Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

DLR Deutsches Zentrum
für Luft- und Raumfahrt
German Aerospace Center

Fachbereich Elektrotechnik, Maschinenbau
und Technikjournalismus (EMT)
Studiengang Maschinenbau (B. Eng.)

Bachelor-Thesis

Geometry matching of CAD-, CT- and melt temperature data for selective laser melting

Submitted by:
Philipp Rosauer
An der Tränke 13
53773 Hennef
Tel. 01578 8519837
Philipp.Rosauer@smail.emt.h-brs.de
Matr.-Nr. 9029367

First assessor: Dr. Jan Kleinert
Second assessor: Prof. Dr. Dirk Reith

Hennef, February 28, 2019

Contents

List of Figures III

List of Tables V

List of Abbreviations VII

List of Symbols IX

1 Introduction 1
- 1.1 Motivation 1
- 1.2 Aim 1
- 1.3 Outline 2

2 Theory 3
- 2.1 Selective laser melting 3
- 2.2 Data formats 4
  - 2.2.1 Product model data 4
  - 2.2.2 Melt temperature data 5
  - 2.2.3 Computer tomography data 6
- 2.3 Geometric transformations and their representation 7
  - 2.3.1 Translation 8
  - 2.3.2 Rotation 8
  - 2.3.3 Scale 10
- 2.4 Numerical optimization 10

3 Methods 11
- 3.1 Transformation based on physical properties 11
  - 3.1.1 Translation 11
  - 3.1.2 Rotation 12
  - 3.1.3 Scale 14
- 3.2 Finding a transformation with minimization 15
- 3.3 Matching faces 15
  - 3.3.1 Face selection 15
  - 3.3.2 Face assignment 16
- 3.4 Matching vertices 17
  - 3.4.1 Vertex selection 17
  - 3.4.2 Vertex assignment 17

4 Results **19**
4.1 Geometry matching of test data 19
4.1.1 Analysis of the approach based on physical properties 19
4.1.2 Analysis of the numerical optimization 21
4.2 Application for measured data 23
4.2.1 Using the physical properties 23
4.2.2 Optimization of the transformation 25
4.2.3 Measuring the error of the methods 27

5 Conclusion **29**

Bibliography **31**

List of Figures

2.1-1 Schematic design of the SLM process [18] 3
2.2.1-1 Local control of a B-spline surface with a grid of control points [13] . . . 5
2.2.1-2 CAD models of the analyzed data 5
2.2.2-1 Point cloud representations of the analyzed data 6
2.2.3-1 Two visualizations of the CT data 7

3.3.2-1 Matching of two edges 16

4.1.1-1 Geometry matching of a T-shape by computation of its physical properties 20
4.1.1-2 Mean error depending on initial orientation 20
4.1.1-3 Mean error of the transformation compared to the mean error of the center of mass depending on the resolution 21
4.1.2-1 Matching three faces of the T-shape 22
4.1.2-2 Mean error after face matching without locked transformations depending on the initial orientation 22
4.1.2-3 Mean error after face matching with locked transformations depending on the initial orientation 22
4.1.2-4 Mean error after vertex matching depending on initial orientations 23
4.2.1-1 Geometry matching of an impeller by computation of its physical properties 24
4.2.1-2 Close ups of matching of CAD and MPM data based its on physical properties 24
4.2.1-3 Results of the matching of CT and CAD data 25
4.2.2-1 Optimization using a face 26
4.2.2-2 Matching of CT and CAD data to locate deformation with the use of the face method 26
4.2.2-3 Optimization using vertices 27

List of Tables

4.1.1-1 Absolute and relative error of the z-value of the center of mass depending on the resolution of the T-shaped point cloud 20
4.2.3-1 Comparison of the quality of the used matching methods 27

List of Abbreviations

BFGS	Broyden-Fletcher-Goldfarb-Shanno algorithm
CAD	Computer aided design
CT	Computer tomography
DFP	Davidon-Fletcher-Powell algorithm
ISO	International Organization for Standardization
MPM	Melt pool monitoring
NURBS	Non-uniform rational B-Spline
SDK	Software development kit
SLM	Selective laser melting
SLS	Selective laser sintering
STEP	Standard for the Exchange of Product model data
TIFF	Tagged image file format
VTK	Visualization Toolkit

List of Symbols

N	Total amount of points
ϕ	Angle of rotation about the x-axis
θ	Angle of rotation about the y-axis
ψ	Angle of rotation about the z-axis
$\vec{L}$	Angular momentum
$\vec{\omega}$	Angular velocity
$\vec{R}$	Center of mass
ρ	Density
$\vec{x}$	Vector storing the elements of the transformation matrix
∇	Gradient
I	Inertia tensor
$\mathit{\Delta}$	Laplace operator
m	Mass
ϵ	Mean error
$\vec{p}$	Momentum
$f(\vec{x})$	Objective function
$\vec{r}$	Point in a point cloud
$\vec{a}$	First principal axis of inertia
$\vec{b}$	Second principal axis of inertia
$\vec{c}$	Third principal axis of inertia
Q	Solid shape
$\vec{q}$	Quaternion
$\vec{i}, \vec{j}, \vec{k}$	Fundamental quaternion units
T_r	Rotation matrix
R_x	Rotation matrix about the x-axis
R_y	Rotation matrix about the y-axis
R_z	Rotation matrix about the z-axis
λ	Scalar variable
$\vec{s}$	Scaling vector
T_s	Scaling matrix
s	Scaling scalar
$\mathbb{R}$	Set of real numbers

M	Total mass of a data set
T	Transformation matrix
T_t	Translational matrix
$\vec{t}$	Translational vector
E	Unit matrix
$\vec{v}$	Velocity
$\vec{d}$	Vertex in the CAD data
x, y, z	Cartesian coordinates
$\vec{u}, \vec{v}, \vec{w}$	Unit vectors

1 Introduction

Selective laser melting (SLM) is an additive manufacturing process that allows the production of complex geometries without the use of welding. As an additive manufacturing process, it is currently used especially in prototype construction and small series. Data can be collected during the production, to better understand and optimize the physical processes. This thesis deals with the visual processing of these data in order to detect anomalies efficiently. For this purpose, the data collected in various manufacturing steps are matched geometrically.

1.1 Motivation

Rocket component engineering requires not only the highest safety and reliability standards, but also the reduction of weight, while increasing strength at the same time. Additive manufacturing processes and especially SLM, with its ability to process high-strength materials, are particularly well suited for the development of turbo-engine applications such as rocket propulsion systems.[4]

One of the core issues in fabrication of engine components using SLM is deformation during melting, due to temperature gradients. The melting of materials such as titanium alloys requires very high temperatures. Using a laser, the necessary thermal energy can be applied locally. Depending on various parameters, like the route of the laser, or the geometry of the work piece, cooling takes place at varying speeds. These result in varying temperature distributions, causing stress and material failure.[7] Already equipped with a laser, the live-reading of the local melt temperature is quite easy and a serial feature of a SLM machine. The collected data can be visualized as a point cloud, every point representing one measurement. Previous research showed that cracks seen in CT scans can also be detected in the melt temperature data as areas of very high thermal energy.

To further analyze and compare various sets of data, it is of great interest to implement some sort of geometry matching. This will allow precise error calculations and visualization of critical areas. This analysis could help to better understand the fabrication process and it provides a tool to evaluate the impact of the melt temperature distribution on its precision. The gathered data could then be used to optimize laser routes, speeds and work piece geometry. This enables the efficient and resource-saving development of new products and opens new markets to additive manufacturing technologies.[7]

1.2 Aim

The aim of this thesis is to find a method that matches data of various origins, computer aided design (CAD), melt pool monitoring (MPM) and computer tomography (CT), to visualize them in reference to each other. To achieve that, first the center of mass and the principal axes of inertia are computed and a transformation is calculated to match the data sets. Both the CT and MPM data are matched to the CAD data, enabling a direct comparison. As both data sets are matched to the static CAD data, an indirect comparison between the two can be achieved. The quality of the matching depends heavily on the resolution and initial orientation of the gathered data. Therefore, a more stable and precise method is developed, selecting faces and vertices in the data sets and minimizing their distance. As the geometry represented by the MPM data is equal to the CAD geometry, a precise matching is expected, to allow the localization of geometries causing temperature gradients. For that, the selection of vertices is chosen. Contrary, CT data represents the actual, deformed geometry of the manufactured work piece. Here, a measurement of the geometry offset relative to one or more aligned faces is more appropriate. For both methods a precision within the range of the layer thickness of the SLM process at $20\ \mu m - 75\ \mu m$ is aspired.

1.3 Outline

First, the origins and properties of the various data sets are presented. Mathematical basics such as the geometrical transformations, including rotation, translation and scale are introduces. Moreover, a short introduction to numerical optimization is given.

Based on this, the methods used in this work are presented: An approach solely based on the physical properties of the data, another using faces of the geometry and one using vertices to minimize the displacement. The key challenges of each approach are stated and various methods to overcome them are discussed.

The proposed methods are first tested with generated example data in form of a T-shape to verify them. Then the results achieved with real data, gathered during the production of an impeller are presented and discussed.

The work concludes with a proposal on the use of the developed methods and their extension in the development of components manufactured with SLM.

2 Theory

In order to implement the proposed methods, first the manufacturing process of SLM is presented. Further, the origin and format of each data set is introduced. In addition, the necessary mathematical context for the composition of a transformation is laid out.

2.1 Selective laser melting

The tool to be implemented primarily serves the purpose of matching data gathered during the SLM process. SLM is a manufacturing technique that has developed from selective laser sintering (SLS). Like SLS, it is an additive manufacturing process that uses bundled radiation to heat a granulate and bond it locally. In the SLM process, higher laser intensities are used, in order to reach temperatures high enough to melt the powder. Melting results in higher densities than sintering, which leads to higher component strengths. This makes this process interesting for new applications, for example in space- or automotive technology.
As shown in Figure 2.1-1, the laser beam is generated by a stationary source. By redirecting the beam using a pivoting mirror a two dimensional geometry can be molten. Lowering the build platform and filling the build chamber with a new layer of powder, the process is set to be repeated. The work piece is produced layer by layer, each thin enough to allow for a strong link in between.

