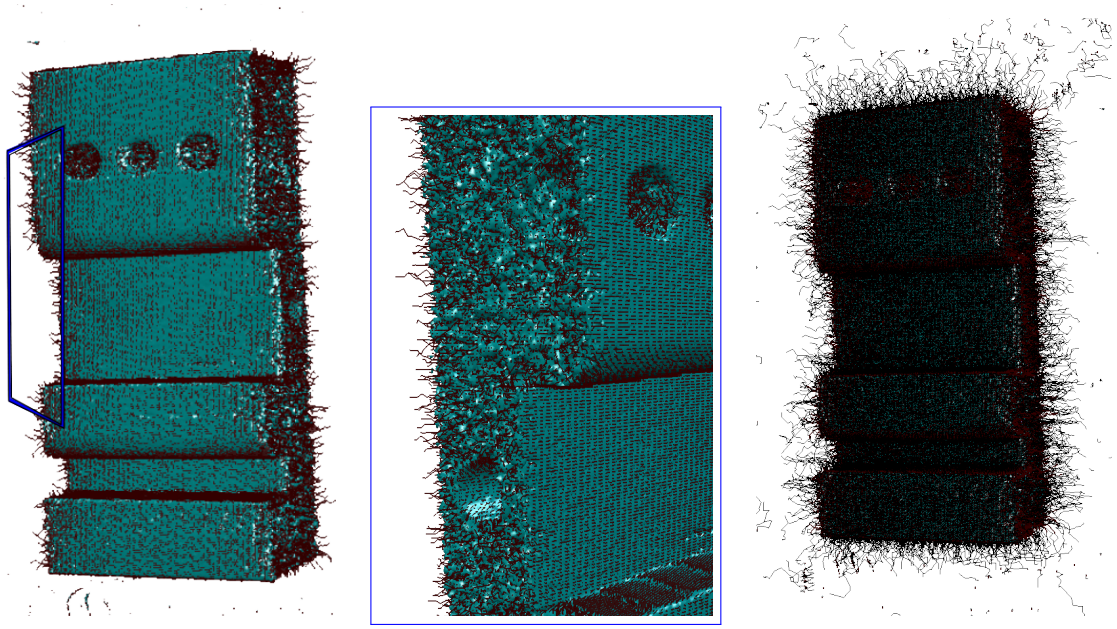
The analysis of surface reliability has to consider the gradients in the volumetric representation. More specificity, we analyze the local gradients between neighboring voxels along the coordinate axes. So for a given voxel v with coordinates (v_x, v_y, v_z) in the reconstruction volume, we compute a gradient for the six neighboring cells (v_{x+1}, v_y, v_z), (v_{x-1}, v_y, v_z), (v_x, v_{y+1}, v_z), (v_x, v_{y-1}, v_z), (v_x, v_y, v_{z+1}), (v_x, v_y, v_{z-1}).

4.4.3 Steepest Path

As the blurring effects in the volumetric representation cannot be observed be comparing just two neighboring voxels, we establish a relation for voxels in greater proximity. The concatenation of edges between neighboring voxel centers forms a path. Along a path all edges are associated with the respective local gradient between the voxels they connect. Although, their length corresponds to the voxel spacing, i.e., all edges in x direction have identical length, as have all edges in y direction and all edges in z direction. To identify areas with limited-angle artifacts we compute paths of steepest descent—according to local gradient direction—in each voxel of the volumetric representation. Also the path of steepest ascent are computed, providing a map for all voxels towards the global minimum and global maximum of all density values in the scalar field. In synthetic data only two values, i.e., the density of material and the density of void space, would occur. Real data is noisy, e.g., as indicated by the tiny spikes in Figure 4.13B distant from the object, thus we truncate the density range by applying percentiles.

4.4.4 Steepest Path with Interchange

For the steepest paths as shown in Figure 4.14A the descent most likely ends in a local minimum instead of the global one. One the one hand the path length to a local minimum approximately corresponds with the presence of LACT artifacts, on the other hand we want to evaluate complete paths and therefore need to enable further descent.



(A) Steepest paths until local minimum

(B) Zoom to defectice side plane in (A)

(C) Steepest paths continued after local minimum

FIGURE 4.14: Steepest path between voxel centers following local gradient. Only paths attached to surface truncated to 20 edges.

If trapped in a local minimum, the obvious solution is to continue the path at the closest voxel with lower density even if this voxel is several steps away. We refer to this as a path *interchanging* in order to bypass local minima and continue its descent.

4.4.5 Corrections along each Path

Noisy data and discretization might be responsible for small deviations in the density values which cause paths to be trapped in local minima. Greater fluctuations as especially observed in defective areas due to LACT atrifacts cannot be explained by noise or scattering. In fact the alternations of maxima and minima along each path lack any justification and therefore could be corrected such that the resulting scalar field does not contain local minima in direct neighborhood of local maxima. The following Figure depicts a path and this correction attempt. As the specimen described in Section 2.5.1 all are single components of homogenous material those fluctuations —especially around the threshold for isosurface extraction—correspond to multiple penetrations of the objects surface. A path of steepest descent should exist in the given scalar field and the interchanging mechanisms should only account for small defects. Also, each path should cross the given threshold only once while ascending from the start voxel to the maximal density and once during descent to the minimal density. Therefore, corrections on the scalar field as shown in Figure 4.16 have a positive effect on the surface representation.

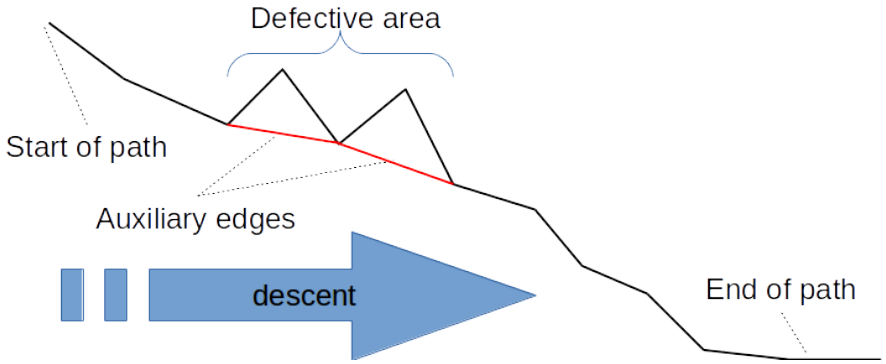


FIGURE 4.15: Descending path with defective areas and intended correction attempt.

Alternating minima and maxima resulted in a very fragmented isosurface in Subfig. (A) while the adjusted scalar field permits the extraction of a far more realistic isosurface without fragmented object center.

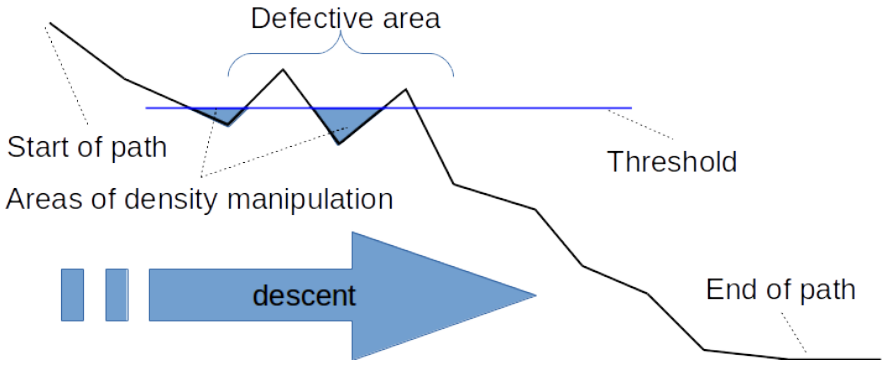
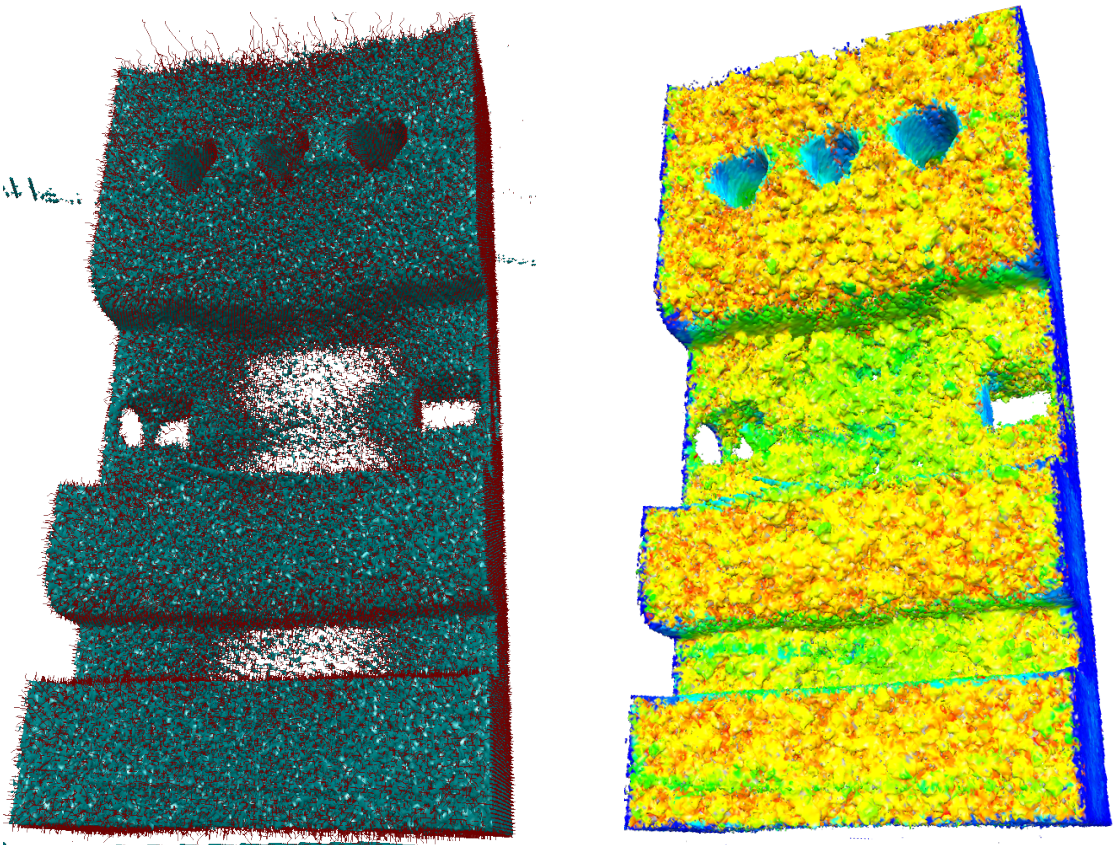


FIGURE 4.16: Scalar field manipulation based on path monotony.

As a result, defective areas as shown in Figure 4.17 can be restored. The truncated trajectory, only comprising a 60° segment does not allow to restore the correct surface but the density distribution within the reconstruction volume now fits to the assumption that a single component with homogenous materials is represented. Furthermore the resulting paths connect each voxel with the global maximum and minimum of density values, which allows for the coloring shown in Figure 4.17B depicting the total path length to the respective extreme value.

4.5 Mask Generation

Two approaches for providing prior knowledge to reconstruction have been investigated. On the one hand each ray—as measured in CT—can be segmented by comparing trajectory information with aligned optical measurements. We refer to this as “Ray clipping”. On the other hand, the reconstruction volume itself can be segmented in material and air, to which we refer as “Volume clipping”.



(A) ILATO sample before manipulation

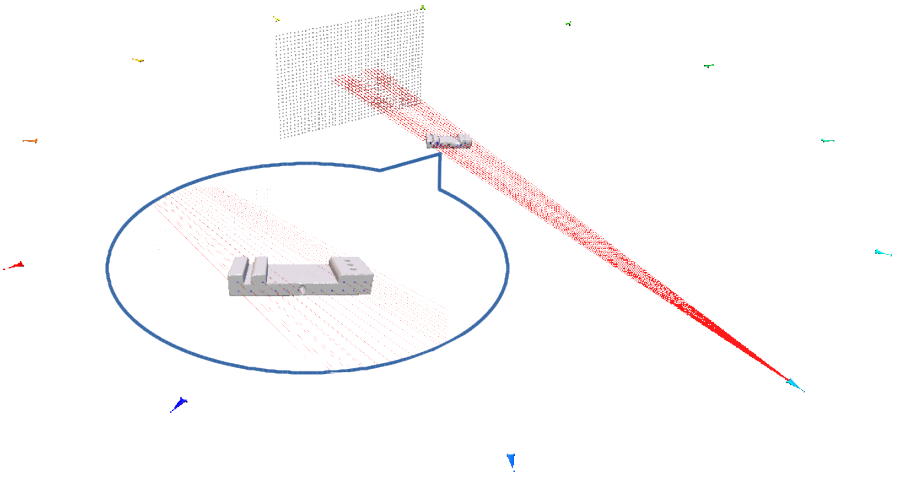
(B) Scalar field corrected for monotonic paths

FIGURE 4.17: Manipulation of voxel densities to provide monotonic paths. Surface color for path length from blue (short) to red (long).

