

Accurate and reliable extraction of surfaces from image data using a multi-dimensional uncertainty model



Christina Gillmann^{*,a}, Thomas Wischgoll^b, Bernd Hamann^c, Hans Hagen^a

^a University of Kaiserslautern, Germany

^b Wright State University, Dayton

^c University of California, USA

ARTICLE INFO

Keywords:

Surface extraction

Uncertainty visualization

Parameter space exploration

ABSTRACT

Surface extraction is an important step in the image processing pipeline to estimate the size and shape of an object. Unfortunately, state of the art surface extraction algorithms form a straight forward extraction based on a pre-defined value that can lead to surfaces, that are not accurate. Furthermore, most isosurface extraction algorithms lack the ability to communicate uncertainty originating from the image data. This can lead to a rejection of such algorithms in many applications. To solve this problem, we propose a methodology to extract and optimize surfaces from image data based on a defined uncertainty model. To identify optimal parameters, the presented method defines a parameter space that is evaluated and rates each extraction run based on the remaining surface uncertainty. The resulting surfaces can be explored intuitively in an interactive framework. We applied our methodology to a variety of datasets to demonstrate the quality of the resulting surfaces.

1. Introduction

Surface extraction is an important task in the image processing pipeline. The goal is to transform selected pixels of the input image into a surface, representing the boundary of the object visible in the image [1]. These surfaces are used in different applications, for example to analyze the size, position and shape of tumors in the human body [2].

The extraction of surfaces was subject of many prominent algorithms during the last decades (see Section 2). Unfortunately, surface extraction methods are not widely spread in many applications. A major problem with these algorithms is the lack of uncertainty quantification and communication [3]. Real world datasets can be highly affected by uncertainty, meaning that domain scientists cannot determine the objects captured in the image data with absolute certainty. When exposing these experts with a surface extraction, they tend to reject them as they cannot rate the reliability and accuracy of the extraction algorithm's output. In addition to that, surface extraction algorithms work on a globally selected isovalue, determining the resulting surface. This assumption is wrong in many cases as the actual surface can alter slightly around the predefined isovalue [4].

In order to solve the mentioned problems, the goal is to design a surface extraction algorithm, that outputs a reliable and accurate surface. Therefore, this paper presents a novel surface extraction

methodology, that is able to quantify the uncertainty of each image pixel as a multi-dimensional vector. This uncertainty space is utilized to optimize an initially extracted surfaces such that the remaining surface uncertainty becomes minimal. Therefore, the presented method evaluates a high-dimensional parameter space, performs multiple surfaces optimizations and rates the resulting surfaces to present the best results and its corresponding parameters to the user. Finally, the user can inspect the best results in an interactive system through comparing different surfaces with each other as well as identifying uncertain areas in selected surfaces (see Section 4).

Therefore, this paper contributes:

- An optimization approach for surfaces based on an high-dimensional uncertainty model
- A quantification of surface uncertainty (global and local)
- An intuitive visualization to explore and compare surfaces

To show the effectiveness of our approach, we tested our methodology by reconstructing surfaces from predefined objects and compared our results to a state of the art marching cubes algorithm outputs. In addition to that, the algorithm was applied to a variety of real world datasets and it can be shown, that the overall error can be minimized (see Section 5). At last, the paper will be concluded and future

* Corresponding author. +4915119168698

E-mail addresses: c.gillma@cs.uni-kl.de (C. Gillmann), thomas.wischgoll@wright.edu (T. Wischgoll), hamann@cs.ucdavis.edu (B. Hamann), hagen@cs.uni-kl.de (H. Hagen).

<https://doi.org/10.1016/j.gmod.2018.07.004>

Received 8 November 2017; Received in revised form 25 April 2018; Accepted 23 July 2018

Available online 31 July 2018

1524-0703/ © 2018 Elsevier Inc. All rights reserved.

directions are given in [Section 7](#).

2. Related work

Surface extractions have been object of research since decades. A proper summary can be found in [\[5\]](#). This Section will summarize related work in the area of surface optimization and uncertainty visualization for surfaces.

2.1. Optimization of surface extraction

The original surface extraction algorithm, know as marching cube [\[6\]](#) is a well known state of the art algorithm. It is based on a selected isovalue that determines the resulting surface elements. Although, this algorithm has been successfully applied to many problems, the algorithm is not able to adapt its isovalue throughout the image to match the desired surface. Glaznig et al. [\[7\]](#) presented a marching cubes method, that is able to automatically adapt its isovalue throughout the extraction process. Although this improves the quality of the resulting isosurface, it holds higher potential for topological errors, than the classical marching cubes approach. Therefore, the presented methodology starts with a classic surface extraction algorithm and improves the resulting geometry through an uncertainty model.