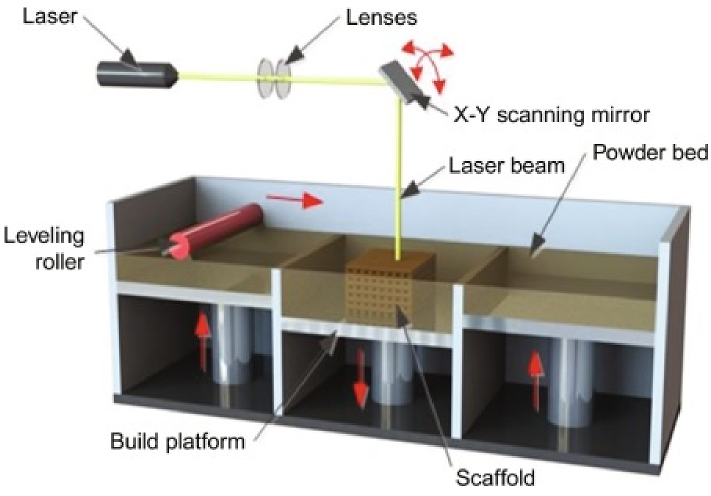


Figure 2.1-1: Schematic design of the SLM process [18]

To enable high production speeds, high temperatures and fast cooling is necessary. The fast heating of one layer and the slower heat conduction into previously molten layers result in steep temperature gradients which cause the development of residual stresses. These can lead to deformations or even cause part failure. Kruth et al. discuss the impact of various parameters in the manufacturing process, such as scanning strategy, laser power, layer thickness and energy distribution. Scanning strategy means mainly the route of the laser. The amount of times an area is heated and cooled again and the time in between are all parameters emerging from the strategy and affecting the outcome. The authors conclude that especially the scanning strategy can have significant impact on the measured stresses in a manufactured work piece.[7]

As the use of SLM increases in research and development, a better knowledge of the impact of each process parameter is of great value. This can be acquired by analyzing the quality and correlations between anomalies occurring in the process.

2.2 Data formats

Data is gathered in three different stages of development. In the first stage, the target geometry is determined by creating a draft using CAD. In the second stage, the work piece is manufactured and current laser intensities are measured and stored. The last stage is production metrology. Here, CT is used to measure the actual manufactured geometry.

2.2.1 Product model data

CAD data represents the work piece in target condition. The comparison of measured data, CT and MPM, to CAD data allows the visualization of deviations.

To enable cross platform development and exchange, an ISO standardized file format was developed between 1984 and 2002. ISO 1030, commonly known as STEP (standard for the exchange of product model data), regulates the storage of any information related to the design of a part, or assembly. Although this powerful format is capable of storing all kinds of information about materials to be used, kinematics or computational fluid dynamics data, only the general geometric and topological representation is used here.[11] Such a representation is constructed using non-uniform rational B-Splines (NURBS). This mathematical description of curves and surfaces originated from the difficulties in the design of smooth surfaces of ships and later cars. B-Spline curves and surfaces are parametric representations of an object. They allow an axis-independent representation of multiple-valued functions. A spline is a curve of one or more segments, described by polynomial functions. It can be described using control points and, if multi segmented, knot points. One specific type of NURBS is a Bézier curve. The polynomial defining such a curve, named after its developer Pierre Bézier, always has one degree less than it has control points. Therefore, such a curve can only be single segmented. The control points of such a single segmented curve allow global control of the whole curve. This is of use in the design of car bodies, aircraft fuselages or ship hulls.

On the other hand, the design possibilities are limited by the global influence of each control point. Another disadvantage of Bézier curves is the strict connection between the degree of the polynomial and the amount of control points. B-Splines solve both problems. As the degree of the polynomial of such a curve can be smaller, the curve can be segmented by knot points. This results in a more localized impact of each control point, shown in Figure 2.2.1-1

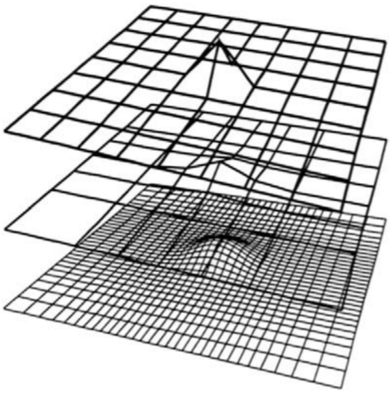
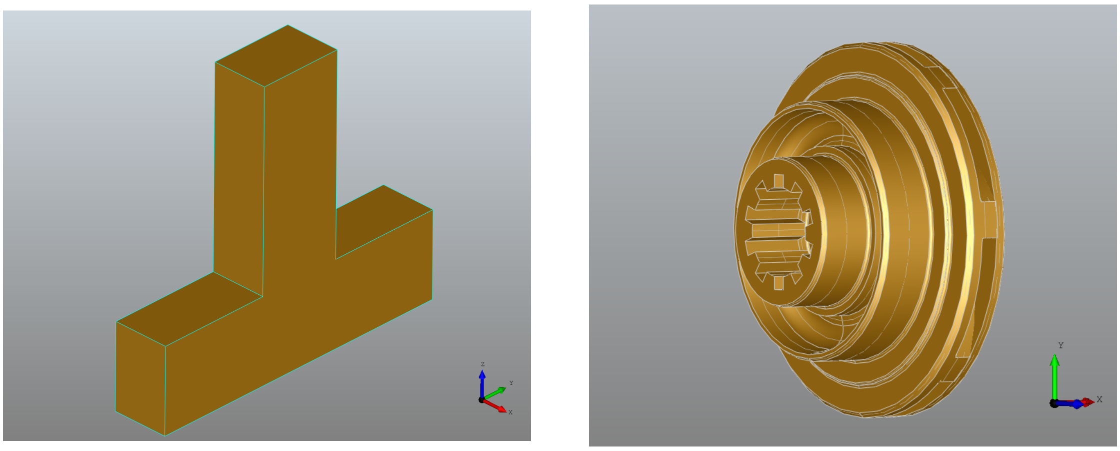


Figure 2.2.1-1: Local control of a B-spline surface with a grid of control points [13]

for a B-Spline surface.

Modeling a three dimensional object requires the use of surfaces, in addition to curves. Similarly to the description of Bézier- and B-Spline curves, surfaces can be characterized and manipulated by control points. The same advantages and disadvantages of one or the other apply here.[13]

The representations of a T-shape and an impeller as CAD models are shown in Figure 2.2.1-2. The processing of NURBS is well implemented, because many CAD programs represent geometries in this way. The stored information about surfaces, edges and vertices can be used directly with OpenCASCADE[14], a software development kit (SDK) providing the necessary tools to handle and manipulate not only CAD data. The SDK is used to implement the majority of the needed methods, as well as the visualization of the outputs in Chapter 4.



(a) T-shape

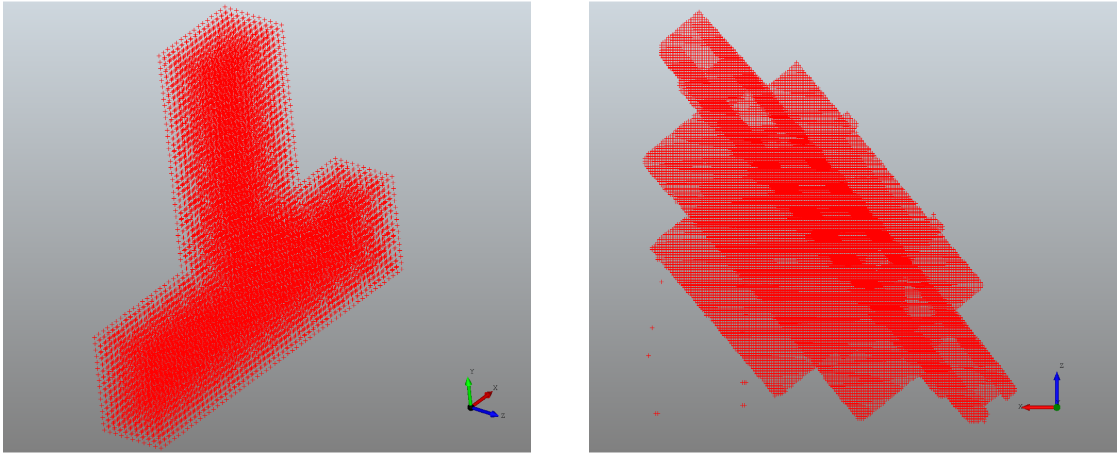
(b) Impeller

Figure 2.2.1-2: CAD models of the analyzed data

2.2.2 Melt temperature data

Using a laser beam to melt the metallic powder during SLM provides the possibility to gather live laser intensity data. As the measured intensity yields a good approximation to the prevailing melt temperature, the gathered readings are valuable in the analysis of the

process.[10]
Since the pivoting mirror is controlled in a way that the laser follows the exact route computed based on CAD data, each reading is located at a defined point in a structured grid. The grid spacing in the two directions of the build platform, x and y, is uniform and determined by the measuring frequency of the laser. The spacing in the third direction, z, is set by the layer thickness and therefore unequal to the x- and y-direction.
To visualize the data gathered during the process, the exact position in the grid and the corresponding intensities need to be stored. This is accomplished by using the Visualization Toolkit (VTK) file format. Only the intensities and general parameters of the grid are stored directly. By holding information about the bounds and the origin of the grid, as well as the relative spacing in all three directions, each stored scalar can be allocated to one unique location in that grid. Thus, the geometry of the measured piece is stored indirectly.[15] The efficiency of this data storage is useful especially as the amount of gathered measuring points can easily exceed a few million.
Figure 2.2.2-1b shows the measured data of the impeller represented as a point cloud. The scalar values are not represented, as this is of no use for the matching process itself. MPM data is colored in red throughout this thesis, CAD is colored in gold and CT data is colored in purple. As the T-shape is only used to verify the implementations, it is generated and its dimensions match those of the CAD T-shape in Figure 2.2.1-2a.