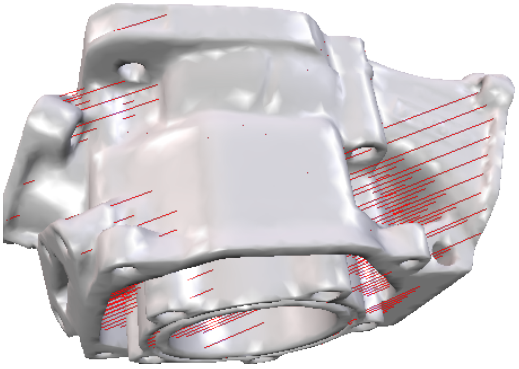
4.5.1 Ray Clipping

To improve the reconstruction process we provide run length of the ray in material. This information is obtained from applying ray casting techniques to the mesh from optical scan as presented in Section 3.6. A prerequisite is to align the optical mesh to the VESTA surface of the reconstructed scalar field.

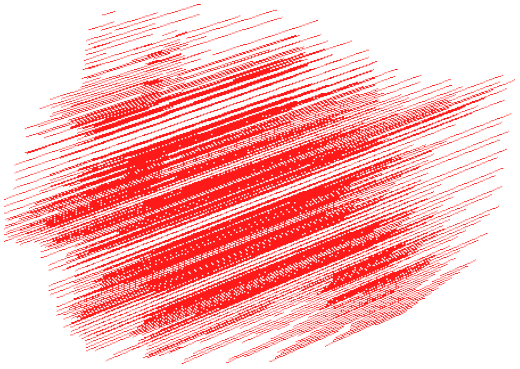
Raycasting tests a given ray defined by an origin and an orientation against the mesh. Our implementation returns the coordinates of any intersection along the ray, as well as the direction of the intersection. Therefore, it is possible to distinguish entry and exit events as intersection from front or back of triangle. The intersection check consists of multiple steps starting with a test of every ray for intersection with the bounding box around $\mathcal{M}$ via *SLAB* method explained in Section 3.6.1. The majority is usually discarded during the first test, for the remaining rays each triangle in $\mathcal{M}$ has to be tested for a potential hit via *Plücker Coordinates* (see Section 3.6.3). For identified all triangles the exact intersection coordinate is calculated via *Barycentric Coordinates* as presented



(A) trajectory and with sensor pixels and x-rax sources



(B) object with rays



(C) rays in material

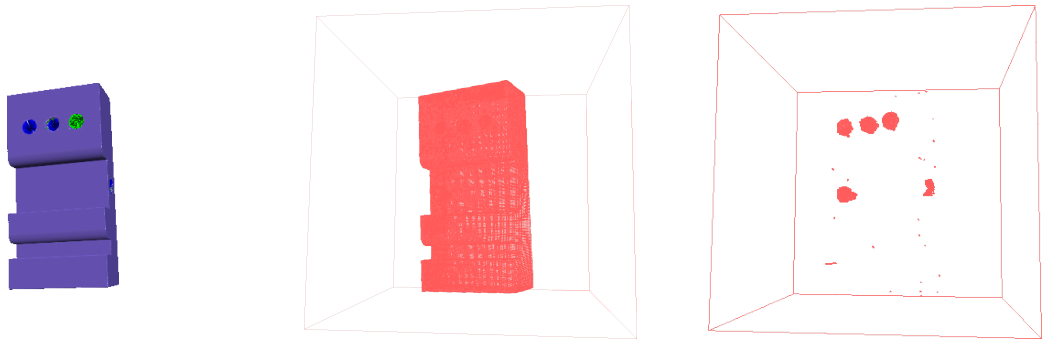
in Section 3.6.2. Optimized space partitioning structures can support those tests and drastically reduce the run time. Also specialized hardware, e.g., GPUs, CPUs with SSE4 instruction set or AVX2 extensions, makes some of above pre-processing steps obsolete.

The Mask Format in this case lists each ray to each pixel in each projection with its number of intersections followed by the respective distances. The projections occur in the same order as in the trajectory file and pixels are iterated from top left by line to bottom right. Negative distances indicate exit events while positive distances mark penetrations from air to material. In case the intersected triangle does not origin from the acquisition, but from a retrospectively added patch, the distance value is increased by the source-sensor distances as indication.

4.5.2 Volume Clipping

The prior knowledge represented as volumetric mask of course assumes successful alignment. As explained in Section 3.6.4 this algorithm processes all triangles in the mesh $\mathcal{M}$ and all voxels within the reconstruction volume $\mathcal{V}$. For each triangle an axis aligned bounding box is computed and expanded to match the voxel spacing of $\mathcal{V}$. The provided

information indicates if a given voxel in the reconstruction volume overlaps with surface parts of $\mathcal{M}$. In case the expanded box extents to several voxels, intersection test as in Section 3.6.5 *triangle versus box* determine if the respective voxel is tagged within the mask.



(A) input mesh (B) cells intersected by mesh triangles (C) cells intersected by patches

The Mask Format in this case includes the spacing and step size of the intersected volume.

$$\underbrace{340 \quad 670 \quad 340}_{\text{(int) DIM}} \quad \underbrace{0.15 \quad 0.15 \quad 0.15}_{\text{(float) SPACING}} \quad \underbrace{0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ldots}_{\text{(char) status}}$$

In contrast to ray based masks, no distance measure is provided. Instead, we annotate each voxel in the volume in the same order it is stored in file and assign a status to distinguish between *no intersection*, *intersection with mesh* or *intersection with patch*.

Chapter 5

Evaluation and Results

This Chapter presents evaluations of the methods developed for alignment and compares them to established methods. The evaluation focuses on effectiveness and efficiency of both approaches. The goal is to extract exterior surface parts as described in Section 1.5 to collect the maximum of corresponding measurement points from presumably LACT artifact free regions and to compute an alignment based on them.

A brief summary of results originating from Gradient Assessment and CMM Conversion is given in the following. Although, the former is still ongoing work and the latter lacks comparison partners for a thorough evaluation. The discussion of results as well as starting points for future work is given in the next chapter.

5.1 Surface Extraction Comparison

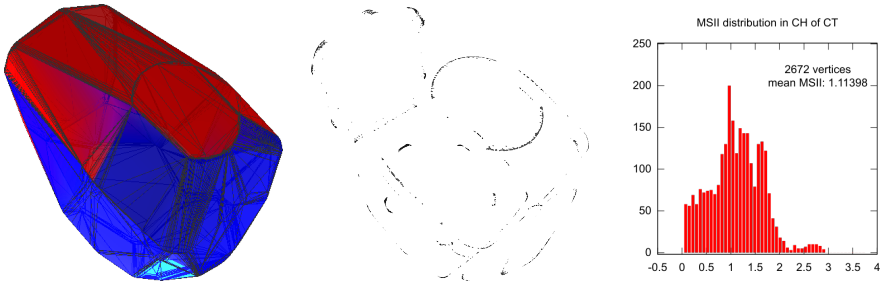
Exterior surface identification is expected to include salient regions visible from the outside, i.e., from the perspective of an optical scanner. Internal structures and parts of the mesh covered by obstacles shall not be included. In our context the salience of a region is closely related with curvature. On the one hand, curved structures are always more attractive for alignment than flat regions since they offer distinguishable features. On the other hand, LACT artifacts are most prominent at surface parts not parallel to any central beam in any projection. Thus, it is much more likely for curved regions than for flat regions that any central ray corresponds to the tangent of the surface. Therefore, curved regions, e.g., the rounded corners of inlets and outlets of the cylinder cast object, are usually reliable surface parts and should be preferred over flat regions for alignment. We measure curvature as salience via Multi-Scale Integral Invariants (Section 3.4), determine the fraction of the total surface included in the extracted surface and the salience of all vertices within this subset.

5.1.1 Enclosing Primitives

Approaches like AABB and MVBB identify six vertices each which is not sufficient for providing an alignment. Likewise, the MVEE is calculated from the set of vertices in the CH, but defined by eight points. Thus, they only allow for an estimation of object dimensions, but no identification of the exterior surface is performed.

5.1.2 Convex Hull

Convex Hull identifies the convex set of any $\mathcal{M}$ including the extreme points as described in Section 3.3, and allows for generating $\mathcal{M}_{\text{CH}}$ as shown in Figure 3.15B. For $\mathcal{M}_{\text{CT}}$ 1785 of 1.30 million vertices contribute to the CH, for $\mathcal{M}_{\text{Opt}}$ these are 2903 of 4.81 million vertices. Since the CH is not influenced by any parameter except the vertices of the mesh itself and each mesh contains exactly one convex set, no alternative subset can be identified. With the highest mean MSII value of all subsets and the absence of vertices with a MSII value close to zero, the result as shown in Figure 5.1C is a sufficient feature extraction. The CH never contains internal structures but contains only the most prominent protruding structures. It is therefore not suitable to provide the basis of an accurate alignment in general. Intuitively, it seems sufficient in the presented case, but the applied ICP algorithm cannot compute a valid transformation.



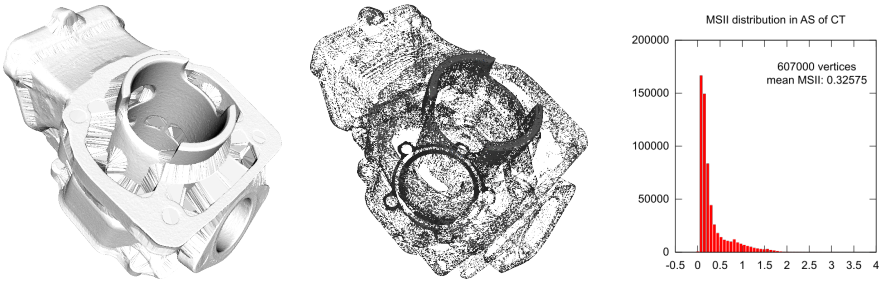
(A) $\mathcal{M}_{\text{CH}}$ of $\mathcal{M}_{\text{CT}}$ (B) Identified subset (C) MSII distribution

FIGURE 5.1: Convex Hull algorithm applied to $\mathcal{M}_{\text{CT}}$.

5.1.3 Alpha Shapes

Alpha shapes generate a surface $\mathcal{M}_{\text{AS}}$ for $\mathcal{M}_{\text{CT}}$. As shown in Figure 5.2, AS does not identify the exterior surface, since interior structures are covered by $\mathcal{M}_{\text{AS}}$ and therefore included in the resulting subset. The same holds for $\mathcal{M}_{\text{AS}}$ of $\mathcal{M}_{\text{Opt}}$ shown in Figure 3.15C. The mean MSII values in the subsets are in the region of the corresponding original meshes (see Table 5.1) and the histogram in Figure 5.2C is dominated by MSII

values close to zero which makes AS unsuitable for feature extraction. Experiments with lower α-values did not improve the result.



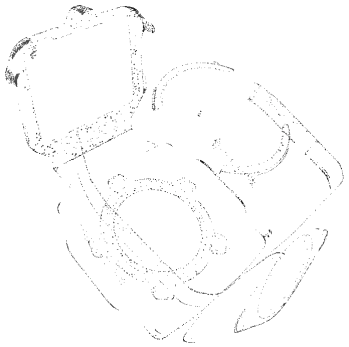
(A) $\mathcal{M}_{\text{AS}}$ of $\mathcal{M}_{\text{CT}}$ (B) Identified subset (C) MSII distribution

FIGURE 5.2: Alpha Shape algorithm applied to $\mathcal{M}_{\text{CT}}$.

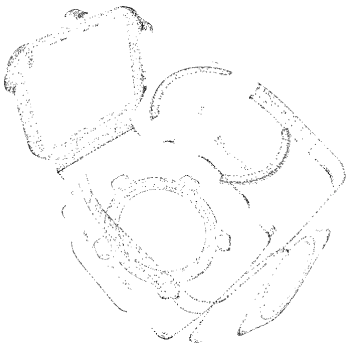
5.1.4 Randon Convex-Edge Affine Features

RanCEAF subset of $\mathcal{M}_{\text{Opt}}$ (Figure 5.3A) for 50k seeds contains 7474 vertices of all 4.81 million vertices. The RanCEAF subset of $\mathcal{M}_{\text{CT}}$ (Figure 5.3B) contains 5023 vertices of all 1.30 million vertices. As the extracted surface parts in both cases represent less than 0.05% of the vertices $p \in \mathcal{M}$, only the most prominent structures have been selected. The number of (removed) duplicates within the selection indicates that a small fraction of the exterior surface dominates the result by answering multiple NNS queries each. Thus, for sample sizes larger that 50k seeds no drastic change in the extracted subset is expected since we already over-sampled this subset by one order of magnitude. To allow for scalable mesh reduction, shift-seed operations (Figure 4.1) provide sufficient data for alignment. The mean MSII values provided in both subsets are second highest after CH, which makes RanCEAF a suitable method for feature extraction. None of the presented RanCEAF results include interior structures and only after the third seed-shift operation MSII values close to zero dominate the histogram (Figures 5.4F and 5.5F). For illustration purposes, Figures 5.3,5.4 and 5.5 show the extracted set of vertices and their connected faces. Via region growth in each vertex of the subset, more faces can be included to extract a larger portion of the exterior surface.