In general, the marching cubes algorithm can lead to degenerated meshes. Approaches, that try to preserve the topology of thin structures [\[8\]](#), eliminate degenerated triangles [\[9\]](#) or insert additional points to preserve topological features [\[10\]](#) are available. These methods are proper improvements for the classic marching cubes approach and the presented methodology is able to include them in a straight forward manner. Instead of starting with a marching cubes approach it is possible to start from any surface and apply the optimization procedure presented in this manuscript.

Lopes et al. [\[11\]](#) presented an extension of the marching cubes algorithm, that was designed to improve the accuracy and robustness of the original algorithm. To achieve this, their method subdivides cells to identify key features of the resulting surface. Although this is a suitable approach to improve the quality of a surface extraction, the approach does not consider the uncertainty contained in many real world datasets. Therefore, the presented approach in this paper uses uncertainty measures to improve the surface generated by the marching cubes algorithms.

Athawale and Entezari [\[12\]](#) developed a method that is able to detect and quantify the effect of uncertainty throughout the computation of the marching cubes algorithm. They were able to propagate uncertainty measures described in the original image data in the interpolation during the surface extraction. Although this is a good starting point to estimate how uncertain input data affect the marching cubes algorithm, the authors did not propose a method to handle this information. Therefore, the presented approach that uses uncertainty measures in the original image to improve surfaces and make them more accurate and trustworthy.

2.2. Uncertainty visualization of surfaces

The communication of uncertainty is an important topic in many applications such as medical visualization and visualization is a key tool to achieve this goal. An overview of uncertainty visualization techniques can be found in [\[13\]](#). Although there exists a large variety of visualization techniques for data affected by uncertainty, the visualization of uncertainty of surfaces is subject of just a few works. The most important are discussed below.

Pöthkow and Hege presented a method [\[14\]](#), where isosurfaces are surrounded by heatmaps that indicate the probability of a surface to alter its position in space. Although this visualization can provide a good overview over the possible locations of surfaces, it can result in visual clutter. Therefore, the presented method uses a colorcoding of

the isosurface to indicate the remaining uncertainty after the optimization process without introducing further visual objects.

Drapikowski [\[15\]](#) described a model for isosurface uncertainty in medical applications based on geometric features such as smoothness and curvature. These features were mixed with knowledge about the underlying image structure and the human anatomy to determine the quality of the isosurface. Although this method outputs promising results for medical datasets it is dependent on suitable knowledge from the underlying object. In contrast to this, the presented method is able to quantify and visualize uncertainty independent from the underlying object and use this knowledge to optimize arbitrary geometries.

Rhodes et al. [\[16\]](#) evaluated different techniques to visually encode uncertainty on isosurfaces. They tested different modes as color-coding, textures and combination of these techniques for multi-modal visualization. They found that colorcoding is a suitable method to visualize uncertainty on a surface. Unfortunately, their work does not provide a model to describe the uncertainty of a surface resulting from image data. In the presented method, we use colorcoding to communicate the uncertainty that can be defined by using our model.

He et al. [\[17\]](#) presented an extension of the marching cubes algorithm, that utilizes an uncertainty model to quantify uncertainty in image data and transformed this information throughout the marching cubes algorithm. This leads to an uncertainty visualization on top of the extracted isosurface. Although this is a good starting point to introduce uncertainty information into the isosurface representation, the algorithm is not able to optimize the marching cubes extraction based on the given uncertainty model. Therefore, the presented approach utilizes an uncertainty model to optimize marching cubes results and visualize the remaining uncertainty.

The examination of the state of the art methods showed, that there is a need to develop a surface extraction algorithm that outputs accurate and trustworthy results while considering a proper uncertainty model.

3. Requirements for surface extraction

In order to develop a geometry extraction technique, that can be used in the decision making process of various applications, the following Section describes the requirements that need to be fulfilled to obtain such surfaces.

R1: Accuracy [\[18\]](#). In order to achieve a high user acceptance in real world applications, the extracted Surface needs to be as accurate as possible. In many cases, users are interested in exact sizes, shapes and positions of the extracted surfaces and therefore require accurate extraction methodologies.

R2: Reliability [\[19\]](#). Users from real world applications usually have no background knowledge of the underlying mathematical principle of the surface extraction algorithm. Instead, these algorithm are a black box for those users. Therefore, users from different domains need to be sure, that they can rely on the extracted surface during their daily tasks.