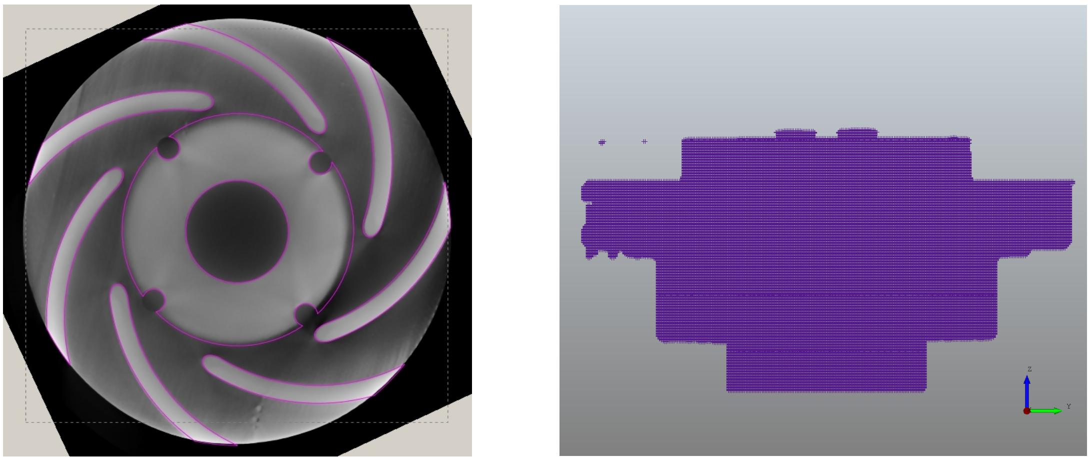
(a) Generated T-shape

(b) Measured MPM data of the impeller

Figure 2.2.2-1: Point cloud representations of the analyzed data

2.2.3 Computer tomography data

CT enables the visualization of the inside of an object without destroying it. X-rays are taken from different angles to generate a two dimensional image of the measured slice. Repeating this process results in a collection of images, one for each scanned slice.[5] This collection is stored in the tagged image file format (TIFF) as a stack. Each image in the stack holding data in two directions, x and y. Unlike MPM data, the stored information is only binary. Figure 2.2.3 shows a CT image, each white pixel indicating the presence and each black the absence of material. As each image provides the same amount of pixels, again a structured grid is



(a) One image of the TIFF-stack **(b)** Measured CT data represented as a point cloud

Figure 2.2.3-1: Two visualizations of the CT data

given. Its spacing in x- and y-direction depends on the resolution of each image, whereas the spacing in z-direction can be computed from the layer thickness. This information can be stored in a VTK file which can be visualized as a point cloud, as in Figure 2.2.3-1b. As the CT is done after cooling, it holds all information about possible deformations and cracks of the actual work piece.

2.3 Geometric transformations and their representation

To geometrically match the previously presented data sets, they need to be transformed into one shared coordinate system. To accomplish that, the possible transformations are introduced, as well as a strategy to represent them in one function.

There are different types of geometric transformations. A rigid transformation preserves the shape and size of a geometry, only rotations and translations are allowed. In a three dimensional space and one degree of freedom per transformation and dimensions these sum up to six. Of course, the number of degrees can be decreased, by locking certain rotations or translations. A similarity transformation preserves the shape, but not the size. This is achieved by extending the transformation by a uniform scale. If the scale is not uniform, the shape is not preserved. Shearing is allowed and the transformation, now with nine degrees of freedom, is called affine. Allowing even projective transformations results in homography, only preserving collinearity, which is the property of a set of points to lie on a straight line.[20]

Since the shape is to be preserved, to allow the analysis of deformations, only a uniform scale is used. All transformations computed are therefore similarity transformations.

2.3.1 Translation

A translation relocates a point $\vec{r}$ by a given distance in a given direction. The most simple way to describe such a transformation is as an addition of a vector $\vec{t}$, see equation 2.3.1-1.

$$\begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \begin{pmatrix} r_x + t_x \\ r_y + t_y \\ r_z + t_z \end{pmatrix} \tag{2.3.1-1}$$

Each element of this vector describes a distance in each direction the point is moved. This means that a translation is independent of the current position of the point. It is a transformation with no fixed points.

2.3.2 Rotation

Unlike a translation, a rotation is a transformation with at least one fixed point. A rotation of an object or point can either be accomplished by rotating it in a fixed coordinate system, or by rotating the coordinate system and fixing the object or point.

The most intuitive way of describing a rotation is by defining a rotation axis and rotating about that axis by a certain angle. This so called ”Euler axis and angle” rotation is based on Euler’s rotation theorem, stating that any rotation can be expressed as a single rotation about an axis. Mathematical operations using this representation are not as intuitive and combinations with translations or scaling are tedious.

A more common rotation formalism in practical applications is the ”Euler rotations” formalism. It splits the rotation into three basic rotations around the axes of the coordinate system. Each rotation can be represented by a matrix, as in Equation 2.3.2-1.

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix} \tag{2.3.2-1a}$$

$$R_y(\phi) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.3.2-1b}$$

$$R_z(\psi) = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.3.2-1c}$$

There are different conventions being used. In robotics most often z-x-z Euler angles are used, indicating that the first rotation is about the z-, the second about the x- and the third again about the z-axis. The axes are these of the fixed coordinate system. These kind of rotations are called extrinsic. Another convention, often used in aviation, is the intrinsic z-y’-x” convention. Contrary to the z-x-z convention, the axes are not these of the fixed coordinate system, but of the coordinate system rotating with the object. The individual rotations are often referred to as yaw, pitch and roll .

If the individual rotations in Equation 2.3.2-1 are performed successively, they can be multiplied in the chosen order. A common problem with this notation is that of gimbal lock. Multiplication of the equations in 2.3.2-1 in the order x-y-z in the case of a rotation by 90 degrees about the y-axis results in the rotation matrix in Equation 2.3.2-2.[17]

$$T_r(\phi, \psi) = \begin{pmatrix} 0 & 0 & 1 \\ \sin(\phi + \psi) & \cos(\phi + \psi) & 0 \\ -\cos(\phi + \psi) & \sin(\phi + \psi) & 0 \end{pmatrix} \tag{2.3.2-2}$$

It is clear that altering ϕ or ψ both will result in a rotation about the same axis. A degree of freedom is lost. Other than that, this representation is very useful, as it allows intuitive manipulation. Thus, it is used in the implementation, whenever a manual adjustment of the rotation about a global axis is necessary.

A more compact way to represent a rotation can be achieved by using versors, normalized quaternions, as in Equation 2.3.2-3.

$$\vec{q} = q_i\vec{i} + q_j\vec{j} + q_k\vec{k} + q_r = \begin{pmatrix} q_i \\ q_j \\ q_k \\ q_r \end{pmatrix} \tag{2.3.2-3}$$

These qauternions, a four dimensional number system, can be interpreted like the ”Euler axis and angle” approach. Each representing the direction of an axis pointing from the origin to the shell of a unit sphere and the rotation angle. Quaternions can be multiplied, using rules similar to complex numbers. With that, the orientation, or rotation of a three dimensional object can be described by only one vector.[16]

Another way to compose a rotation matrix is by the use of a basis, a triad of unit vectors. Creating a rotation matrix that describes the orientation of a shape can therefore be accomplished by using the principal axis of inertia as a basis.

$$T_r = \begin{pmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{pmatrix} \tag{2.3.2-4}$$

This is a very convincing formalism, as it allows to represent orientations computed based on physical properties, like the inertia tensor. Also, a rotation into the global coordinate system can be accomplished by simply applying the orientation to the data set. A matched orientation is achieved indirectly, as all data sets are oriented, so that their principal axes of inertia face the same directions of the coordinate system.[17]

In the implementation and this thesis, mainly rotation matrices are used to describe orientations and rotations, as they allow an intuitive way to describe both orientations, based on the axes of inertia, as well as rotations about the coordinate axes.

2.3.3 Scale

Scaling an object is accomplished by multiplying a scalar to the location of each point. This multiplication is dependent on the current position of this point, as it multiplies its distance to the origin.

There are two types of scalings, the uniform and non-uniform scaling. A uniform scaling is one with equal multiplication factors for each direction, preserving the shape of an object, or the relative displacement of two points. A non-uniform scaling results in shearing, not preserving the original shape.[20]

Due to the presented measuring methods, the point cloud data is not scaled uniformly. The grids are uniformly spaced in x- and y-direction, but not in z-direction. To avoid the necessity of non-uniform scaling, the spacing is adjusted when loading the data.

2.4 Numerical optimization

The goal of any optimization is to find a minimum or maximum of an objective function $f(\vec{x})$, with $\vec{x}$ as a vector, storing the parameters. The parameters $\vec{x}$ are real numbers, $\vec{x} \in \mathbb{R}^n$. An optimization with an open set is called unconstrained. The Newton's method is an example of an algorithm finding the roots of the first derivative $\nabla f(\vec{x})$. In each iteration method, the Jacobian matrix $\nabla f(\vec{x})$, as well as the Hessian matrix $\mathit{\Delta} f(\vec{x})$ are needed, see Equation 2.4-1.

$$x^{k+1} = x^k - (\mathit{\Delta} f(x^k))^{-1} \nabla f(x^k) \tag{2.4-1}$$

A starting point x^0 needs to be provided. As the elements of $\vec{x}$ are used to compose the transformation matrix, a decent starting point is the transformation matrix calculated based on the physical properties. Although this algorithm converges quadratically, the costs of the computation of the inverse of the Hessian are high. Instead of computing both, the Hessian can be approximated. Such a method is called Quasi-Newton and there are different update formulas for the approximation of the Hessian, such as the Davidon-Fletcher-Powell (DFP), or the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm.[12]

3 Methods

In order to apply the presented transformations, properties must be found and strategies need to be developed. In this chapter, three methods will be presented, aiming to find a geometrical matching for the CT, CAD and MPM data.

The first method only uses physical properties to compute the necessary transformations, introduced in Chapter 2.3. As the results presented in Chapter 4 will show, this method can only serve as an initial orientation and further optimization is necessary. The key challenge is the computation of the principal axes of inertia.

The second method targets the optimization by matching certain faces. This method is mainly used to visualize deformations in the CT data. With knowledge about the analyzed geometry and deviations after the transformation based on the principal axes of inertia, this method can also be used to optimize the matching of MPM data.

The third and final method aims to minimize the distance between selected vertices, mainly used to optimize the MPM matching and is independent of previous transformations or the initial orientation.

Both methods share the goal to minimize the distance between certain geometries. The selection of these geometries, as well as the correct assignment are the key challenges for both.

3.1 Transformation based on physical properties

This first method uses the center of mass, the inertia tensor and the bounds of the data sets to compute a transformation matrix T, so that they are oriented with the coordinate system. This transformation can be composed of a rotation, a translation and a scaling, as in Equation 3.1-1.

$$\vec{d_i} = T_s \cdot T_r \cdot T_t \cdot \vec{r}_i = T \cdot \vec{r}_i \tag{3.1-1}$$

In the following the composition of each matrix is derived from a physical property.