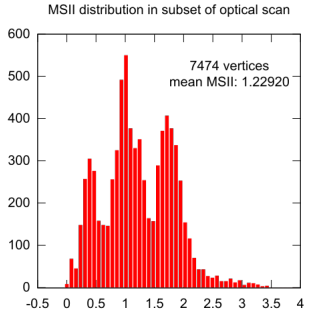
Seed-shift operations, as applied to $\mathcal{M}_{\text{Opt}}$ in Figure 5.4, and to $\mathcal{M}_{\text{CT}}$ in Figure 5.5 expand the regions from which exterior surface points are selected and still provide a higher mean MSII value than the original meshes in Figure 3.18. As the percentage of vertices with an MSII value $\geq$1.0 in Table 5.1 indicates, expanding the subset does not over-represent regions with low MSII values. In our experiments, the best increase in alignment accuracy was based on the output of the second seed-shift operation for $\mathcal{M}_{\text{Opt}}$ and $\mathcal{M}_{\text{CT}}$.



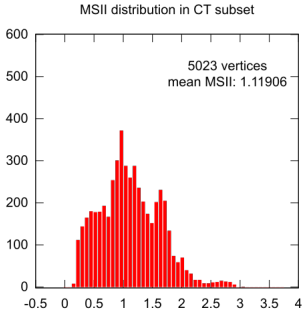
(A) Result for $\mathcal{M}_{\text{Opt}}$ (7.5k vertices)



(B) Result for $\mathcal{M}_{\text{CT}}$ (5k vertices)



(C) MSII of vertices in 5.3A



(D) MSII of vertices in 5.3B

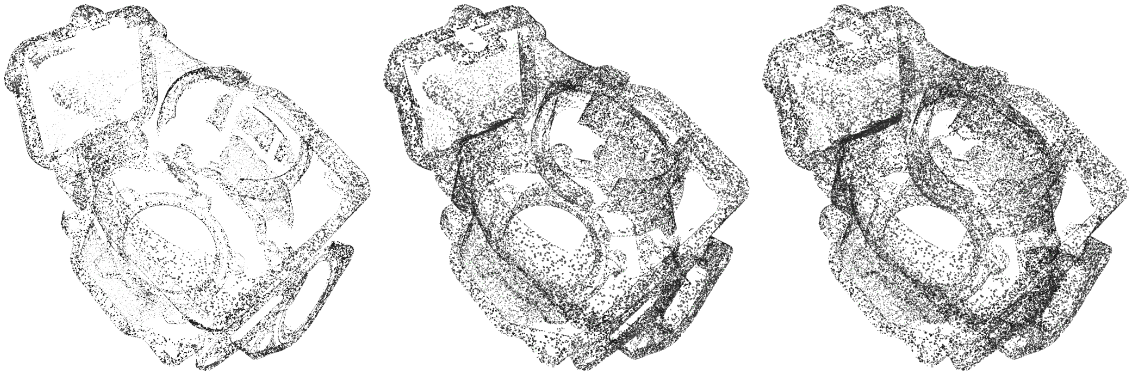
FIGURE 5.3: RanCEAF result for 50k seeds.

5.2 Alignment Comparison

The alignment of mesh-based object representations usually follows one of two principles, either continuously evaluating randomly generated transformations or iteratively

	Vertices (total)	Surface area (in cm^2)	Surface coverage ($in\%$)	Salient vertices (in % with MSII ≥ 1.0)	Mean salience (in subset via MSII)	CPU time (in sec)
Mesh from optical scan $\mathcal{M}_{\text{opt}}$	4813688	1042.9	69.19	12.47	0.444	–
Mesh from CT isosurface $\mathcal{M}_{\text{CT}}$	1303299	1507.3	100.00	4.94	0.349	–
Convex Hull $\mathcal{M}_{\text{CH}}$ of $\mathcal{M}_{\text{opt}}$	2903	0.5	0.04	67.30	1.343	35.8
Convex Hull $\mathcal{M}_{\text{CH}}$ of $\mathcal{M}_{\text{CT}}$	1785	4.4	0.29	57.45	1.114	8.7
Alpha Shape $\mathcal{M}_{\text{AS}}$ of $\mathcal{M}_{\text{opt}}$	449773	16.6	1.10	13.41	0.464	551.9
Alpha Shape $\mathcal{M}_{\text{AS}}$ of $\mathcal{M}_{\text{CT}}$	607004	593.7	39.39	1.17	0.326	135.5
RanCEAF subset of $\mathcal{M}_{\text{opt}}$	7474	1.9	0.13	64.78	1.229	37.4
RanCEAF subset of $\mathcal{M}_{\text{CT}}$	5023	13.1	0.87	55.15	1.119	7.1
RanCEAF 1^{st} seed-shift of $\mathcal{M}_{\text{opt}}$	38527	25.6	1.70	29.66	0.714	76.8
RanCEAF 1^{st} seed-shift of $\mathcal{M}_{\text{CT}}$	29802	90.7	6.02	23.87	0.643	14.1
RanCEAF 2^{nd} seed-shift of $\mathcal{M}_{\text{opt}}$	45954	40.2	2.67	17.42	0.497	116.3
RanCEAF 2^{nd} seed-shift of $\mathcal{M}_{\text{CT}}$	39958	129.0	8.56	17.80	0.488	21.3
RanCEAF 3^{rd} seed-shift of $\mathcal{M}_{\text{opt}}$	47499	47.1	3.12	11.45	0.392	156.6
RanCEAF 3^{rd} seed-shift of $\mathcal{M}_{\text{CT}}$	43671	144.9	9.61	16.29	0.410	28.1

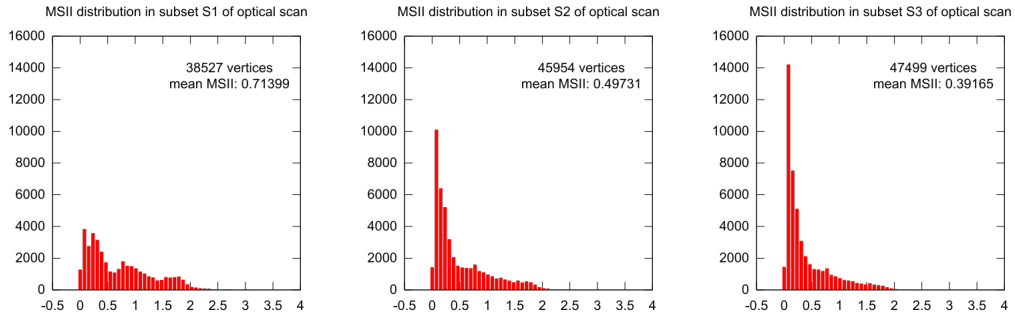
TABLE 5.1: Objects shown in Figure 1.2 after applying evaluated approaches.



(A) $\mathcal{M}_{\text{Opt}}$ subset: 1^{st} seed-shift

(B) $\mathcal{M}_{\text{Opt}}$ subset: 2^{nd} seed-shift

(C) $\mathcal{M}_{\text{Opt}}$ subset: 3^{rd} seed-shift



(D) MSII of vertices in 5.4A

(E) MSII of vertices in 5.4B

(F) MSII of vertices in 5.4C

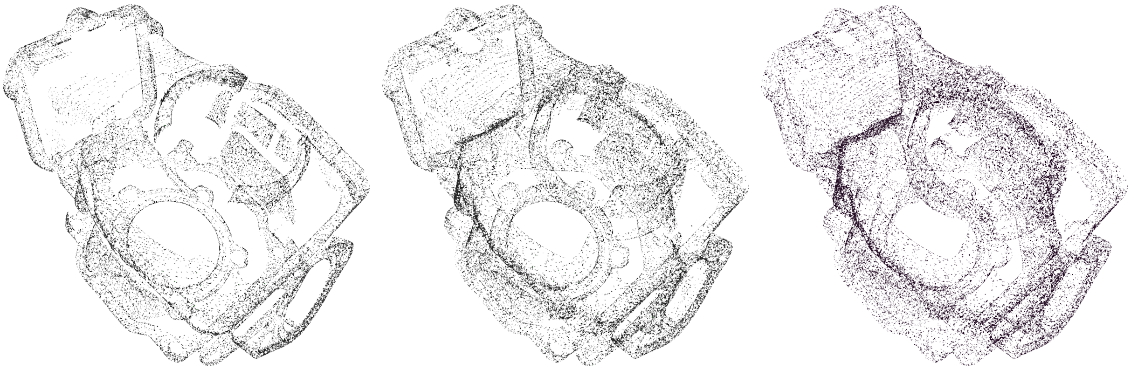
FIGURE 5.4: RanCEAF with seed-shifts applied to $\mathcal{M}_{\text{Opt}}$.

converging to a solution. Whereas the former is implemented in our project, the latter is applied via Meshlab[1].

5.2.1 RANdom Sample And Consensus

This approach is very reliable and converges quickly to a suitable solution if both meshes are from the same imaging technique or at least have similar spatial resolution. For $\mathcal{M}_{\text{Opt}}$ and $\mathcal{M}_{\text{CT}}$ this is generally not the case which either causes the absence of hypotheses at all due to the lack of sufficiently similar c vectors and therefore no convergence. In case the similarity condition and the verification threshold are relaxed, the approach converges to alignments which are not accurate enough for our scenario. Thus, the presented implementation of RANSAC is preferred to, e.g., align partial mesh representations, as an optical scanner provides them, to construct the complete scan result, but is not suited for aligning $\mathcal{M}_{\text{Opt}}$ and $\mathcal{M}_{\text{CT}}$.

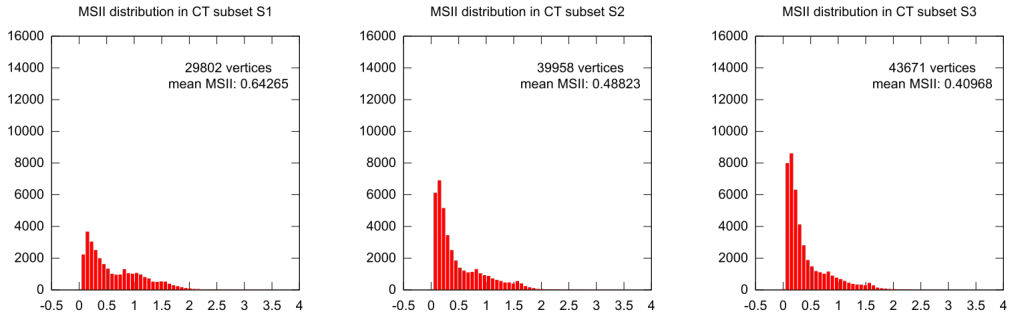
[1]Software provided by: Visual Computing Lab, CNR-ISTI, Pisa, Italy: `http://meshlab.sourceforge.net/`



(A) $\mathcal{M}_{\text{CT}}$ subset: 1^{st} seed-shift

(B) $\mathcal{M}_{\text{CT}}$ subset: 2^{nd} seed-shift

(C) $\mathcal{M}_{\text{CT}}$ subset: 3^{rd} seed-shift



(D) MSII of vertices in 5.5A

(E) MSII of vertices in 5.5B

(F) MSII of vertices in 5.5C

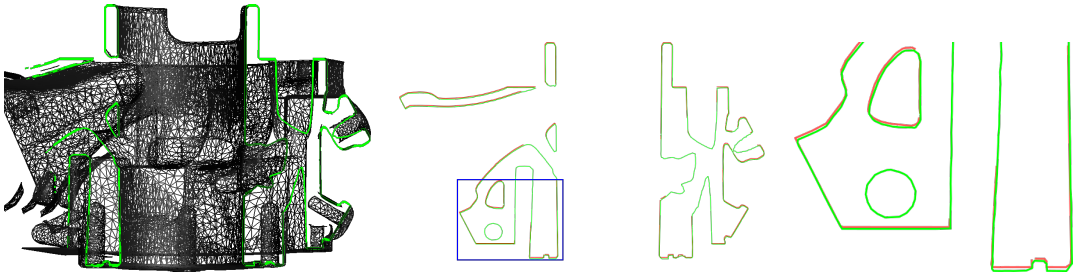
FIGURE 5.5: RanCEAF with seed-shifts applied to $\mathcal{M}_{\text{CT}}$.

5.2.2 Iterative Closest Point

The approach works fine for data points generated from the same imaging technique and, contrarily to RANSAC, does not seem to suffer from the difference in spatial resolution for $\mathcal{M}_{\text{Opt}}$ and $\mathcal{M}_{\text{CT}}$. On the downside, due to the imbalance of information as described above, ICP tends to introduce a drift in the resulting transformation. This is caused by the internal structures only represented in $\mathcal{M}_{\text{CT}}$ and the attempt to minimize the distance per vertex between the meshes. Since those vertices do not have a suitable counterpart in $\mathcal{M}_{\text{Opt}}$ the introduced drift can be seen as over-compensation. Figure 5.6 presents the offset as cross sections of both meshes.