R3: Fast to compute [\[20\]](#). In real world applications, decisions often need to be made fast. Therefore, the computation of an isosurface is not allowed to take too long.

R4: Comparative [\[21\]](#). Often, different options need to be discussed during the decision making process. Therefore, isosurface options should be represented in an interactive system.

R5: Uncertainty-awareness [\[22\]](#). As the underlying image data of an isosurface extraction process is highly affected by uncertainty, a communication of this uncertainty throughout the isosurface extraction process is required. In addition, the final isosurface should include uncertainty information as well to allow users a better understanding of the trustworthiness of a considered isosurface.

4. Methods

In this manuscript we present a novel methodology to find a proper

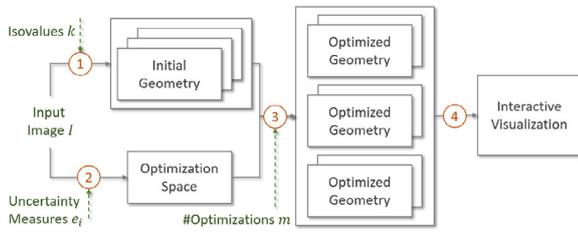


Fig. 1. Workflow of the presented methodology, consisting of the computation of the uncertainty model (1), initial geometry extraction (2), geometry optimization (3) and intuitive representation of optimization results (4).

geometric description for a depicted structure in an image. The goal is to obtain a geometry, that as accurate and reliable as possible. To achieve this, the presented method evaluates the input parameter space by extracting various geometries based on different isovalues and optimizing them with differently weighted uncertainty measures. The presented algorithm presents statistical information of all extracted geometries and allows the user to browse them in order to find his preferred result. An overview of the presented methodology can be found in Fig. 1.

The input of the presented methodology is a three-dimensional image, that can be defined as $\mathbb{I} := (\mathbb{V} \times \mathbb{X})^z$, where $\mathbb{V} := \{1, \dots, t_1\} \times \{1, \dots, t_2\} \times \{1, \dots, t_3\}$ where $t_k \in \mathbb{N}_{>0}$ and $\mathbb{X} := \{1, \dots, 256\}$. In other words, an input image is a three-dimensional image where each dimension has a maximum number of pixel and each pixel holds a value between 0 and 256. In the following manuscript $\mathbb{I}(v)$ refers to a value of a specific pixel.

The following sections will describe each step of the computational pipeline in detail starting from the input Image $\mathbb{I}(v)$.

4.1. Initial isosurface extraction

As shown in Section 2 isosurface extraction has been a subject of research since decades. The goal is to define a surface, that represents all values in an image containing a pre-defined isovalue k . In the presented method, we utilize an extracted surface provided by an arbitrary extraction algorithm such as marching cubes [23]. The choice of extraction algorithm can be based on the application and has no restrictions, besides that the algorithm needs to output a geometry containing points and triangles.

This extracted geometry \mathbb{G} can be defined as $\mathbb{G} := (\mathbb{P}, \mathbb{T})$, where $\mathbb{P} := \mathbb{R} \times \mathbb{R} \times \mathbb{R}$ is the point space and $\mathbb{T} := \{1, \dots, N\} \times \{1, \dots, N\} \times \{1, \dots, N\}$ is the triangle space, where triangles are constructed through three points.

Although a surface extraction outputs a first guess about the surface of an object, these class of algorithms are not considering the uncertainty of the underlying image data into account. One problem of these algorithms is based on the chosen isovalue required for surface extraction. In many cases, it becomes not directly clear which exact isovalue is a proper choice. This often results in a rerun of the algorithm, where users need to visually inspect their result to obtain a proper isovalue. In addition to that, isosurface extraction algorithms neglect the uncertainty of the underlying image data in two different manners. First, this information is not taken into account during the extraction process and second it is not assumed for representing the quality of the extracted surface.

Therefore, the presented methodology utilized uncertainty measures for image data and utilizes them to optimize the initial extraction of a geometry and visually encode the quality of the resulting geometry.

4.2. Uncertainty model

In order to obtain trustworthy and accurate geometry extraction the goal is to minimize the surface's uncertainty. This uncertainty origins from the input image data. Real world datasets are affected by

uncertainty that is introduced through the image reconstruction process. When applying a surface extraction algorithm to such data, the resulting surface is also affected by uncertainty. Usually, this uncertainty is not communicated, which can lead to a rejection of surface representations in applications.