3.1.1 Translation

Since the points within a data set do not move in relation to each other during a translation of the entire data set, it is sufficient to only compute the displacement of one point. The resulting translation is then applied to all points. Finding the same existing point in each data set can be challenging and is accomplished by computing the center of mass:

$$\vec{R} = \frac{1}{M} \iiint\limits_{Q} \vec{r}\ dm = \frac{1}{M} \iiint\limits_{Q} \rho(\vec{r})\vec{r}\ dV. \tag{3.1.1-1}$$

Since a homogeneous mass distribution can be assumed, the center of mass is equal to the geometrical center, as the density is independent of $\vec{r}$.
This integral can be solved using discretization, assuming each voxel has the same volume, so that the centroid can be computed as the mean voxel:

$$\vec{R} = \frac{1}{N}\sum_{i=1}^{N}\vec{r}_i \tag{3.1.1-2}$$

where N is the amount of volumes and $\vec{r}$ their position in space. The more and thus the smaller the volumes, the better the quality of the solution. That way, the complex geometry is decomposed into simpler sub volumes. Point cloud data is already discretized by its nature. By simple vector algebra, the displacement of the centroids and consequently of the data sets can be found by subtracting the position vectors:

$$\vec{R}_1 - \vec{R}_2 = \vec{t} \tag{3.1.1-3}$$

With that, the first transformation matrix can be created. To allow the computation of the transformation as a linear function, the dimension is expanded by one. Like so, the multiplication of a translation matrix in Equation 3.1.1-4 with other transformation matrices, like rotation matrices, is possible.

$$T_t \cdot \vec{r} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_x \\ r_y \\ r_z \\ 1 \end{pmatrix} = \begin{pmatrix} r_x + t_x \\ r_y + t_y \\ r_z + t_z \\ 1 \end{pmatrix} \tag{3.1.1-4}$$

3.1.2 Rotation

As the goal of this project is the matching of two near identical geometries, the key challenge is to orient both in the same directions. For that, the orientation of a three dimensional object must be defined in some way. Here, the principal axis of inertia are introduced to provide a set of orthonormal unit vectors as a basis representing the orientation of the object.
The principal axes of inertia are determined by the eigenvalues of the inertia tensor.
The inertia tensor can be constructed by calculating the angular momentum $\vec{L}$ and the assumptions of evenly distributed mass and the center of mass being in the origin:

$$\vec{L} = \sum_{i=1}^{N}\vec{r}_i \times \vec{p}_i = \sum_{i=1}^{N}\vec{r}_i \times m_i\vec{v}_i = \sum_{i=1}^{N}\vec{r}_i \times m_i(\vec{\omega}_i \times \vec{r}_i) \tag{3.1.2-1}$$

with $\vec{p}$ as the momentum, m the mass, $\vec{v}$ the velocity and $\vec{\omega}$ the angular velocity of each point. In matrix notation and because every point or voxel has the same mass, Equation 3.1.2-1 can be written as:

$$\vec{L} = I \cdot \vec{\omega} = \begin{pmatrix} \sum_{i=1}^{N} y_i^2 + x_i^2 & \sum_{i=1}^{N} -x_i y_i & \sum_{i=1}^{N} -x_i z_i \\ \sum_{i=1}^{N} -y_i x_i & \sum_{i=1}^{N} z_i^2 + x_i^2 & \sum_{i=1}^{N} -y_i z_i \\ \sum_{i=1}^{N} -z_i x_i & \sum_{i=1}^{N} -z_i y_i & \sum_{i=1}^{N} x_i^2 + y_i^2 \end{pmatrix} \cdot \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{3.1.2-2}$$

I is called the inertia tensor and will be referred to as:

$$I = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} \tag{3.1.2-3}$$

An axis of rotation with angular momentum $\vec{L}$ and angular velocity $\vec{\omega}$ facing the same direction is called principal axis of momentum, the off-diagonal elements become zero. Therefore the principal axes are the eigenvectors of I. To compute the eigenvalues of I, the characteristic polynomial in Equation 3.1.2-4 is solved.

$$det(I - \lambda E) = det \begin{pmatrix} I_{xx} - \lambda & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} - \lambda & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} - \lambda \end{pmatrix} = 0 \tag{3.1.2-4}$$

If two eigenvalues are equal, there are multiple solutions. This is the case for a circular or hexagonal rod. Only the principal axis in the direction of the rod is unique and every other axis perpendicular to this axis and going through the center of mass is a principal axis. A precise transformation based on the principal axes of momentum into the global coordinate system is therefore only possible, if the object is not rotational symmetrical.[2]
Using these principal axes, the rotation matrix can be assembled as described in Chapter 3.1.2. For that, the principal axes of inertia $\vec{a}$, $\vec{b}$ and $\vec{c}$ are stored in a matrix with the three components in each row (see Equation 3.1.2-5).

$$T_r = \begin{pmatrix} \vec{a} & \vec{b} & \vec{c} \end{pmatrix} \tag{3.1.2-5}$$

Again extending the dimension by one results in Equation 3.1.2-6.

$$T_r = \begin{pmatrix} a_x & a_y & a_z & 0 \\ b_x & b_y & b_z & 0 \\ c_x & c_y & c_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.1.2-6}$$

3.1.3 Scale

After transforming the data sets into the coordinate system they only need to be scaled. The relative scale of one shape to another can be calculated using the bounds of both shapes in a plane. For uniform scaling, the bound in one direction is sufficient, for a nonuniform scaling, the bounds in all three directions must be compared.
To find the bounds of a body, the minimal bounding box is calculated. If the scaling is applied last, the edges of the minimal bounding box of the point clouds are parallel to the coordinate axes. This box is called axis-aligned bounding box. This box can be found for a point cloud, by finding the minimal and maximal values for each direction within its points.[3] While providing a precise measure for the point cloud data, the minimal bounding box of the CAD data is not precise. Because it is likely that a control point of a NURBS surface or edge is the element farthest away from the origin, the computation of the axis-aligned bounding box is not a precise measure. In the case of the impeller, the shape is aligned to the axes of the coordinate system by the computed rotation. The front side of the shape is parallel to an axis. As the dimensions of the shape are known, the scaling factor can be computed using those, instead of the bounding box. However, this only applies to this specific shape and can not be generalized. For other geometries, the shape could be meshed. Computing the axis aligned bounding box of this mesh can be achieved as it is for the point cloud data. This method was tested for the impeller and produced precise scaling factors.
With that, a four by four matrix can be built, with the scaling factors $\vec{s}$ lying on the diagonal, see Equation 3.1.3-1.

$$T_s \cdot \vec{r} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_x \\ r_y \\ r_z \\ 1 \end{pmatrix} = \begin{pmatrix} r_x \cdot s_x \\ r_y \cdot s_y \\ r_z \cdot s_z \\ 1 \end{pmatrix} \tag{3.1.3-1}$$

For a uniform scaling all factors are equal:

$$s_x = s_y = s_z \tag{3.1.3-2}$$

resulting in:

$$T_s \cdot \vec{r} = s \cdot \vec{r} \tag{3.1.3-3}$$

The uniform spacing of the grid simplifies Equation 3.1-1 to:

$$\vec{d} = s \cdot T_r \cdot T_t \cdot \vec{r} = T \cdot \vec{r} \tag{3.1.3-4}$$

The transformation into the global coordinate system for each data set is represented by a four times four matrix T.

3.2 Finding a transformation with minimization

As Chapter 4.2.1 will show, the method based solely on the calculation of the physical properties is not without flaws.
To improve the matching, a transformation can be found by minimizing the displacement of to data sets by solving Equation 3.2-1 for $\vec{x}$. Using the BFGS algorithm, $f(\vec{x})$ can be minimized.

$$f(\vec{x}) = \sum_{i=1}^{N} \left\| \vec{r}_i \cdot \vec{x} - \vec{d}_i \right\|^2 \tag{3.2-1}$$

For that, two point arrays representing the points of the point cloud $\vec{r}$ and the vertices of the CAD shape $\vec{d}$ must be created. Those arrays are filled with selected points, representing the geometry. Like so, a transformation can be found, matching the selected points. Arun et al. present a Least-Squares Fitting algorithm for point clouds similar to this one.[1]
In the following two strategies are introduced to choose the points which take part in the minimization. The selection and assignment of those points are the key challenges of each minimization method.

3.3 Matching faces

The first approach is the selection of points on a face. For simple geometries like the T-shape, these points can be found in the point cloud using the boundary box. More complex geometries demand manual selection by user input. This method is used to align CT and CAD data and to improve transformations with regards to one or more faces. As mentioned before, the CT data represents the deformed shape of the produced work piece. To visualize these deformations, one or more faces can be aligned, enabling measurements relative to those.

3.3.1 Face selection

A face of a CAD shape is accessible as a NURBS surface in the STEP file. Unfortunately, there is no information in a VTK file describing the faces of the point cloud.
The first step in the selection of a certain face is to exclude all points not located on the hull of the point cloud. The marching cubes algorithm does that by using a look-up table and information about the neighborhood of each point.
The marching cubes algorithm was introduced in 1987. It enables the fast visualization of tomography scans, as it reduces the amount of visualized data significantly. The algorithm divides the point cloud into cubes, each made of eight vertices. Each point is assigned a binary value, depending whether its data value exceeds a threshold defined by the user. A look-up table is used to identify the topology within the cube. If a point is assigned a zero, it lies outside of the surface, otherwise it lies within and is therefore a vertex of the triangulated surface. The algorithm then marches to the next cube. For each cube there are $2^8 = 256$ possibilities. Due to symmetries of the potential cases, only 14 patterns need to be distinguished.[8] For the presented use-case of selecting the points on the surface of a point cloud, the vertices of the triangulated surfaces can be used. Unfortunately, these vertices are

not necessarily points of the point cloud, but are created in between the grid. Therefore, the precision of this approximation depends on the resolution of the grid.
To select one face, the location and orientation are analyzed by observing the point cloud in the coordinate system. This is a very cumbersome process, algorithms detecting edges and with that segmenting the surfaces into its faces are mentioned in Chapter 5.
A random selection of surface points is used in Chapter 4.2.3 to measure the distance between the surface of the CAD shape and the point cloud.