5.2.3 Random Convex-Edge Affine Features for Alignment

We have shown that RanCEAF efficiently identifies the exterior surface of a given mesh. Furthermore, it allows to over-represent convex areas since they serve as attractors for regional queries from seeds on the enclosing ellipsoid. The protruding areas include the local maxima of the object under investigation and the resulting subset of all data



(A) Cross section of $\mathcal{M}_{\mathrm{CT}}$

(B) Cross sections of aligned meshes

(C) Zoom to blue box in (b)

FIGURE 5.6: ICP offset of cross sections from $\mathcal{M}_{\mathrm{CT}}$ (green) and from $\mathcal{M}_{\mathrm{Opt}}$ (red).

points is suitable for alignment. The presented approach does not—in contrast to AS—introduce additional faces or require any further post-processing. The proposed method provides a reliable surface reduction, which can be iteratively expanded by applying multiple seed-shift operations. In general, the RanCEAF algorithm only relies on the vertices of the mesh $\mathcal{M}$ and therefore can be applied to point clouds. Only for the analysis based on MSII, faces are required in a pre-processing step and only for the sake of evaluating our approach. The comparison of size and salience of extracted subsets, as shown in Table 5.1, indicates that the subsets extracted by our method are sufficiently large to serve as input for computing an alignment, and yet salient enough to grasp the essential structures of the presented geometry. The inherent parallelism of our approach is easily exploited (in our evaluation on an *Intel Xeon E7-4870*)and therefore not corrected for comparison to single-threaded algorithms in Table 5.1. For the presented object, the alignment of the complete meshes $\mathcal{M}_{\mathrm{Opt}}$ and $\mathcal{M}_{\mathrm{CT}}$ via the ICP algorithm in Meshlab resulted in an RMSE of 2.736 mm. Computing the transformation matrix based on the extracted surfaces of both meshes and applying the obtained transformation to $\mathcal{M}_{\mathrm{Opt}}$ and $\mathcal{M}_{\mathrm{CT}}$, provided a RMSE of 2.722 mm. The increase in accuracy reads as 0.5% or an RMSE reduction of 14 μm, which potentially affects the selection of cells on the dense voxel grid as reconstructed from CT scans. Notice that there is no perfect alignment for both meshes. Therefore, the RMSE cannot be zero and the real increase in accuracy is higher than 0.5%.

5.2.4 Octants' Center of Gravity Alignment

The presented alignment method OctaCoG, operating on RanCEAF surface subsets of the respective meshes, is tested against ICP. Alignments were computed by both algorithms from identical input data and the resulting transformation matrix applied to the corresponding full meshes for RMSE measurement. In addition the full meshes were aligned via ICP including RMSE measurement. All ICP runs initially require

TABLE 5.2: Run time and RMSE of alignment approaches for optical scan and CT scan. RMSE is measured for full meshes. "Correctness" is fraction of current versus known best solution. Processing time is single threaded.

	Name Angular range	LACT sample (0°..60°)	LACT sample (120°..180°)	Sample (360°)	Industrial (360°)	Wing (synth)
ICP (meshes)	RMSE [mm]	0.747	1.251	0.289	1.935	0.155
	run time [s]	14.520	23.840	10.350	57.860	0.340
	correctness [%]	95.96	85.03	84.85	94.03	63.40
ICP (subsets)	RMSE [mm]	0.737	1.274	0.315	1.917	0.115
	run time [s]	0.530	0.270	0.450	0.720	0.140
	correctness [%]	97.33	83.49	77.77	94.95	85.47
OctaCoG (subsets)	RMSE [mm]	0.734	1.134	0.245	1.820	0.099
	run time [s]	0.187	0.155	0.203	0.498	0.073
	correctness [%]	**97.74**	**93.77**	**100.0**	**100.0**	**100.0**

user interaction for selecting appropriate reference points, this step is not necessary for OctaCoG. For comparison, the computation time for each alignment is measured as accumulated CPU cycles divided by CPU clock frequency. It corresponds to single threaded processing on a *Intel® i7-4770* and is averaged over 100 runs. We would like to point out, that OctaCoG is designed for parallel execution, and on multicore systems results are provided in a fraction of the time given in Table 5.2.

LACT scans are indicated by denoting the respective angular range, while 360° corresponds to a complete circular trajectory. OctaCoG run time includes PCA calculation, assignment of vertices to octants, computation of barycenters and the comparison of two OctaCoG shapes to find the final transformation. Besides time and RMSE, also the fraction of current RMSE and known best solution per object is presented as "correctness". In the lack of ground truth for the alignment, we set the lowest RMSE value of all methods as best result—although a better alignment with even lower RMSE might exist. For limited angle trajectories, the known best RMSE results from applying the correct 360° alignment, thus none of the approaches achieve 100%. The presented results in Table 5.2 show, that our approach is roughly twice as fast compared to subset processing via ICP and yields a RMSE improvement of 15% in the case of the wing model or full-angle scan of the sample object. RMSE is up to 8% better for aligning LACT data and 5% for the industrial object.

5.3 CMM to Mesh Conversion

An example of successful conversion from CMM (in Figure 4.7B) results to a closed surface is shown in Figure 5.7. For simplicity the current implementation neglects all cylinders, in principle they do not change the combinatorics of the problem and can be added at a later stage as rounded corners and drilling holes. In the current approach, results for cylinders and the associated cylinder bottom plane are automatically removed. Adding them in the final representation is of course intended to fully enable comparisons to other acquisition results.

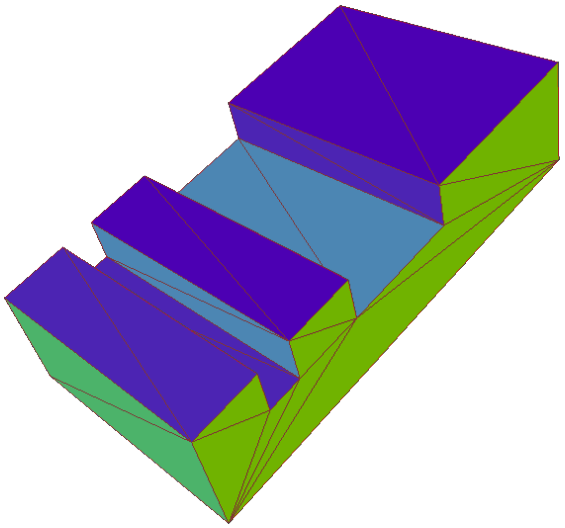


FIGURE 5.7: Resulting mesh $\mathcal{M}_{\text{CMM}}$ from measurement results (neglecting cylinders).

5.4 Gradient Assessment Result

Isosurface areas robust against small changes in the threshold are located at steep local gradients, respective paths descent quickly to low density values. Defective areas with LACT artifacts correspond to paths slowly descending or ascending. This does not permit a quantitative analysis, but is sufficient to indicate surface robustness to an alignment algorithm or to determine more suitable thresholds for isosurface extraction. Applied to the sample in Figure 4.11, the corresponding coloring depicting surface robustness by length of descending paths is shown in Figure 5.8.

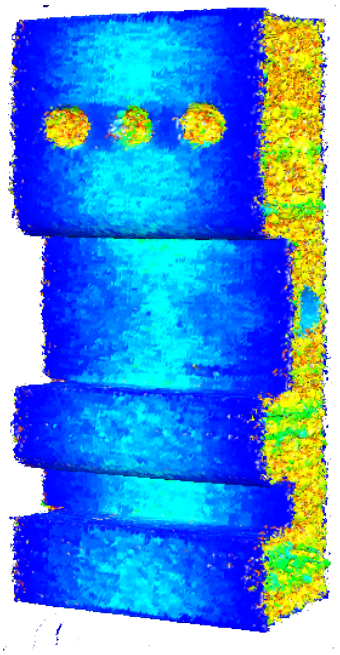


FIGURE 5.8: Coloring indicating surface robustness by voxel distance from density at surface to a density near the global minimum.

Chapter 6

Discussion, Conclusions and Future Work

For each method, the question has to be answered if the presented approach is sufficient and how it can be improved. An outlook for each approach and the project as a whole is given to indicate further areas of application or potential continuations.

6.1 Random Convex-Edge Affine Features

Our RanCEAF approach extracts the exterior surface layer from meshes or point clouds via nearest neighbor searches (NNS) [BLMK16]. This method is intended to mimic the acquisition scheme of an optical scanner. Therefore, surface areas similar to those an optical scanner can acquire are preserved during extraction. Interior surfaces as represented in CT scans are neglected, and also data points in narrow cavities or on exterior surfaces behind undercuts. Applied to $\mathcal{M}_{\mathrm{CT}}$, the extracted point cloud $\mathcal{P}_{\mathrm{CT}}$ not only lacks interior structures but also areas on the exterior surface which are unlikely to be accessible for an optical scanner. To generate a surface subset, seed vertices s for NNS queries are randomly distributed on an ellipsoid enclosing the data set. The proximity search from seed vertices s then returns points p of the object representation. Since all seeds are located sufficiently far from the object, it is guaranteed that identified vertices p are located on the exterior surface. Protruding structures, or in general extreme points of the mesh, are likely to dominate the result set by being nearest neighbor of multiple seed vertices. To bypass those structures in order to identify more points p for the extracted subset, so-called "seed-shift" operations are applied. They iteratively expand the set $\mathcal{P}$ by shifting all seeds s towards the center of the ellipsoid by the distance $|\overline{sp}|$ to their respective nearest neighbor—which restricts the penetration of the exterior

surface. Subsequent NNS queries from shifted seeds identify further exterior vertices p. The resulting RanCEAF extractions $\mathcal{P}_{\mathrm{OPT}}$ and $\mathcal{P}_{\mathrm{CT}}$ as in Figures 4.4 and 4.5 enable a ICP alignment with reduced errors or serve as input for our OctaCoG approach, yielding a more accurate alignment in a fraction of the time.

6.1.1 Future Directions for RanCEAF

Although the described imbalance in information, contained in $\mathcal{M}_{\mathrm{Opt}}$ and $\mathcal{M}_{\mathrm{CT}}$, could be mitigated by the presented approach, the mismatch in resolution of both imaging techniques still presents a challenge to alignment algorithms. In the further pursuit of our work, our focus will be to investigate alignment schemes which do not rely on point to point comparison for registration. Instead of performing seed-shift operations for all seed vertices alike, adaptive application to selected seeds, based on the local geometry, would reduce runtime and preserve more features. The fact that both representations are known to describe the very same object and that they also both contain the object as a whole, matches with the challenges within our joint project ILATO[4]. Especially for dealing with artifacts from Limited-Angle CT, any data point irrelevant for alignment has to be neglected since the registration based on the remaining exterior surface points is already very difficult. For metrology applications and industrial quality inspections, technical drawings of the specimen are available as CAD files. Registration of an optical scan surface and CAD, which contains interior structures, can benefit from the presented approach. Likewise, coordinate-measuring machines (CMM) provide highly accurate tactile measurements of an object's surface. Registering the CMM output with $\mathcal{M}_{\mathrm{CT}}$ of this object can be enhanced by neglecting the interior structures of $\mathcal{M}_{\mathrm{CT}}$. We will pursue further investigations to estimate the minimal RMSE for the given alignment depending on resolution and fidelity of the acquisition systems and to determine the actual increase in accuracy our approach provides.

6.2 Alignment via OctaCoG

We have shown that the characteristic OctaCoG shape, each Octants' Centers of Gravity, per point cloud is unambiguous and permits fast and highly accurate alignment. Our presented approach takes advantage of the fact that the data sets contain information about the very same specimen and can be subsampled to a common representation. It has already been shown, that the accuracy of classical approaches for aligning $\mathcal{M}_{\mathrm{Opt}}$ and $\mathcal{M}_{\mathrm{CT}}$ suffer from the mentioned imbalance in resolution and contributing surfaces of

[4] `http://www.iwr.uni-heidelberg.de/groups/ngg/ILATO/`

both representations. For RANSAC approaches, the difference in spatial resolution poses a challenge in formulating transformation hypotheses in the first place. ICP approaches can deal with the difference in resolution, but in the attempt to minimize the quadratic error of vertex distances between the two meshes, tend to overcompensate for interior structures. Especially for double walled specimen those are present in the CT output and cause a misalignment which manifests as a drift. With RanCEAF we developed an algorithm to extract a common subset of both representations as a point cloud of the exterior surface. Solving the alignment of these subsets—which is a less complex problem than solving it for the full data sets—yields a transformation which is also a valid alignment of the original representations. The subset extraction was originally developed to improve the alignment accuracy of ICP approaches in finding a valid transformation. But under the assumption that we obtain complete scans of the very same object, we can apply OctaCoG without the need of iteratively converging towards a transformation matrix. We have shown, that based on the existing surface subsets, a highly accurate alignment can be found in $\mathcal{O}(n)$. The applied method also allows for small differences in scale of the respective representations, as they definitely occur for CT.

6.2.1 Future Directions for OctaCoG

In the continuation of this work, we will apply our approach to LiDAR data. In contrast to our current use case, registration of scan segments and drift compensation does not rely on a pre-processing step to generate suitable subsets. To improve the alignment of LACT data with optical scans, we aim at including prior knowledge about artifact distribution in the isosurface, e.g., from Gradient Assessment,to reflect this in the OctaCoG shape.

6.3 Gradient Assessment

The identification of reliable and LACT artifact compromised regions in the isosurface or the volumetric representation is an ongoing project. The intermediate results presented in Section 5.3 are based on local gradient computation and analyze the vicinity of each voxel. Qualitative statements about the robustness of certain areas against changes in the isosurface threshold are already possible and even corrections to the volumetric data can be applied.