Therefore, the goal is to quantify, minimize and communicate the uncertainty of an extracted surface. To achieve this, it is required to know the uncertainty of the underlying image data. In order to do so, the goal is to determine the uncertainty of each pixel contained in the original image. Contrary to the term error, uncertainty has not a clear definition. This is based on the fact, that the uncertainty of an image and its pixels cannot be determined based on a groundtruth as in the case of image errors. Instead, the uncertainty of an image pixel is an estimation based on a model.

The utilized model considered a pixel and its surrounding. Still, this does not result in a unique description as there exist a variety of model that estimate the uncertainty of a pixel. These model are build on different assumptions what pixel settings lead to a high amount of uncertainty. Therefore, an holistic view of uncertainty cannot be accomplished while using a single uncertainty measure. Instead, the model utilized in this paper is based on a collection of uncertainty measures that are selected to cover the important image quality aspects such as contrast, blur, noise, artifacts, and distortion [24].

A single uncertainty measure u is a function defined as:

$$u(\mathbb{I}, v) \rightarrow [0, 1] \quad (1)$$

In other words an uncertainty measure function is defined for a pixel of an image outputting a value between 0 and 1. If the uncertainty measure outputs 0, the input pixel is not affected by uncertainty according to the underlying uncertainty measure. In contrast to that, if the pixel is absolutely not trustworthy according to the used uncertainty measure, the output will be 1.

Throughout this paper the total number of utilized uncertainty measures is n whereas e_i is an uncertainty measure with $i \in \{1, \dots, n\}$.

In particular, the utilized uncertainty measures in the presented model are:

- **Acutance:** This measure assumes the image pixel to have a large gradient to be trustworthy [25].
- **Distance to original value:** The uncertainty outputted by this measure increases if the voxel value is altering from the chosen isovalue.
- **Gaussian error:** Estimates the gaussian noise assigned to an image pixel [26].
- **Local contrast:** Computes how homogeneous the surrounding of a pixel is [27].
- **Salt and pepper noise:** Estimates the salt and pepper noise assigned to an image pixel [28].
- **Brightness:** The brighter a pixel is, the lower the response of this uncertainty measure is [29].
- **Contrast correction:** Assuming the input image can be optimized by an contrast histogram stretch, this measure estimates the improvement for the performed stretch [30].

The output of the utilized measures is shown in Fig. 2. Pixels with a high uncertainty measure are indicated through white color whereas pixels with a low color are indicted through black color. It can be observed, that this measures can output very opposite results depending on the underlying assumption of uncertainty.

As the goal is to obtain an overall impression of the uncertainty contained in an image pixel, the mentioned uncertainty measures need to be combined. Therefore, the uncertainty of an image pixel is not a single scalar in our model. Instead, the uncertainty of a pixel can be defined as an vector containing all mentioned uncertainty measures as single components. This vector can be defined as $u(\mathbb{I}(v)) = (u_1, \dots, u_n)^T$.

In the following, this vector is utilized to optimize extracted

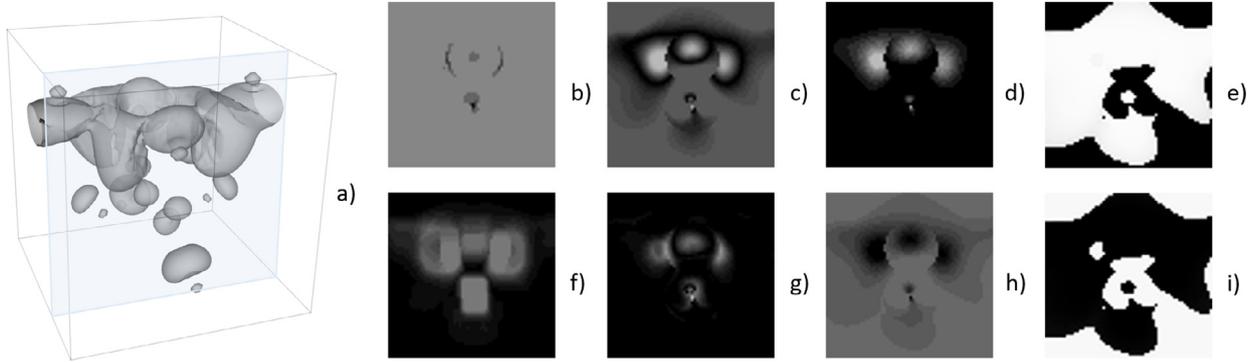


Fig. 2. Image errors output, based on the slice of a computed tomography scan showing two phases of a fluid a). The resulting error measures are b) Acutance, c) Distance to original Value, d) Gaussian Error, e) Local Contrast, f) Salt and Pepper Noise, g) Brightness and h) Contrast Correction.

geometries thus they result in certain geometries.