3.3.2 Face assignment

As Equation 3.2-1 demands an array of points, such must be created on the face of the CAD shape. This can be achieved by projecting the points of the point cloud onto the corresponding face of the CAD shape.
To project a point onto a face, a minimization problem has to be solved. The projected point $\vec{r}\,'$ lies on the face and on the normal of that face going through point $\vec{r}$. Therefore, $\vec{r}\,'$ is the point on the face closest to $\vec{r}$. With this in mind $\vec{r}\,'$ can be found by minimizing the target function, with $S(u,v)$ describing the surface.

$$\vec{r}_i\,' = min \parallel S(u,v) - \vec{r} \parallel^2 \tag{3.3.2-1}$$

Projecting and matching more than one face can be challenging. The main idea of this approach is that the projected points describe the target locations of the points. If the deviation of the point cloud is only about one axis, this works well. If more than one direction needs to be optimized, projecting on all faces does not produce the desired matching, as Figure 3.3.2-1a pictures for two dimensions. To match both geometries, both edges must be aligned. The points of those edges in the point cloud, pictured in red, are projected onto those of the CAD data. The projected points, highlighted in green, do clearly not describe the target location of the point cloud in the first step. A minimization will result in a unsatisfactory matching.
For a perpendicular corner the consecutive transformation of the edges will result in a precise matching, since all translations are in the direction of the other edge. As Figures 3.3.2-1b and 3.3.2-1c picture, this does not apply to corners that are not perpendicular. A matching based on more than one face is therefore only possible, if the projection is computed in every iteration,

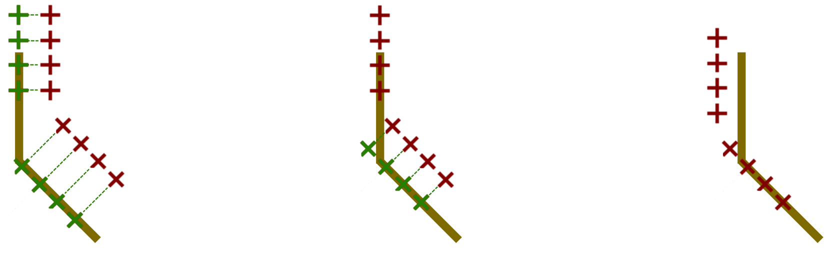

(a) (b) (c)

Figure 3.3.2-1: Matching of two edges

of the minimization. In this work the matching of only one face is executed. Nonetheless, the strategy to match perpendicular faces consecutively is verified by the T-shaped test data in Chapter 4.1.2.

3.4 Matching vertices

For this second approach the points in the corners are selected. As the corner points are present in both a point cloud and a CAD shape, there is no need for a projection. The selection of the points is the key challenge here.

3.4.1 Vertex selection

Because they are stored directly in the STEP file as vertices, they can be accessed easily and provide a precise specification of the location. Finding vertices in the point cloud data is not as trivial. As the use-case is a prototyping project, a manual selection via user-input is aspired. A challenge regarding this selection is the resolution and with that, the sharpness of the corners. The lower the resolution, the more difficult it is to select the correct vertices by eye. As a result, the precision of the selection is at most as great as that of the measuring method. This method qualifies especially for the matching of MPM data and CAD data, since they both describe the same geometry. Therefore, it can be assumed that a perfect fit can be achieved when all vertices are matched.

3.4.2 Vertex assignment

Assigning the vertices of the point cloud to those in the CAD shape is accomplished by positioning in the same order in each array. Again, this is done manually.

4 Results

In this chapter the results of all three previously presented methods are discussed. First, the introduced T-shaped example data is matched, using all methods to verify them. The results are presented and compared to one another. Then the same methods are applied to the impeller data. To compute these methods, a script was implemented, reading STEP and VTK files and matching them by applying the presented methods. The script was programmed using the OpenCASCADE [14] python bibliography for all NURBS and point cloud analysis, such as the computation of the physical properties and minimal bounding boxes and for the visualization of the data. Moreover, VTK [15] was used to read the provided point cloud files and apply thresholds and the marching cubes algorithm. Finally, Scipy [6] was used to provide the implementation of the minimization algorithm.

4.1 Geometry matching of test data

To verify the the proposed methods and measure their accuracy they are applied to the T-shaped test data first.

4.1.1 Analysis of the approach based on physical properties

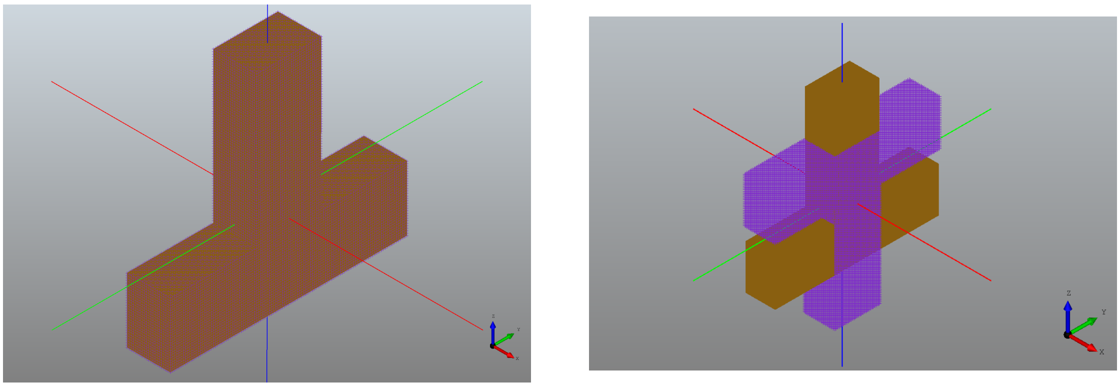
The point cloud is analyzed and transformed, so that a geometrical matching with the CAD data is achieved. The principal axes of inertia, as well as the centers of mass of both data sets are computed.

The principal axes in y- and z-direction are given by the symmetry-axes and the principal axis in x-direction is therefore the x-axis, to fulfill orthogonality. Computing the principal axes of inertia as described in chapter 3.1.2, the unit matrix is outputted for both data sets. Due to their initial orientation after loading, with all edges and faces parallel to a coordinate axis, this is expected and verifies their correct calculation.

In a structured grid with equally weighted points, the center of mass equals the geometric center. The geometric center can be easily found by decomposition of the shape. Table 4.1.1-1 shows the absolute and relative error of the computed center of mass of T-shaped point clouds with various resolutions. It is clear to see, that the precision increases with the resolution. Applying the computed transformation based on these properties to the point cloud results in the matching depicted in Figure 4.1.1-1. On the left side of the figure, the matching was successful, on the right side it clearly was not. While the initial orientation of the shape in Figure 4.1.1-1a is only slightly off its target orientation, the starting orientation of the shape in Figure 4.1.1-1b is far off. Diagram 4.1.1-2 shows the mean error depending on the initial orientation. The error is computed using eight vertices and measuring the distance between corresponding points in the CAD and point cloud data. In all three dimensions

Resolution [1/mm]	Absolute error [mm]	Relative error
0.5	0.1709	0.004963
1	0.0859	0.002495
2	0.0431	0.001251
3	0.0288	0.0008348
4	0.0216	0.0006258

Table 4.1.1-1: Absolute and relative error of the z-value of the center of mass depending on the resolution of the T-shaped point cloud



(a) Successful matching

(b) Unsuccessful matching

Figure 4.1.1-1: Geometry matching of a T-shape by computation of its physical properties

an initial orientation rotated by more than 45 degrees off the target orientation results in erroneous transformations. In all cases the principal axes of inertia and the center of mass are matched correct. This is caused by the ambiguity of the rotation axes. A rotation by 180 degrees around one of the principal axes will also result in a precise matching of the axes, but an incorrect matching of the shapes. Therefore, to achieve correct transformations, the initial orientation must be considered. When looking at Diagram 4.1.1-2 again, even for

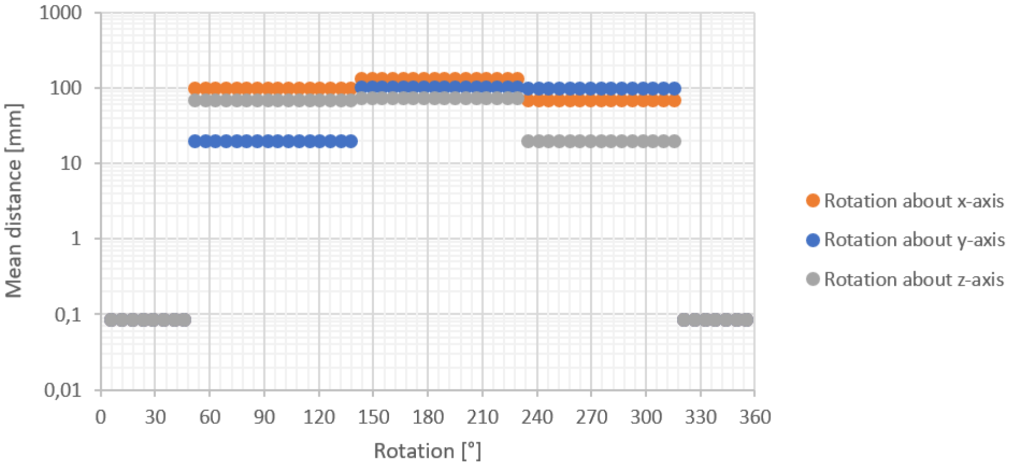


Figure 4.1.1-2: Mean error depending on initial orientation

the visually correct transformations, a mean error of about $\epsilon = 0.085$ is outputted. Diagram

4.1.1-3 compares this error to that in the calculation of the center of mass depending on the resolution of the point cloud.