6.3.1 Future Directions for Gradient Assessment

The concept of computing paths of steepest descent is very intuitive and obviously suitable to analyze the scalar field close to the isosurface. For a more general approach and eventually considering multiple thresholds to identify the interface between material and air, we will apply the Sobel operator. The current path computation only considers six neighbors per voxel and in the resulting paths consecutive edges are either straight or orthogonal. On the one hand, Sobel operators will permit adjacent edges with arbitrary angles. On the other hand, we aim to compare gradients above the isosurface with gradients below the isosurface to verify the plausibility of the selected threshold. The latter case might also result in identifying the proper location of the isosurface by providing an alternate set of *Face Center Points* as introduced in Section 3.2.2.

6.4 CMM to Mesh Conversion

CMM measurements have proven to be an excellent tool for bridging the gap between tactile measurements and image acquisition techniques. Their superior accuracy offered an unbiased view on flaws during specimen production and the measurement errors of CT and optical scans. The tedious conversion of a mesh representation in a set of individual distance measures is overcome by instead converting the CMM measurement in a mesh. This permits to apply standard approaches like Hausdorff distance or RMSE computation to evaluate deviations.

6.4.1 Future Directions for CMM conversion

The presented result is final except for the inclusion of cylindricity measures to represent drilling holes and rounded corners. The essential properties of of the measurement are preserved in the resulting mesh $\mathcal{M}_{\text{CMM}}$ and the coloring corresponds to deviations from planar or cylindrical shape. The same deviations are also stored in each point of the mesh.

6.5 Outlook for the ILATO Project as a Whole

The presented approach was developed with industrial quality inspection in mind. Of course other fields of work also facilitate multiple imaging techniques and therefore can benefit from our solution. Projects reconstructing internal building structure from Wifi attenuation measurements [Mon09] combine this measurement with prior knowledge.

They provide blueprints of the internal building structure or the known exterior surface of the building under investigation to enhance the resulting representation. For sure this approach fits our profile. Other projects in the non-destructive testing domain combine ultrasound imaging and surface information to discover cracks in concrete blocks or pipes. Agin, this is a possible usecase for our methods. Any combination of transmission-based measurement and distance-based measurement potentially benefits from the presented methods.

6.5.1 Incorporate more Knowledge from CT Acquisition

Reconstruction algorithms struggle to solve underdetermined systems as they occur for limit-angle CT. Yet, the resulting scalar field does not convey any uncertainty or error measures and cannot be easily distinguished from a reconstruction result from full angular trajectory. The local measurement errors and distortions identified during reconstruction would allow to work with another set of attributes similar to the local robustness value from *Gradient Assessment*.

6.5.2 Incorporate more Knowledge from Optical Acquisition

Introducing clipping information in volumetric representation or per projection is one way to express the absence of material in certain areas of the data set as indicated by optical scans. Since these scans also have defects like partially missing surface information due to reflections, matte black coloring or other harsh conditions the clipping is incomplete. Closing the resulting holes as discussed in Section 3.1.1 always introduces unvalidated information with cannot directly be used to generate masks. If we could exploit the equivalent of a trajectory description for optical scans, the overlapping camera views could help to identify *more air* around the surface acquired by the optical scanning system. This information is provided in our Breuckman setup but currently cannot be exported. The knowledge of the exact camera position for each partial scan would provide new means of assessing the merged representation $\mathcal{M}_{\mathrm{Opt}}$ in terms of visibility of surface features and more important the absence of those.

6.6 Discarted Approaches

Failed attempts to compute an alignment solely based on PCA, MVBB and intersection comparisons in an early stage of the project are documented in Appendix F. Although

those approaches were discarded, they motivated the works on *RanCEAF* and *OctaCog*, presented in Section 4, and shall therefore be included in this thesis.

An evaluation between RANSAC and ICP led to the implementation of a new way to formulate hypotheses—not based on point pairs and their property vector as in Section 3.5.1, but small surface patches of similar area. The presented *Stencil selection* was intended to select those patches, the comparison of two patches would serve as basis for hypothesis as currently two point-pairs and their 4D property verctor. Non-Uniform Rational B-Splines should provide parametric forms as *NURBS surface* which are to be characterized by a similar property vector. For more details, see Appendix F.

Appendix A

CMM Measurements

Fitting planes and cylinders to the respective measurement points in the CMM protocol (Table A.1) results in measurement summary in Table A.2. The naming convention for planes and cylinders is provided in Figure A.1.

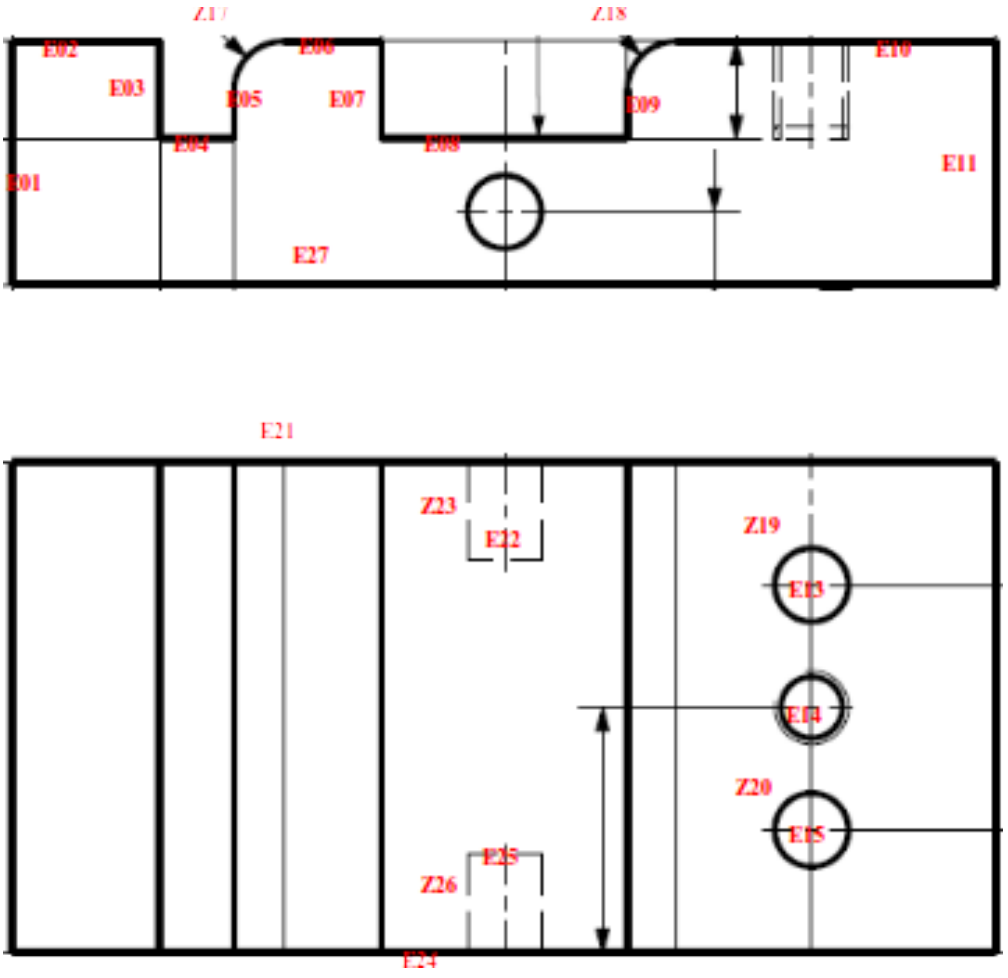


FIGURE A.1: Labeling of cylinders and planes as reference for CMM measurement.

TABLE A.1: Excerpt of CMM scan protocol

MEAS	THEO	DEV	UT	LT	OOT
Sample	1B				
========					
Coorinate	system	established			
==========	========	========			
Inspection					
#	5	PLANE	Ref.	Sys	1
X	-0.0015				
Y	20.1744				
Z	-5.9025				
CX	1				
CY	-0.00019				
CZ	-0.000404				
FLATNES	0.0046				
Inspection					
#	6	PLANE	Ref.	Sys	1
X	-6.1279				
Y	20.8282				
Z	0.0002				
CX	-0.000088				
CY	-0.000007				
CZ	1				
FLATNES	0.0006				
Inspection					
#	7	PLANE	Ref.	Sys	1
X	-11.9863				
Y	19.1587				
Z	-3.7132				
CX	-1				
CY	0.000562				
CZ	-0.000009				
FLATNES	0.0027				
Inspection					
#	8	PLANE	Ref.	Sys	1
X	-15.0728				
Y	20.2732				
Z	-7.9994				
CX	-0.000194				
CY	-0.000026				
CZ	1				
FLATNES	0.0009				
Inspection					
#	9	PLANE	Ref.	Sys	1
X	-17.9859				
Y	19.8042				
Z	-5.6183				
CX	0.999999				
CY	-0.000542				
CZ	0.001263				
FLATNES	0.0012				
Inspection					
#	10	PLANE	Ref.	Sys	1
X	-25.4378				
Y	20.5211				
Z	-0.0013				
CX	0.000004				
CY	0.000032				
CZ	1				
FLATNES	0.002				
.....	.....	.....	.....	...	.....
.....	.....	.....	.....	...	.....
.....	.....	.....	.....	...	.....
Inspection					
#	35	CYLINDER	Ref.	Sys	1
X	-39.9488				
Y	36.1708				
Z	-14.0021				
CX	0				
CY	1				
CZ	0				
DM	6.0004				
CYLINTY	0.0074				
Inspection					
#	36	PLANE	Ref.	Sys	1
X	-42.138				
Y	19.8438				
Z	-19.9777				
CX	0.000365				
CY	0.000645				
CZ	-1				
FLATNES	0.0034				
========	========	========	====_x000C_		

TABLE A.2: CMM output for sample 1B

ID	#	X	Y	Z	CX	CY	CZ	FLATNES	
E01	12	-0.001	20.089	-5.660	1.000000	-0.000214	-0.000369	0.007	
E02	13	-5.605	17.943	0.001	-0.000167	-0.000045	1.000000	0.001	
E03	14	-11.987	19.378	-3.581	-1.000000	0.000560	-0.000412	0.001	
E04	15	-14.824	19.518	-7.999	0.000101	0.000011	1.000000	0.001	
E05	16	-17.985	20.008	-5.448	0.999999	-0.000561	0.000941	0.003	
E06	17	-25.907	20.164	-0.001	-0.000037	-0.000015	1.000000	0.000	
E07	18	-29.979	19.296	-3.436	-1.000000	0.000585	0.000698	0.003	
E08	19	-40.222	19.797	-7.997	0.000141	0.000062	1.000000	0.003	
E09	20	-49.981	20.037	-5.390	1.000000	-0.000534	-0.000272	0.001	
E10	21	-66.071	19.590	0.001	-0.000006	0.000024	1.000000	0.000	
E11	22	-79.915	19.209	-5.567	-0.999995	-0.001226	0.002808	0.016	
E13	23	-64.887	9.988	-8.002	0.002536	-0.004807	0.999985	0.020	
E14	24	-64.510	20.311	-7.992	0.006275	0.007413	0.999953	0.011	
E15	25	-64.867	30.025	-8.004	-0.004774	0.001010	0.999988	0.018	
E21	30	-43.846	-0.005	-8.604	-0.000030	-1.000000	0.000113	0.004	
E22	31	-40.044	7.991	-14.027	-0.006869	-0.999965	-0.004815	0.018	
E24	33	-34.006	39.970	-8.965	0.000298	1.000000	0.000381	0.011	
E25	34	-39.991	31.981	-14.253	0.003481	0.999847	-0.017164	0.008	
E27	36	-42.138	19.844	-19.978	0.000365	0.000645	-1.000000	0.003	
		X	Y	Z	CX	CY	CZ	DM	CYLINTY
Z17	26	-21.982	20.901	-3.997	0.000000	1.000000	0.000000	7.988	0.019
Z18	27	-53.986	19.921	-4.003	0.000000	1.000000	0.000000	8.007	0.017
Z19	28	-64.978	10.006	-3.493	0.000000	0.000000	1.000000	6.005	0.002
Z20	29	-64.975	30.002	-3.616	0.000000	0.000000	1.000000	6.007	0.005
Z23	32	-40.028	4.162	-14.013	0.000000	1.000000	0.000000	6.006	0.009
Z26	35	-39.949	36.171	-14.002	0.000000	1.000000	0.000000	6.000	0.007

Appendix B

Random Point on Sphere Evaluation

Random vertices on unit sphere according to Equation (4.4) and corresponding local linearity measure l according to Equation (2.12). Radomized u and v with (intuitively correct) ranges $u \in (0..1)$ and $v \in [0..1)$ result in mean linearity $\bar{l} = \frac{\sum l}{n} = 0.000539808$ over $n = 50$ runs with 1.000.000 vertices per sphere (see Listing B.2). The better local linerity of $\bar{l} = 0.000504256$ is achived in boundaries $u \in (0..1)$ and $v \in (0..1)$.

In the 3D case, a perfectly uniform distribution of normal vectors results in three identical eigenvalues $\lambda_i = \frac{1}{3}$ from PCA. The following output lists the difference to the idea results and read as
$l - \frac{1}{3}$ `with` $\lambda_1 - \frac{1}{3}$ `/` $\lambda_2 - \frac{1}{3}$ `/` $\lambda_3 - \frac{1}{3}$
and for ideal runs should print
0.0 `with` 0.0 `/` 0.0 `/` 0.0

```
0.000458761 with -0.000542214/8.30109e-05/0.00045854
0.000598077 with -0.000404531/-0.000194917/0.000597389
0.000551782 with -0.000656291/0.000103655/0.000551354
0.000304746 with -0.000572415/0.000267023/0.000304423
0.000805091 with -0.000695703/-0.000109909/0.00080483
0.000619941 with -0.000476353/-0.000143696/0.000619886
0.000431677 with -0.000482464/5.0545e-05/0.000431555
0.000607387 with -0.00105757/0.000449649/0.00060712
0.000518047 with -0.000501038/-1.72582e-05/0.000517922
0.000799118 with -0.000605226/-0.00019459/0.000798767
0.000472661 with -0.000317667/-0.000155163/0.000472576
0.000235037 with -0.000179021/-5.66046e-05/0.000234743
0.000502147 with -0.00033572/-0.000166717/0.000502002
0.000713865 with -0.000675613/-3.83235e-05/0.00071383
0.00060367 with -0.000750414/0.000146409/0.000603502
0.000819154 with -0.00071337/-0.000106795/0.000818647
0.000989334 with -0.000659632/-0.000331271/0.000988546
0.000278509 with -0.000466782/0.000187573/0.000278158
0.000489871 with -0.000528692/3.84999e-05/0.00048971
0.000569621 with -0.000576072/6.29552e-06/0.000569543
0.00050766 with -0.000638862/0.000131031/0.000507574
0.000450431 with -0.000477186/2.48024e-05/0.000449453
0.000371255 with -0.000393737/2.20649e-05/0.000371046
0.000755066 with -0.000712088/-4.46658e-05/0.000754219
0.000414773 with -0.00041018/-5.41628e-06/0.00041436
0.000626126 with -0.000467613/-0.000158904/0.00062593
0.000480621 with -0.000705528/0.000224593/0.000480463
0.000770888 with -0.000517685/-0.000253505/0.000770736
0.000668518 with -0.00114299/0.000474257/0.000668414
0.000290702 with -0.000466913/0.000175965/0.000290578
0.000652803 with -0.000746113/9.15125e-05/0.000651902
0.00024101 with -0.000386991/0.000145214/0.000240627
0.000336418 with -0.000538518/0.000201528/0.000336131
0.000384608 with -0.000678202/0.000292828/0.000384224
0.000520965 with -0.000376511/-0.000144601/0.000520892
0.000781222 with -0.000588006/-0.000193432/0.000781115
0.000832879 with -0.000789385/-4.46337e-05/0.000832308
0.00014448 with -0.000161967/1.64748e-05/0.000143974
0.000450853 with -0.000375449/-7.57781e-05/0.000450665
0.000418244 with -0.000294891/-0.000124158/0.000417841
0.000526346 with -0.000575338/4.88705e-05/0.000526285
0.000375242 with -0.00048863/0.000113309/0.000375202
0.000487441 with -0.000247097/-0.000240803/0.000487211
0.000600018 with -0.000399138/-0.000201503/0.000599706
0.000594602 with -0.000514461/-8.02784e-05/0.000594534
0.001059 with -0.000810314/-0.000249688/0.0010585
0.000150074 with -0.000199001/4.87131e-05/0.000149967
0.000193623 with -0.000154133/-4.04911e-05/0.000193122
0.00059235 with -0.000509171/-8.44866e-05/0.000591695
0.000943705 with -0.000807893/-0.000136199/0.00094351

linearity/n: 0.000539808
```

LISTING B.1: $u \in (0..1)$ and $v \in [0..1)$.

```
0.000439513 with -0.00049265/5.20013e-05/0.000438944
0.000514412 with -0.000528082/1.34861e-05/0.00051432
0.000293239 with -0.000270009/-2.4632e-05/0.000292537
0.000387827 with -0.000433474/4.45945e-05/0.0003873
0.000273935 with -0.000201955/-7.22947e-05/0.000273777
0.000541038 with -0.000467375/-7.38837e-05/0.000540928
0.00038476 with -0.000597899/0.000212369/0.000384374
0.000329248 with -0.000250951/-7.91096e-05/0.000328841
0.000346733 with -0.000297767/-4.907e-05/0.000346681
0.000233959 with -0.000185221/-4.95941e-05/0.000233531
0.000448863 with -0.000561717/0.000112673/0.000448773
0.000829833 with -0.000860197/2.97736e-05/0.000829538
0.000252637 with -0.000352863/9.99807e-05/0.000252514
0.000171735 with -0.000294219/0.000122266/0.000171626
0.000862343 with -0.000996443/0.000133604/0.000862094
0.000458057 with -0.000401382/-5.73683e-05/0.00045771
0.000286778 with -0.000249302/-3.79934e-05/0.000286518
0.000208657 with -0.000269235/6.02604e-05/0.000208497
0.000650525 with -0.000504547/-0.000146117/0.000650455
0.00065042 with -0.000473106/-0.000177361/0.000650396
0.000360621 with -0.00056194/0.000201148/0.000360536
0.000791153 with -0.000877396/8.59261e-05/0.000790993
0.000313664 with -0.000268015/-4.68234e-05/0.000313076
0.00068947 with -0.000854142/0.000164126/0.000689196
0.00039815 with -0.000290845/-0.000108165/0.000397719
0.000943449 with -0.000653696/-0.000290328/0.000943161
0.000501634 with -0.000284141/-0.000218514/0.000501123
0.000975335 with -0.00101009/3.16631e-05/0.000973781
0.000434933 with -0.000278377/-0.00015765/0.000434385
0.000351531 with -0.000494183/0.000141733/0.000351072
0.000741022 with -0.000669069/-7.38043e-05/0.000740093
0.00046394 with -0.000399173/-6.52391e-05/0.000463704
0.000467051 with -0.000374514/-9.35374e-05/0.00046655
0.000598356 with -0.00082334/0.000224675/0.000598201
0.00039791 with -0.000413518/1.49748e-05/0.000397593
0.000294534 with -0.000276016/-1.9973e-05/0.000293805
0.000744112 with -0.00057448/-0.000170055/0.0007439
0.000998561 with -0.00062941/-0.0003694/0.000998436
0.000464268 with -0.000422747/-4.25096e-05/0.000463773
0.000377639 with -0.000459633/8.15645e-05/0.000377424
0.000816495 with -0.00105251/0.000235843/0.000816409
0.00042689 with -0.000420825/-7.66787e-06/0.000426087
0.000523346 with -0.000446412/-7.71574e-05/0.000523234
0.000642073 with -0.00053341/-0.000109528/0.000641639
0.000368266 with -0.000397281/2.88405e-05/0.000368179
0.000480201 with -0.000381177/-9.98808e-05/0.000479772
0.000554925 with -0.000574747/1.95142e-05/0.00055477
0.000393521 with -0.000461835/6.71256e-05/0.000392925
0.000777933 with -0.000534822/-0.000243467/0.000777754
0.000357316 with -0.000420914/6.34484e-05/0.00035724

lineatriy/n: 0.000504256
```

LISTING B.2: $u \in (0..1)$ and $v \in (0..1)$.

Appendix C

RanCEAF Run-time Comparison

The following tables hold all measurement results for evaluating the performance RanCEAF subset extraction. An excerpt is shown in Section 5.2.3.

	Vertices (total)	Surface area (in cm^2)	Surface coverage ($in\%$)	Salient vertices (in % with MSII ≥ 1.0)	Mean salience (in subset via MSII)	CPU time (in sec)
Mesh from optical scan $\mathcal{M}_{\text{opt}}$	4813688	1042.9	69.19	12.47	0.444	–
Mesh from CT isosurface $\mathcal{M}_{\text{CT}}$	1303299	1507.3	100.00	4.94	0.349	–
Convex Hull $\mathcal{M}_{\text{CH}}$ of $\mathcal{M}_{\text{opt}}$	2903	0.5	0.04	67.30	1.343	35.8
Convex Hull $\mathcal{M}_{\text{CH}}$ of $\mathcal{M}_{\text{CT}}$	1785	4.4	0.29	57.45	1.114	8.7
Alpha Shape $\mathcal{M}_{\text{AS}}$ of $\mathcal{M}_{\text{opt}}$	449773	16.6	1.10	13.41	0.464	551.9
Alpha Shape $\mathcal{M}_{\text{AS}}$ of $\mathcal{M}_{\text{CT}}$	607004	593.7	39.39	1.17	0.326	135.5
RanCEAF subset of $\mathcal{M}_{\text{opt}}$	7474	1.9	0.13	64.78	1.229	37.4
RanCEAF subset of $\mathcal{M}_{\text{CT}}$	5023	13.1	0.87	55.15	1.119	7.1
RanCEAF 1^{st} seed-shift of $\mathcal{M}_{\text{opt}}$	38527	25.6	1.70	29.66	0.714	76.8
RanCEAF 1^{st} seed-shift of $\mathcal{M}_{\text{CT}}$	29802	90.7	6.02	23.87	0.643	14.1
RanCEAF 2^{nd} seed-shift of $\mathcal{M}_{\text{opt}}$	45954	40.2	2.67	17.42	0.497	116.3
RanCEAF 2^{nd} seed-shift of $\mathcal{M}_{\text{CT}}$	39958	129.0	8.56	17.80	0.488	21.3
RanCEAF 3^{rd} seed-shift of $\mathcal{M}_{\text{opt}}$	47499	47.1	3.12	11.45	0.392	156.6
RanCEAF 3^{rd} seed-shift of $\mathcal{M}_{\text{CT}}$	43671	144.9	9.61	16.29	0.410	28.1

TABLE C.1: Objects shown in Figure 1.2 after applying evaluated approaches on an *Intel Xeon E7-4870*.

Appendix D

OctaCoG Evaluation

The following tables hold all measurement results for evaluating OctaCoG performance. An excerpt is shown in Section 5.2.4.

TABLE D.1: Run time and RMSE of alignment approaches for optical scan and CT scan. RMSE is measured for full meshes. "Correctness" is fraction of current versus known best solution. Processing time is single threaded.

		1A LACT983			1A LACT323			1A fullCT	Cast	MTU	Wing
angular range		0°..60°			120°..180°			360°	360°	360°	
myName		85	86	87	1	2	3	1A_CT	Alu_CT	MTU_CT	airfoil
vertices (full)		762502	832134	743760	352442	293856	290398	732338	1333364	10256674	38596
Vertices (subset)		46197	58866	50460	38293	38450	38653	118398	182742	564614	5006
isovalue		0.020	0.018	0.015	0.020	0.010	0.008	0.020	7000	2939	synth
ICP(subsets)	time_subset (s)	0.310	0.700	0.530	0.610	0.460	0.270	0.450	0.630	0.720	0.140
	RMSE_subset (mm)	1.291	1.552	0.886	1.117	2.288	1.814	0.342	0.462	1.489	0.388
	RMSE_full (mm)	1.285	0.959	0.737	0.990	2.169	1.274	0.315	0.394	1.917	0.115
	delta2FT_rel (%)	97.98%	50.69%	97.33%	80.30%	34.14%	83.49%	77.77%	87.12%	94.95%	85.47%
	delta2FT_abs (mm)	0.026	0.473	0.020	0.195	1.428	0.210	0.070	0.051	0.097	0.017
ICP (full)	time (s)	22.340	15.950	14.520	19.820	9.330	23.840	10.350	22.620	57.860	0.340
	RMSE_subset (mm)	—	—	—	—	—	—	—	—	—	—
	RMSE_full (mm)	1.624	0.522	0.747	1.249	0.861	1.251	0.289	0.355	1.935	0.155
	delta2FT_rel (%)	77.53%	93.13%	95.96%	63.66%	86.03%	85.03%	84.85%	96.77%	94.03%	63.40%
	delta2FT_abs (mm)	0.365	0.036	0.030	0.454	0.120	0.187	0.044	0.011	0.116	0.057
OctaCoG	time_subset (s)	0.178	0.166	0.187	0.184	0.16	0.155	0.203	0.335	0.498	0.073
	RMSE_subset (mm)	1.332	0.937	0.759	0.930	0.848	1.798	0.284	0.642	1.489	0.367
	RMSE_full (mm)	1.324	0.904	0.734	0.914	0.801	1.134	0.245	0.526	1.820	0.099
	delta2FT_rel (%)	95.10%	53.75%	97.74%	87.03%	92.47%	93.77%	100.00%	65.26%	100.00%	100.00%
	delta2FT_abs (mm)	0.065	0.418	0.017	0.119	0.060	0.071	0.000	0.183	0.000	0.000
ICP(OC-aligned)	time (s)	3.700	5.300	3.900	20.800	13.600	6.600	2.500	21.710	30.380	30.380
	RMSE_subset (mm)	—	—	—	—	—	—	—	—	—	—
	RMSE_full (mm)	1.524	0.685	0.763	1.358	0.912	1.200	0.309	0.343	1.937	1.937
	delta2FT_rel (%)	82.62%	70.94%	93.96%	58.55%	81.23%	88.60%	79.24%	100.00%	93.96%	5.09%
	delta2FT_abs (mm)	0.265	0.199	0.046	0.563	0.171	0.137	0.064	0.000	0.117	1.838

Table D.2: Conclusions from Table D.1.