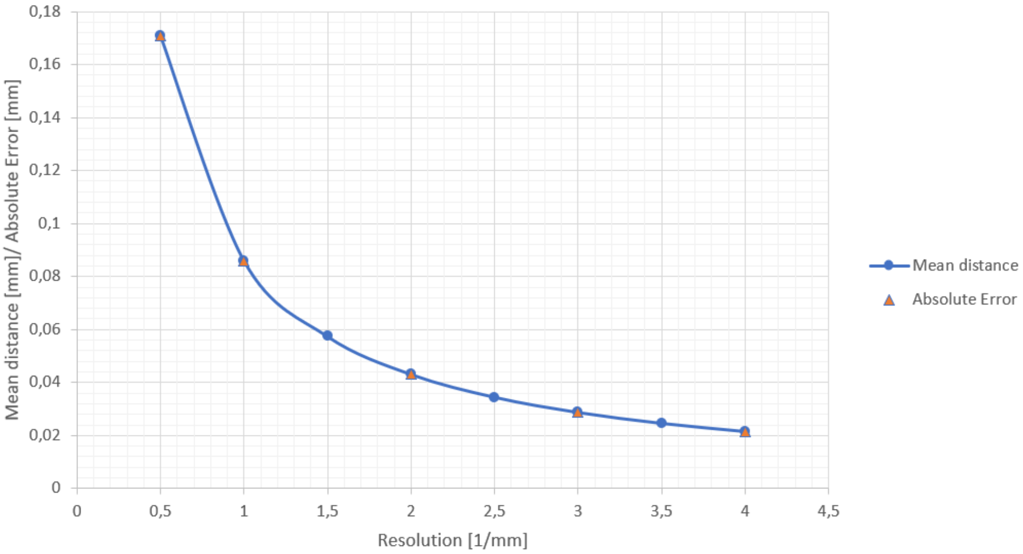
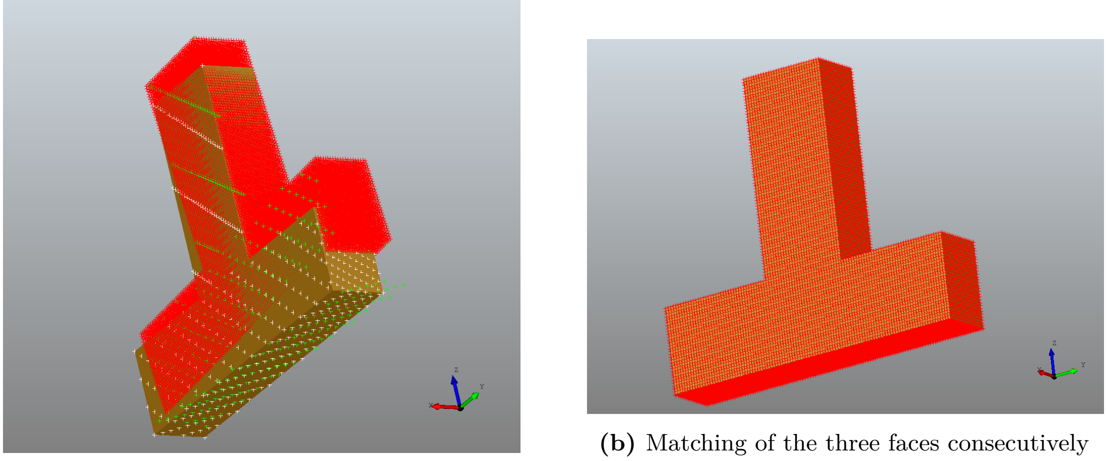


Figure 4.1.1-3: Mean error of the transformation compared to the mean error of the center of mass depending on the resolution

It becomes clear that the error in the transformation is caused by the error in the computation of the center mass.
To compute a precise transformation to match both data sets based on their physical properties, the initial orientation and resolution of the point cloud must be considered.

4.1.2 Analysis of the numerical optimization

To decrease the influence of these restrictions, the transformation is optimized. For that, the measured error is minimized. To accomplish that, first the method of matching faces is used. As before, the quality is measured with the use of eight vertices.
For the matching, three faces are selected. Two different approaches are tested, the first one selecting, assigning and projecting all three faces at once and the second one to project them consecutively. To verify the methods, described in Chapter 3.3.2, the T-shape is matched, using the points in the faces, pictured in white in Figure 4.1.2-1a. As predicted, the matching is not successful, as the projected points, pictured in green, are clearly not at the target locations. Therefore, the point cloud in red is not matched correctly to the CAD data.
Diagram 4.1.2-2 shows the mean error of the method and how it depends on the initial orientation. A correct transformation can only be achieved, within the bounds of 90 degrees deviation about each axis. If the angular deviation is greater than 90 degrees, the faces are matched backwards. Also, the error increases, even within the bounds of 90 degrees. The increasing error for the correct transformations can be avoided by locking certain transformations after each face matching.
As discussed, for simple geometries with perpendicular faces only, like this one, a consecutive matching of the faces can provide a better quality for the transformation. After one face is matched, only the rotation about the axis normal to that face and the translations in the two



(a) Matching of all three faces at once

(b) Matching of the three faces consecutively

Figure 4.1.2-1: Matching three faces of the T-shape

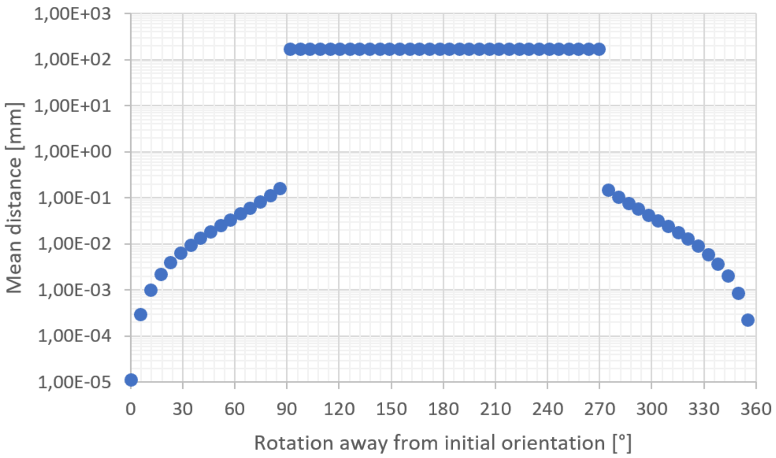


Figure 4.1.2-2: Mean error after face matching without locked transformations depending on the initial orientation

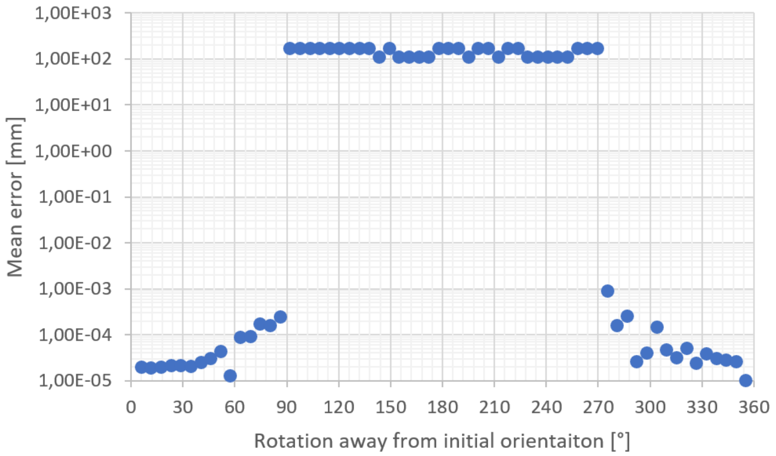


Figure 4.1.2-3: Mean error after face matching with locked transformations depending on the initial orientation

directions spanning the plane of the face are allowed. For the third face only the translations in the direction of its normal is allowed. In Diagram 4.1.2-3 the results of this measure can be seen. While the maximal error for a correct matching is greater than 0.1 for the first method with all faces being matched at the same time, it does not surpass 0.001 when only certain entries of the transformation matrix are used in each step. Again, this approach can only be applied to faces perpendicular to each other.
To avoid the restrictions of perpendicularity and initial orientations, vertices offer a more stable approach. As no projection is necessary, there is only one unique solution to minimize the distance between two points. For that, eight vertices are selected and assigned to their counterparts in the CAD data. Diagram 4.1.2-4 displays the mean distance depending on the rotation away from target orientation. The error is measured using the three faces, already selected for the face method. As they are only used to measure the distance between their location and the location of their projected counterparts, the measuring is precise. As one can see, the quality of transformation is independent of the initial orientation. Therefore, minimizing the distance between certain vertices, present in both data sets, is the most stable and precise method to geometrically match two generated data sets.

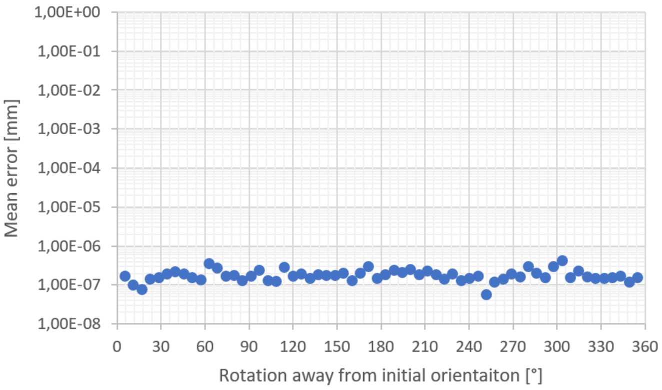


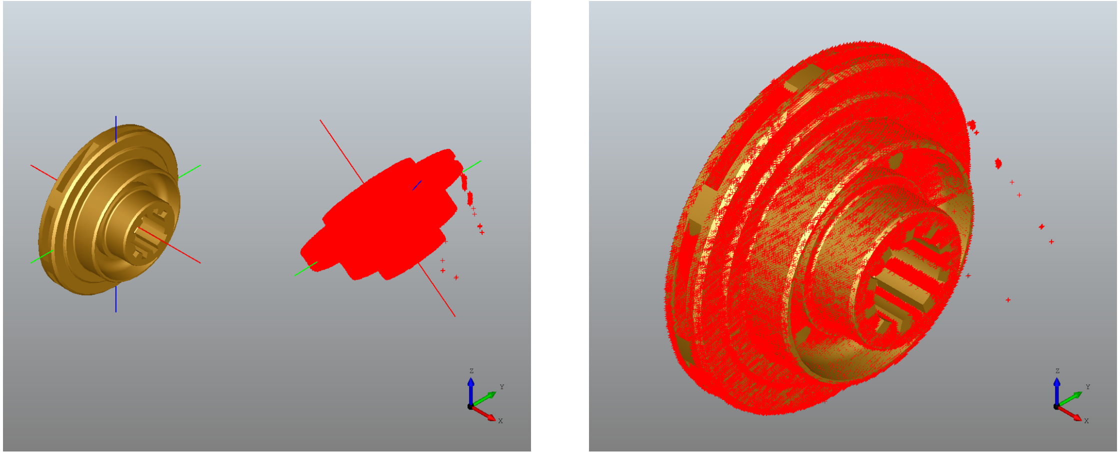
Figure 4.1.2-4: Mean error after vertex matching depending on initial orientations

4.2 Application for measured data

While the matching of the example data is only restricted by the initial orientation and resolution of the point cloud, there are far more hurdles to take for measured data with more complex geometries. For the transformation based on physical properties a threshold is applied, to exclude the supporting structure, present in the data. The optimized transformations on the other hand are computed using the full data set, without a threshold, to allow more precise point selection. To visualize the matching, the threshold is applied after the transformation.

4.2.1 Using the physical properties

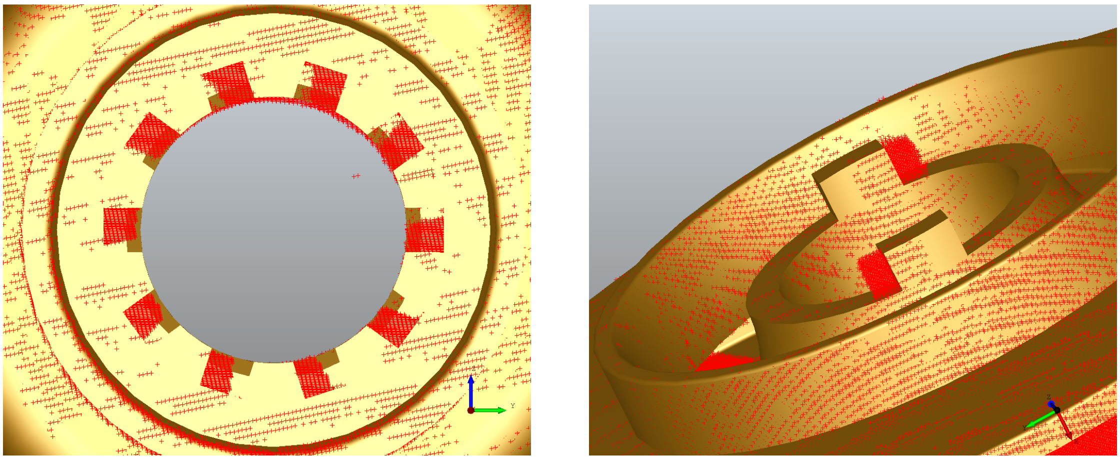
The method solely based on physical properties is not precise for the impeller (see Figure 4.2.1-1). While the scale and position look like a good fit, the rotation about the x-axis is far off. The teeth of the gear and the blades do not align correctly. In Figure 4.2.1-2a the



(a) Before matching **(b)** After matching

Figure 4.2.1-1: Geometry matching of an impeller by computation of its physical properties

incorrect rotation about the x-axis can be analyzed by visual observation. The hubs of the jaw coupling in Figure 4.2.1-2b illustrate this deviation even more. As the impeller is close to being rotational symmetrical and the x-axis being the axis of (near-)symmetry, this is likely to be the reason why the transformation is not successful. When computing the principal axes of inertia, as in Equation 3.1.2-4, symmetries cause inconclusive solutions for the eigenvalues. If symmetries occur in an object, the symmetry axes are always principal axes of inertia. For a rotational symmetry, only one principal axis is defined. All other axes perpendicular to the first, are principal axes of inertia. Therefore, only for a symmetry free object it is possible to compute three conclusive principal axes of inertia precisely.



(a) Teeth of a gear **(b)** Hubs of a jaw coupling

Figure 4.2.1-2: Close ups of matching of CAD and MPM data based its on physical properties

Another error source can be detected. To the right of the impeller in Figure 4.2.1-1b residue of the supporting structure can be seen. In the particular case of measuring MPM data in the process of SLM, cleaning the data from its supporting structure is necessary. A supporting structure is manufactured with the objective to allow for designs with overhangs.

The collected data needs to be cleaned from this supporting structure, as it is not part of the work piece and with that not present in the CAD or CT data. Because it only supports the weight of the work piece during the manufacturing process, the structure is produced with low intensities and thin wall thickness. This allows to exclude this part of the data by applying a threshold. As setting the threshold higher removes data of the geometry, this residue can not be removed.

To analyze whether the existence of residue causes the error, the matching of the CAD and MPM data can be compared to the matching of CAD- and CT data. As mentioned in Chapter 2.2.3, the CT data does not include a supporting structure, therefore a threshold is not applied. This allows to determine the impact of residue and the threshold for the matching based on the center of mass and the inertia tensor. In Figure 4.2.1-3, the matching of CT and CAD data is displayed. The rotational deviation around the z-axis is present in this matching, too. This indicates that the impact of the symmetry is the reason for the deviation and not the residue.

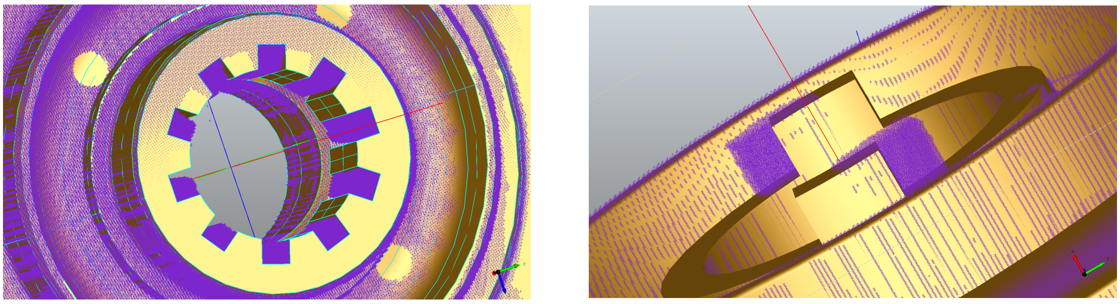


Figure 4.2.1-3: Results of the matching of CT and CAD data

4.2.2 Optimization of the transformation

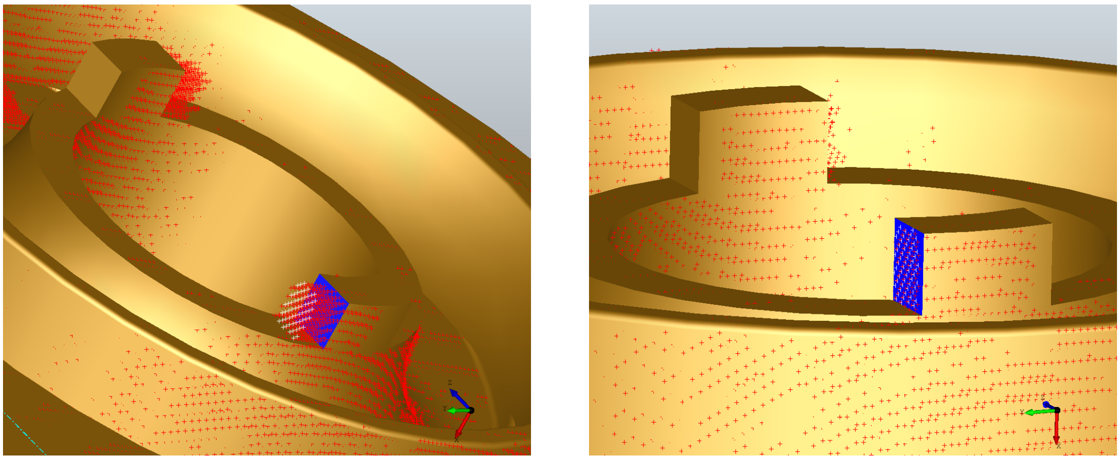
In the following, both methods, face matching, as well as vertex matching are applied in order to improve the matching. The face matching method is used to test the deformation measuring for CT data. The corner method and the face method are used to find a precise fit for the MPM data and are compared with each other.

Face method

First, a face is selected, to allow the minimization of the rotational deviation.

Selecting a face, with its normal facing the direction of the rotational deviation is ideal. In Figure 4.2.2-1a, the selected face is depicted in blue. The points in the point cloud, representing the same face are depicted in white. Figure 4.2.2-1b shows the success of minimizing the rotational deviation about the x-axis. The deviation is not eliminated entirely, but comparing the result with the transformation based on physical properties in Figure 4.2.1-2, the quality of the matching increased significantly.

The main application of this method is the visualization of deformations. For that, the CT data was matched with the CAD data. The face at the end of the gear is matched, to minimize rotational deviations about the y- and z-axis. Figure 4.2.2-2 shows the result. As one can see,



(a) Selection of a face in both data sets

(b) Successful minimization of the rotational deviation

Figure 4.2.2-1: Optimization using a face

it is possible to match the data sets precisely enough, to picture deformations occurring at the outer ring on the side of the internal gear. The CAD shape is pictured transparent, visualizing the geometry of the ring as a sectional view. As the wall thickness decreases towards the right hand of the figure, the stiffness decreases, too. It is likely, that this is the reason for the thermal deformations at this location.

Various faces can be matched to further analyze the impact of the component geometry and the manufacturing process.

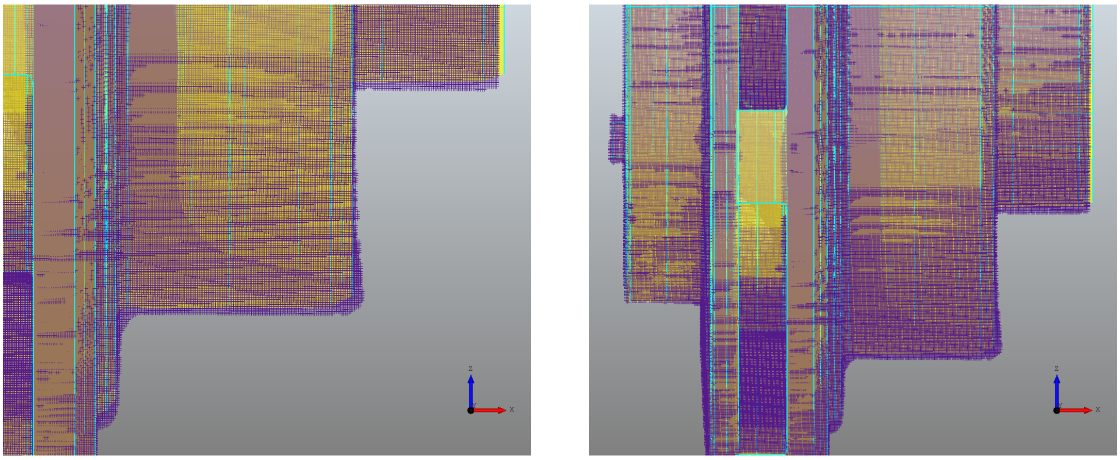
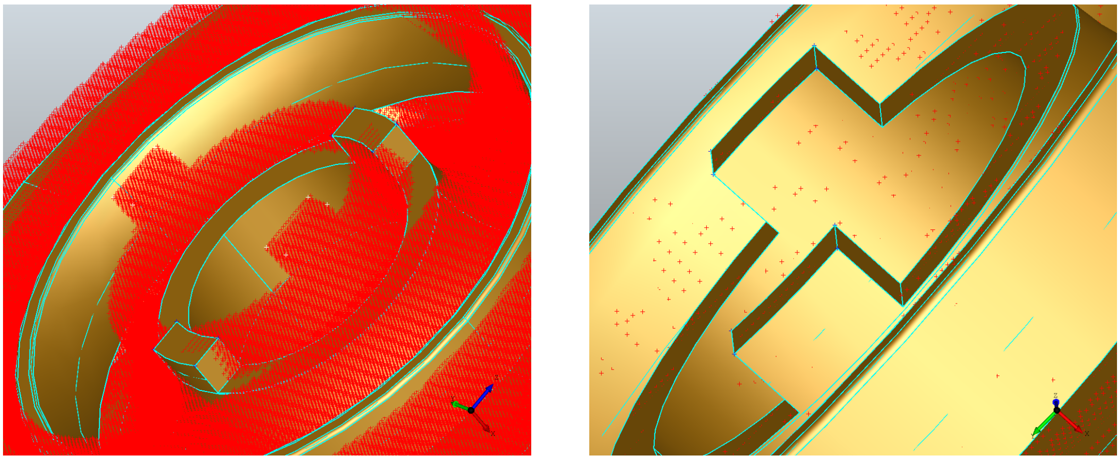