		1A LACT983			1A LACT323			1A fullCT	Cast	MTU	Wing
	RanCEAF helped ICP?	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
	OC better than RanCEAF?	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
	ICP made OC worse?	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
		Mean/max RMSE									
ICP (subsets)	max	7.696	6.953	6.193	4.327	7.198	7.180	1.660	7.249	8.424	1.372
	mean	0.870	1.001	0.632	0.760	1.764	1.104	0.291	0.420	0.710	0.339
ICP (subset2full)	max	7.790	4.286	5.291	4.352	7.635	5.567	2.305	5.986	8.448	0.592
	mean	0.855	0.612	0.421	0.592	1.627	0.617	0.242	0.283	0.713	0.083
ICP (only full)	max	7.858	2.657	5.370	4.910	5.496	6.764	2.364	5.947	8.457	0.705
	mean	1.054	0.355	0.456	0.678	0.519	0.590	0.218	0.295	0.708	0.117
OC	max	7.723	3.692	2.814	3.788	5.534	6.089	2.425	6.246	7.967	0.397
	mean	0.930	0.631	0.471	0.733	0.568	0.759	0.194	0.490	1.023	0.073
ICP after OC	max	8.206	2.774	5.462	5.563	5.438	6.646	2.643	6.297	8.455	1.193
	mean	1.013	0.478	0.475	0.737	0.488	0.625	0.240	0.283	0.709	0.159
speed factor	OC vs. ICP (subsets)	1.7	4.2	2.8	3.3	2.9	1.7	2.2	1.9	1.4	1.9
	OC vs. ICP (meshes)	125.5	96.1	77.6	107.7	58.3	153.8	51.0	67.5	116.2	4.7
	OC vs. ICP (accuracy)	97.07%	57.72%	100.42%	108.37%	107.48%	110.28%	117.86%	67.43%	105.32%	117.00%
	OC%-bestICP	-2.87%	-39.38%	0.41%	6.72%	6.43%	8.74%	15.15%	-31.51%	5.05%	14.53%

Appendix E

Measurement Comparison

	Opt IWR (*)	Tactile IWR	Tact Empa	CT Empa (+)	CAD
E21-E24 (width)	(1) 39,917 (2) 39,917 (3) 39,867 (*) 39,915	39,969	39,9753	40,45	40,0
E10-E27 (outer depth top)	(4) 20,549 (5) 20,557	19,9806	19,9783		20,0
E2-E27 (outer depth bottom)	(6) 20,533 (7) 20,524	19,9701	19,9722		20,0
E8-E27 (inner depth plane)	(8) 12,420 (9) 12,413 (*) 12,415	11,9716	11,9807	12,18	12,0
E04-E27 (inner depth trench)	(10) 12,538	11.9864	11,9817		12,0
E1-E11 (heigth)	(11) 79,929 (12) 79,909 (*) 79,950	79,958	79,8213	81,07	80,0

(*) Measurements via OptoCat inspection of plane-point-distance of „ILATO_1B_SM2066-HE5-150_030214001_realigned.ply“
(+) Measurements via VG-Studio inspection of plane-point-distance of „ILATO_1B_FullAngleuCT_quickmesh20150722“

FIGURE E.1: Measurement comparison for applying all approaches to ILATO sample 1B.

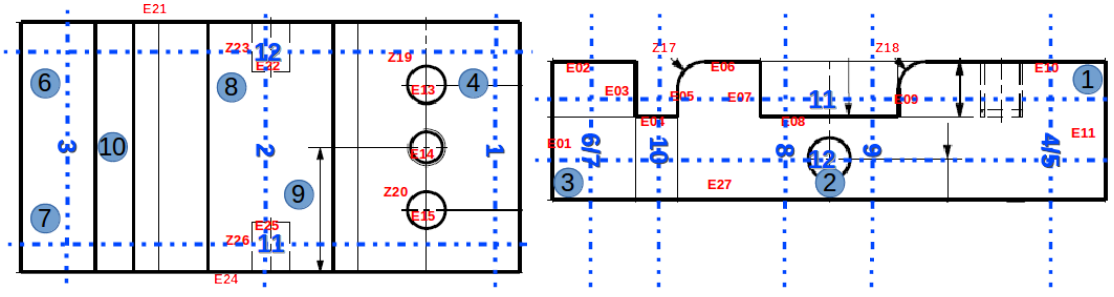


FIGURE E.2: Probing positions for manual tactile measurements via Caliper.

Appendix F

Discontinued Approaches

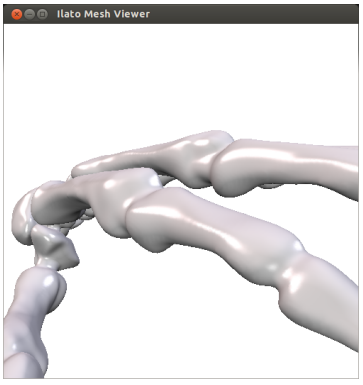
F.1 Smooth Mesh

Shifts each vertex in the barycenter of its one ring neighborhood. is applied iteratively until mesh is sufficiently smooth. Also implements parametrization to sphere [GGS03] by normalizing each position vector before each iteration. In case of parametrization to sphere, iterations continue until all face normals point *outside*. Since this approach requires all position vectors to have length one, a simple check is to sum position vector and normal vector for each face and check if resulting length is greater or equal one.

F.1.1 Assessment of Randomness with a 2D Manifold

Parametrization in unit sphere as workaround. Random of seeds around object is shown in Figure F.2A. Randomness of selection assessed as mapping of selected subset from

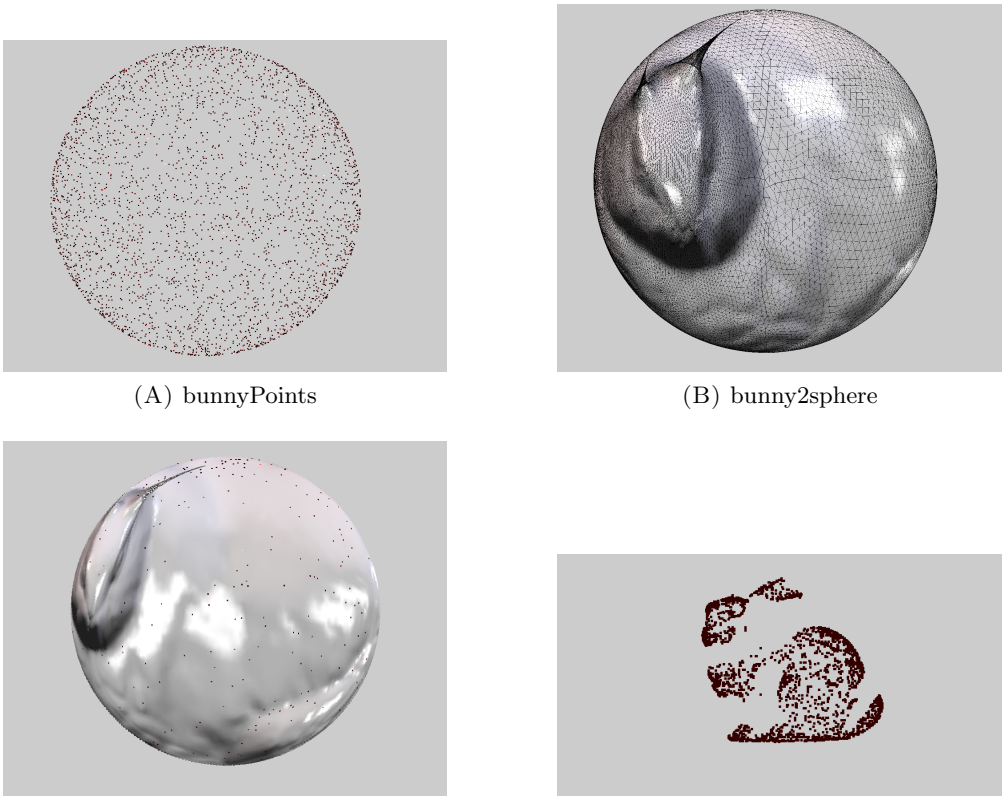
(A) handBeforSmooth



(B) handAfterSmooth

FIGURE F.1: Effect of smoothing algorithm.

specimen as in Figure F.2D to corresponding unit sphere as in Figure F.2A.



(A) bunnyPoints (B) bunny2sphere

(C) bunnyPlusPoints (D) points

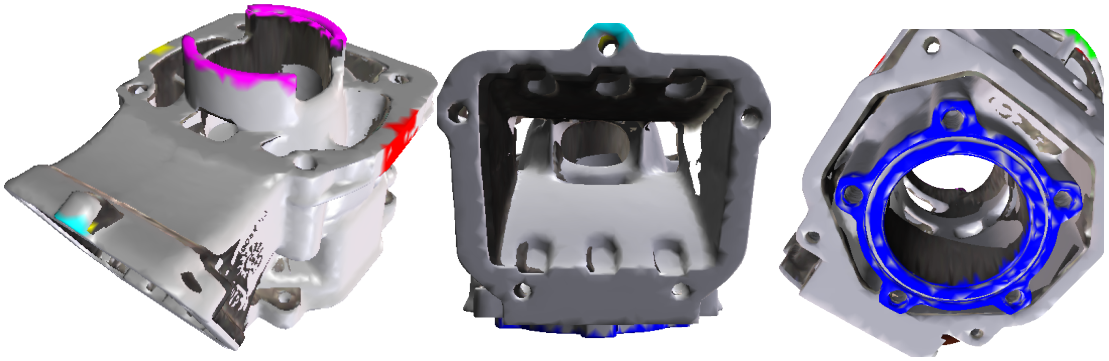
FIGURE F.2: Bunny parametrization to sphere.

For a closed surface such as the bunny, the parametrization to sphere is achieved by alternatingly setting all position vectors to unit length and apply a smoothing iteration.

F.2 Pre-Alignment based on global features

(pre)Alignment:

- compute convex hull for volumetric model and surface model
- align meshes along ellipsoid axis and compute MVBB
- align meshes along MVBB axis
- scale models and adjust center
- extract thin layer of mesh with *touches* MVBB
- compare cut layer volumes to flip axis if necessary



(A) prealigned meshes (B) screw thread in front (C) screw thread in bottom

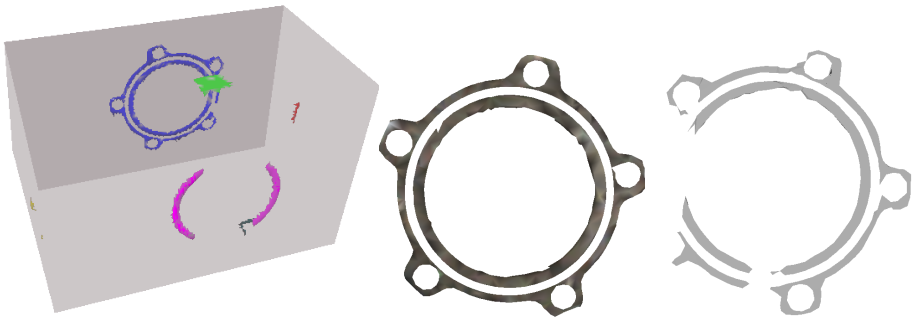
FIGURE F.3: Meshes $\mathcal{M}_{\text{CT}}$ and $\mathcal{M}_{\text{CT}}$ prealigned based on surface parts intersecting with MBB.

F.3 Alignment based on MVBB and outer Surface Layers

The following test is based in the prealigned data as shown in Section F.3. To improve alignment accuracy the surface parts close to the MVBB are extracted for each mesh. Those intersections are referred to as *cut layer* in the following since they only comprise a flat layer or triangles.

F.3.1 AluCylinderCutLayers

Intersection with shrinked MVBB, processed per MVBB side



(A) minBbCutFaces (B) 3D full00 (C) CT partial00

FIGURE F.4: MBB intersections from cylinder cast for alignment test

F.4 RANSAC versus ICP

Comparison of established alignment algorithms challenged with identical problem of aligning highly rotation symmetric shape

F.4.1 Alignment Comparison

Two almost symmetric patterns were extracted from CT mesh and optical mesh. The task was to align them properly. RANSAC produced solutions comparably fast although not all of them were as good as the one shown below (∼33% success rate). ICP had a ∼5% success rate in solving the alignment task by considerably higher computation time.

RANSAC shows good convergence and results. Mesh resolution is a critical factor since the presented metric ($relVec$) only works if similar - almost identical - point pairs are present. Furthermore, high resolution meshes require more iterations which increases hash table size. In theory it is possible for them to grow almost indefinitely.