Figure 4.2.2-2: Matching of CT and CAD data to locate deformation with the use of the face method

Vertex method

As discussed in chapter 4.1.2, an optimization based on vertices is more precise and stable. Unfortunately, the selection of vertices is a challenge by itself. As described, the points are selected in the original data, with no threshold applied. This is necessary, to allow a precise selection of the corners, as these are likely to get removed by a threshold. Selecting the vertices in the MPM data, pictured in white in Figure 4.2.2-3a and minimizing the distances to their counterparts in the CAD data, a matching can be achieved. The result is displayed in Figure 4.2.2-3b.



(a) Selection of eight vertices in both sets

(b) Successful minimization

Figure 4.2.2-3: Optimization using vertices

4.2.3 Measuring the error of the methods

The quality of the transformations can be measured using the face method. The points on the surface are selected using the marching mubes algorithm, introduced in Chapter 3.3.2. To reduce the costs of the computation, only a few thousand of these points are selected. To assign each point to its corresponding face, every point is projected onto every face. The projected point, nearest to each point is expected to lie on the corresponding face. Iterating through all selected points, a mean distance can be calculated, which can be used as a error measure. This method is erroneous for points at edges and vertices. Depending on the orientation, the closest face to such points can differ. Nonetheless this method can be used to approximate the mean error. The results of the measurements for all applied methods are presented in Table 4.2.3-1. The computed errors give a rough approximation of the quality of each transformation. Both, the mean distance of the selected points, as well as the maximum norm are compared.

Method	Mean distance [mm]	Maximal distance [mm]
Physical properties	0.131	1.88
Face matching	0.128	1.22
vertex matching	0.101	1.24

Table 4.2.3-1: Comparison of the quality of the used matching methods

The quality of the transformation achieved by the vertex method is the best, measured by the mean distance. This shows that this method is the most precise and should be used to compare CAD and MPM data.

The transformation based on physical properties is not as precise. The mean distance is only slightly greater, the maximal error on the other hand, is significantly larger. While most points are near their target position, some are far off. These points are the ones pictured in

Figure 4.2.1-2b and on other geometries, where the rotational deviation is recognizable.

The mean distance of the transformation solely based on the physical properties and of the transformation with minimized rotational deviation differ only slightly. Comparing the visualizations of both (Figure 4.2.1-2b and Figure 4.2.2-1b), the quality of the optimized transformation is far better. The reason for this difference is the small and random selection of points on the surface of the point cloud. Only very few points are selected that lie on geometries like the teeth of the gear where the rotational deviation has an impact on the distance to the nearest face. Even if all surface points would be selected, the share of points on those geometries would be very low, because they only make up a small percentage of the point cloud. The success of the optimization using the face is identifiable in the maximal distance, which is far smaller than the one without optimization.

Both presented methods, one using points on faces and the other using vertices resulted in significant improvements. While the face method lacks reliability, for customized optimizations it can be adjusted easily and individually. The corner method is the most reliable and precise of all, but has the problem of elaborate point selection and assignment.

Using the face method to improve the transformation is only possible, because the matching based on the physical properties was analyzed and interpreted by the user. This step can not be automated. Nonetheless it shows how interaction between algorithms and users can generate precise matchings. This could easily be extended, allowing the user to interfere and select certain faces, vertices or edges to match.

5 Conclusion

The goal of this work to precisely match different data sets was achieved. A precision in the range of the layer thickness of the selective laser melting machine seems possible with the implemented methods, if the user is integrated in the process. With that, a precise analysis of the manufacturing quality and process parameters is possible.

The first approach of calculating and using the center of mass, the dimensions and the principal axes of inertia resulted in decent matchings for basic geometries, like the presented T-shape. For more complex geometries, like the impeller, the computed transformation matrix was erroneous. The rotational symmetry of the geometry was identified as the main reason for that. Furthermore, the behavior of the matching was analyzed in dependency of the initial orientation and the resolution of the provided data. It turned out that the method only works for a relatively small range of starting conditions. In fact the rotation away from the target orientation must not exceed 45 degrees in each direction in order to succeed. Furthermore it could be shown, that the resolution has a significant impact on the quality of the calculation of the center of mass.

With these restrictions, a more stable and precise approach was taken. For that, representative points were selected and assigned to their corresponding counterparts. Two selection strategies were tested, one selecting faces, to allow for matching of certain faces and planes, the other selecting vertices to achieve a precise matching. Minimizing the sum of distances between the points and their counterparts, a precise matching is possible. The face method proved to be especially useful for matching CT and CAD data. Vertex matching resulted in more precise matchings for MPM and CAD data.

To further increase the precision of the vertex strategy, more points than the limited number of eight can be selected. In order to simplify this selection and to make the implementation available and easily usable, a user interface could be provided. Both the vertex and face method an be automated more. As the selection of the vertices and identifying of faces are the key challenges, Weber et al. [19] present an algorithm detecting sharp features and Mineo et. al [9] present an alternative approach of selecting surface points. Furthermore, the distance between each point and its corresponding face of the CAD data can be visualized, using a color bar.

Lastly the provided information about thermal distributions and deformations can be used to set the parameters in the manufacturing process, like the intensity of the laser or its route. Even a tool, assisting the engineer in the design of the work piece, so that a manufacturing-specific design is achieved, could be possible.

This enable not only to manufacture components that were previously fabricated conventionally with higher precision, but also to develop new solutions with less restrictions due to manufacturing.

Bibliography

[1] K Somani Arun, Thomas S Huang, and Steven D Blostein. “Least-squares fitting of two 3-D point sets”. In: *IEEE Transactions on pattern analysis and machine intelligence* 5 (1987), pp. 698–700.

[2] Matthias Bartelmann et al. “Starre Körper”. In: *Theoretische Physik 1 / Mechanik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, pp. 127–164.

[3] Murilo G. Coutinho. “Hierarchical Representation of 3D Polyhedra”. In: *Guide to Dynamic Simulations of Rigid Bodies and Particle Systems*. London: Springer London, 2013, pp. 29–88.

[4] Andres Gasser et al. “Laser Additive Manufacturing: Laser Metal Deposition (LMD) and Selective Laser Melting (SLM) in Turbo-Engine Applications”. In: *Laser Technik Journal* 7.2 (2010), pp. 58–63.

[5] Gabor T. Herman. “An Overview of the Process of CT”. In: *Fundamentals of Computerized Tomography: Image Reconstruction from Projections*. London: Springer London, 2009, pp. 27–36.

[6] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. [Online; accessed 10.02.2019]. 2001–.

[7] Jean-Pierre Kruth et al. “Selective laser melting of iron-based powder”. In: *Journal of materials processing technology* 149.1-3 (2004), pp. 616–622.

[8] William E Lorensen and Harvey E Cline. “Marching cubes: A high resolution 3D surface construction algorithm”. In: *ACM siggraph computer graphics*. Vol. 21. 4. ACM. 1987, pp. 163–169.

[9] Carmelo Mineo, Stephen Gareth Pierce, and Rahul Summan. “Novel algorithms for 3D surface point cloud boundary detection and edge reconstruction”. In: *Journal of Computational Design and Engineering* 6.1 (2019), pp. 81–91.

[10] Tatiana Mishurova et al. “Influence of manufacturing parameters on microstructure and subsurface residual stress in SLM Ti-6Al-4V”. In: *Advanced additive Manufacturing Symposium AAMS18*. Sept. 2018.

[11] Michael J Pratt. “Introduction to ISO 10303—the STEP standard for product data exchange”. In: *Journal of Computing and Information Science in Engineering* 1.1 (2001), pp. 102–103.

[12] Rüdiger Reinhardt, Armin Hoffmann, and Tobias Gerlach. “Lösungsverfahren für Optimierungsprobleme ohne Nebenbedingungen”. In: *Nichtlineare Optimierung: Theorie, Numerik und Experimente*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 75–296.

[13] David F Rogers. *An introduction to NURBS: with historical perspective.* Elsevier, 2000.

[14] Open CASCADE SAS. *Open CASCADE Technology, 3D modeling & numerical simulation.* 2008.

[15] Will J Schroeder, Bill Lorensen, and Ken Martin. *The visualization toolkit: an object-oriented approach to 3D graphics.* Kitware, 2004.

[16] John Vince. *Quaternions for computer graphics.* Springer Science & Business Media, 2011.

[17] John Vince. *Rotation transforms for computer graphics.* Springer Science & Business Media, 2011.

[18] C Vyas et al. “3D printing of biocomposites for osteochondral tissue engineering”. In: *Biomedical Composites.* Elsevier, 2017, pp. 261–302.

[19] Christopher Weber, Stefanie Hahmann, and Hans Hagen. “Sharp feature detection in point clouds”. In: *2010 Shape Modeling International Conference.* IEEE. 2010, pp. 175–186.

[20] Leland Wilkinson. “The grammar of graphics”. In: *Handbook of Computational Statistics.* Springer, 2012, pp. 375–414.

Erklärung zur Bachelor-Thesis

„Ich versichere hiermit, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Mir ist bewusst, dass sich die Hochschule vorbehält, meine Arbeit auf plagiierte Inhalte hin zu überprüfen und dass das Auffinden von plagiierten Inhalten zur Nichtigkeit der Arbeit, zur Aberkennung des Abschlusses und zur Exmatrikulation führen können.“

Ort, Datum	Unterschrift