- largest cut layer of mesh and minimal bounding box as input
- as cuts are not perfect similar structures slightly differ
- in this test ICP is easily caught in local minima

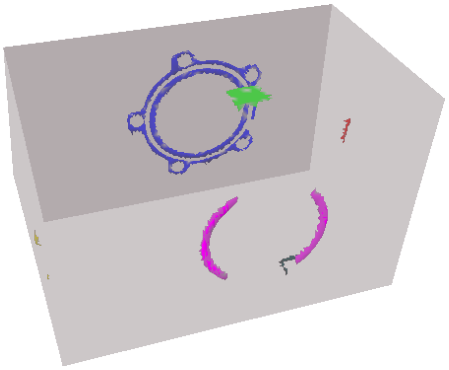
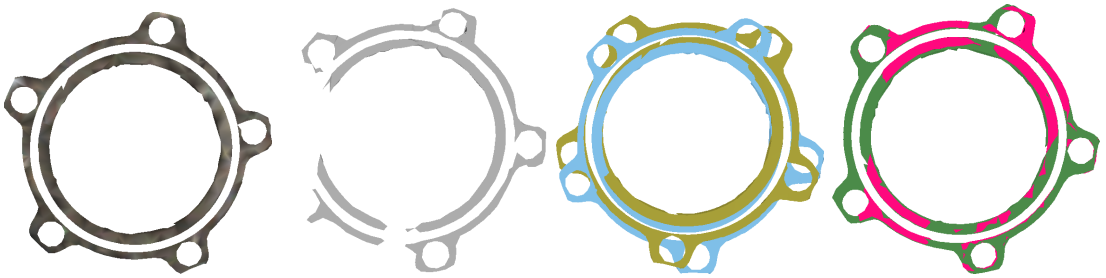


FIGURE F.5: bounding box cuts



(A) input 3D (B) input CT (C) result ICP (D) result RANSAC

F.5 NURBS Surface Patches for Hypothesis Generation

F.5.1 Random Surface Point Selection

RANSAC scheme relies on randomly chosen points but not all choices are equally good. To improve the hypothesis generation rate ($relVec$ hits in hash tables) we need to restrict the points to choose from to a meaningful subset of all vertices in the mesh.

- Mesh from CT covers more surface than mesh from 3D scan
- Matches only occur if counter pair is present in other mesh
- Thus, only exterior vertices are relevant to RANSAC

FIGURE F.6: interior surface

F.5.2 NURBS & Region Growth

To overcome dependency of similar resolution, we describe small portions of the mesh in parametric form. Instead of selecting a pair of points and computing their characteristic vector, we select one point and apply region growth to select a small patch. A description of this path as non rational uniform b-spline should provide basis for a similarity search in analogy to comparison of characteristic vectors. Unable to reduce the knot vector count related resolution of the underlying patch, this is still not resolution independent.

We want to replace RANSACs 4D relation vector $relVec$ as metric for point pair similarity. A resolution independent mathematical definition of a small subset of the mesh surface is desired. The intention is to have a surface patch as introduced in "stencil selection" as control points for a Non-uniform rational B-spline (NURBS) surface.

- stencil patches replace the concept of point pairs

- NURBS replace the concept of $relVec$
- similarity of two NURBS then indicates similarity of associated patches and therefore a hypothesis
- verification remains as distance measure after transformation of random points
- ... and unfortunately this seems not to work.

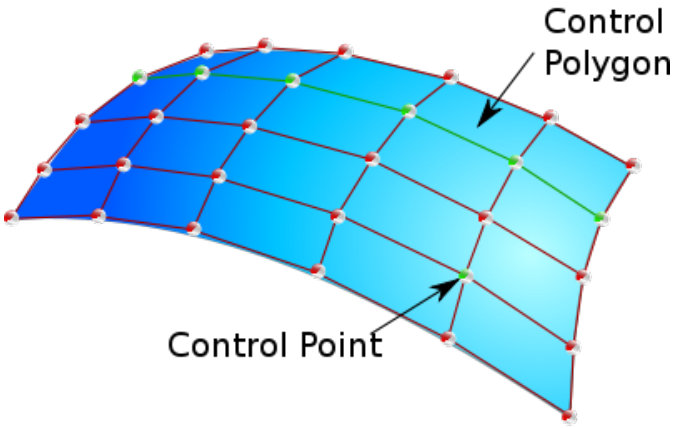


FIGURE F.7: NURBS surface

List of Figures

List of Tables

Bibliography

[AM05] T. Akenine-Möller. Fast 3D Triangle-box Overlap Testing. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA, 2005. ACM Press.

[Bad90] D. Badouel. Graphics Gems. chapter An Efficient Ray-polygon Intersection, pages 390–393. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[Bar81] A. H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, Jan 1981.

[BBS10] C. Bathow, B. Breuckmann, and R. Scopigno. Verification and Acceptance Tests for High Definition 3D Surface Scanners. In *Proceedings of the International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*, Palais du Louvre, Paris, France, 2010. Eurographics Association.

[BF12] F. E. Boas and D. Fleischmann. CT artifacts: Causes and reduction techniques. *Imaging in Medicine*, 4(2):229–240, 2012.

[BLK16] A. Beyer, Y. Liu, and S. Krömker. OctaCoG Alignment: A Fast and Reliable Alternative to Iterative Alignment Schemes. In *19^{th} International Conference on Information Fusion (FUSION)*, Heidelberg, Germany, 2016.

[BLMK16] A. Beyer, Y. Liu, H. Mara, and S. Krömker. Mesh Reduction to Exterior Surface Parts via Random Convex-Edge Affine Features. In *Mathematical Aspects of Computer and Information Sciences (MACIS 2015)*, volume 9582 of *LNCS*, pages 63–77, Berlin, Germany, 2016.

[BM92] P. J. Besl and N. D. McKay. A Method for Registration of 3D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992.

[BMK14] A. Beyer, H. Mara, and S. Krömker. ILATO Project: Fusion of Optical Surface Models and Volumetric CT Data. *The Computing Research Repository (CoRR)*, abs/1404.6583, 2014.

[BPC+] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1).

[Bru02] P. P. Bruyant. Analytic and Iterative Reconstruction Algorithms in SPECT. *Journal of Nuclear Medicine*, 43(10):1343–1358, 2002.

[BSBK02] M. Botsch, S. Steinberg, S. Bischoff, and L. Kobbelt. Openmesh: A generic and efficient polygon mesh data structure. OpenSG Symposium, 2002.

[CC78] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350 – 355, 1978.

[CCC+08] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In *Sixth Eurographics Italian Chapter Conference*, pages 129–136, 2008.

[Coo57] J. Cook. Rational formulae for the production of a spherically symmetric probability distribution. *Mathematics of Computation*, 11(58):81–82, 1957.

[Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[EM94] H. Edelsbrunner and E. P. Mücke. Three-Dimensional Alpha Shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

[FDK84] L. A. Feldkamp, L. C. Davis, and J. W. Kress. Practical Cone-Beam Algorithm. *Journal of the Optical Society of America A*, 1(6):612–619, 1984.

[GGS03] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3d meshes. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 358–363. ACM, 2003.

[GJ+10] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[HH10] J. Havel and A. Herout. Yet faster ray-triangle intersection (using SSE4). *IEEE Transactions on Visualization and Computer Graphics*, 16(3):434–438, 2010.

[HP01] S. Har-Peled. A Practical Approach for Computing the Diameter of a Point Set. In *Proceedings of the 17th Annual Symposium on Computational Geometry*, SCG '01, pages 177–186, New York, NY, USA, 2001.

[Hut02] B. F. Hutton. An introduction to iterative reconstruction. *Alasbimn Journal*, 5(18):Article NAJ18–6, 2002.

[Joe91] B. Joe. Construction of Three-Dimensionalimensional Delaunay Triangulations using Local Transformations. *Computer Aided Geometric Design*, 8(2), 1991.

[KGA+10] A. Klein, S. S. Ghosh, B. Avants, B. T. Yeo, B. Fischl, B. Ardekani, J. C. Gee, J. J. Mann, and R. V. Parsey. Evaluation of volume-based and surface-based brain image registration methods. *NeuroImage*, 51(1):214–220, 2010.

[KK86] T. L. Kay and J. T. Kajiya. Ray tracing complex scenes. In *ACM SIGGRAPH computer graphics*, volume 20, pages 269–278. ACM Press, 1986.

[KL80] V. Klema and A. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control*, 25(2):164–176, 1980.

[Kle16] T. Klemm. Lokale Sensitivitätsanalyse von Isoflächen mittels der Berechnung von Voxelpfaden in Volumendaten. Bachelor thesis, Heidelberg University, Heidelberg, Germany, July 2016.

[Koc95] R. Koch. 3-D surface reconstruction from stereoscopic image sequences. In *Proceedings of the 5th International Conference on Computer Vision*, pages 109–114. IEEE, 1995.

[KS06] A. Kensler and P. Shirley. Optimizing ray-triangle intersection via automated search. In *IEEE Symposium on Interactive Ray Tracing 2006*, pages 33–38. IEEE, 2006.

[LBS+16] Y. Liu, A. Beyer, P. Schuetz, J. Hofmann, A. Flisch, and U. Sennhauser. Cooperative data fusion of transmission and surface scan for improving limited-angle computed tomography reconstruction. *NDT & E International*, 83:24–31, 2016.

[LC87] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH Computer Graphics*, 21(4):163–169, 1987.

[Lie03] P. Liepa. Filling Holes in Meshes. In *Proceedings of Eurographics Symposium on Geometry Processing (SGP) 2003*, pages 200–207. Eurographics Association, 2003.

[Mar72] G. Marsaglia. Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics*, 43(2):645–646, 04 1972.

[Mar12] H. Mara. *Multi-Scale Integral Invariants for Robust Character Extraction from Irregular Polygon Mesh Data*. PhD thesis, Interdisciplinary Center for Scientific Computing, Ruprecht-Karls-Universität Heidelberg, Heidelberg, 2012.

[ML07] P. Montes and G. Lauritsch. A temporal interpolation approach for dynamic reconstruction in perfusion CT. *Medical physics*, 34(7):3077–3092, 2007.

[MN98] M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, 1998.

[Mon09] P. Monigatti. Reconstructing internal building structure from 802.11 attenuation measurements. 2009.

[Mos05] N. Moshtagh. Minimum volume enclosing ellipsoid. *http://www.mathworks.com/matlabcentral/fileexchange/9542-minimum-volume-enclosing-ellipsoid*, 2005.

[MT05] T. Möller and B. Trumbore. Fast, minimum storage ray/triangle intersection. In *ACM SIGGRAPH 2005 Courses*, page 7. ACM Press, 2005.

[Mul59] M. E. Muller. A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM*, 2(4):19–20, 1959.

[NDF14] M. Nießner, A. Dai, and M. Fisher. Combining Inertial Navigation and ICP for Real-time 3D Surface Reconstruction. *Eurographics 2014-Short Papers*, pages 13–16, 2014.

[PEK+01] G. P. Penney, P. J. Edwards, A. P. King, J. M. Blackall, P. G. Batchelor, and D. J. Hawkes. A Stochastic Iterative Closest Point Algorithm (stochastICP). In *Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 762–769. Springer-Verlag, 2001.

[Rad17] J. Radon. Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. *Akad. Wiss.*, 69:262–277, 1917.

[Ram91] A. G. Ramm. Inversion of Limited-Angle Tomographic Data. *Computers & Mathematics with Applications*, 22(4–5):101–111, 1991.

[RL15] B. Rieck and H. Leitte. Comparing dimensionality reduction methods using data descriptor landscapes. In *Symposium on Visualization in Data Science (VDS) at IEEE VIS 2015*, Chicago, IL, USA, 2015.

[Sch12] B. Schlei. Volume-Enclosing Surface Extraction. *Computers & Graphics*, 36(2):111–130, 2012.

[SM92] R. Sablatnig and C. Menard. Stereo and Structured Light as Acquisition Methods in the Field of Archaeology. In *Mustererkennung '92, 14. DAGM-Symposium Dresden*, pages 398–404. Springer, 1992.

[SSKN07] M. Shevtsov, A. Soupikov, A. Kapustin, and N. Novorod. Ray-triangle intersection algorithm for modern CPU architectures. In *Proceedings of GraphiCon*, volume 2007, pages 33–39, 2007.

[TH99] S. Teller and M. Hohmeyer. Determining the lines through four lines. *Journal of Graphics Tools*, 4(3):11–22, 1999.

[Tuy81] H. Tuy. Reconstruction of a Three-Dimensional Object from a Limited Range of Views. *Journal of Mathematical Analysis and Applications*, 80(2):598–616, 1981.

[TY07] M. J. Todd and E. A. Yıldırım. On Khachiyan's Algorithm for the Computation of Minimum-Volume Enclosing Ellipsoids. *Discrete Applied Mathematics*, 155(13):1731–1744, 2007.

[WMW06] S. Winkelbach, S. Molkenstruck, and F. M. Wahl. Low-cost laser range scanner and fast surface registration approach. In *Pattern Recognition*, pages 718–728. Springer, 2006.

[WPS+03] I. Wald, T. J. Purcell, J. Schmittler, C. Benthin, and P. Slusallek. Realtime ray tracing and its use for interactive global illumination. *Eurographics State of the Art Reports*, 1(3):5, 2003.

[YAL+02] T. S. Yoo, M. J. Ackerman, W. E. Lorensen, W. Schroeder, V. Chalana, S. Aylward, D. Metaxas, and R. Whitaker. Engineering and algorithm design for an image processing API: A technical report on ITK-the Insight Toolkit. *Studies in Health Technology and Informatics*, pages 586–592, 2002.

[ZHW05] J.-W. Zhang, G.-Q. Han, and Y. Wo. Image registration based on generalized and mean Hausdorff distances. In *Proceedings of the International Conference on Machine Learning and Cybernetics 2005*, volume 8, pages 5117–5121. IEEE, 2005.