4.3. Geometry optimization

In order to find a surface extraction that represents the selected structure of the image as good as possible while holding a low uncertainty, the presented methodology computes a large variety of surfaces and optimizes them according to different parameter settings. The resulting surfaces are rated based on the remaining uncertainty of the surface and stored by this ranking. Pseudocode 1 shows the steps of the presented method.

The procedure can mainly be divided into three stages: First *Parameter Space Scanning*, where the high dimensional space is evaluated randomly, second *Optimization of Surfaces* where a surface will be optimized with each selected parameter third and *Evaluation of Surfaces* where each optimized surface will be evaluated based on the remaining surface error. The technical details will be described below.

```

1: function OPTIMZEDSURFACELIST( $I, K[], m, E, \epsilon$ )
2:   Initialize Results
3:   for  $k \in K$  do
4:      $G = \text{EXTRACT}(I, k)$ 
5:     Initialize Results[]
6:     for  $1, \dots, m$  do
7:       Initialize  $\omega[]$ 
8:       for  $e \in E$  do
9:          $\text{ADD}(\omega, \text{RANDOM}(0,1))$ 
10:         $\text{NORMALIZE}(\omega)$ 
11:         $G^* = \text{OPTIMIZE}(E, G, \omega, \epsilon)$ 
12:        Initialize  $f = \text{EVALUATE}(G^*)$ 
13:         $\text{ADD}(\text{Results}, (G^*, k, \omega, f))$ 
14:   return Results

```

Algorithm 1. Generation of surface optimizations with different parameters.

4.3.1. Scan parameter space

In order to extract a surface that represents the object of interest properly as well as being maximal trustworthy, the presented algorithm computes several surfaces starting with different isovalues and optimizes them based on the presented uncertainty measures. This is important as the choice of the isovalue is usually done manually by looking into the image and guessing a proper value. Although this guess is not completely wrong, there is no certainty, that the chosen value is the best choice. Therefore, the presented methodology utilizes the chosen isovalue and performs multiple surface extractions based on isovalues, that slightly alter from the chosen isovalue.

For each of those isovalues the goal is to optimize the resulting

surface thus the remaining surface is as trustworthy as possible. Instead of solely optimizing the surfaces thus the result minimizes the remaining uncertainty of the surface, the algorithm optimizes the surfaces based on differently strong weighted uncertainty measures. This is required, as different uncertainty measures make different assumptions how to quantify uncertainty. Depending on the input image these assumptions can be correct, incorrect or something in between.

To solve this problem, each uncertainty measure u_i obtains a weight ω_i , where $\omega_i \in [0, 1]$. The function of this weight is to control the importance of the according error measure during the optimization procedure. If $\omega_i = 0$, then the respective error measure is not considered for optimization. Instead, if $\omega_i = 1$, the respective error measure is strongly optimized.

All possible values of ω_i form a high-dimensional weight space. In this space, not all points have to be considered. In fact, only points, that are located on the hyper unit ball should be considered for optimization. This is caused to the fact, that all other value can be scaled to the surface of the hyperplane. In the optimization procedure scaling is an invariant and therefore, only the points on the hyper sphere are considered.

Obviously, the hyper sphere cannot be evaluated completely, as this would result in an infinite number of optimization runs. Instead, the user selects a number of optimization run m where for each run the weights are randomly generated between 0 and 1 and finally the entire vector of weights is normalized. Based on each of this normalized set of weights, the optimization procedure can be started.

4.3.2. Optimization of a surface

For each combination of an initial geometry extraction and random weight assignment, we present a method to optimize the initial guess of the surface, thus it becomes accurate and trustworthy. Therefore, the underlying uncertainty measures are considered.

As the initial geometry is based on the underlying input image, the geometry is located in the bounds of the volume. Therefore, it is possible to evaluate an uncertainty vector for each point in the initial geometry, referred to as $u(G(p))$. The resulting uncertainty vector is a tri-linear interpolated value based on the position of the point in the underlying image.

For the vector $u(G(p))$, its length $|u(G(p))|$ and its gradient $\nabla|u(G(p))|$ can be computed straightforward by considering the underlying uncertainty image and utilizing a tri-linear interpolation as well. This gradient is utilized to shift the points of the geometry into a direction thus the resulting length of the uncertainty vector becomes minimal.

Therefore, the points are shifted along the gradient until the shift of the point becomes smaller, then a user selected threshold. The algorithm iterates over all points contained in the geometry until each point cannot be shifted more then the set threshold. While doing this, the points of the geometry change their position in space such that the resulting geometry is located at points in space, that have a low

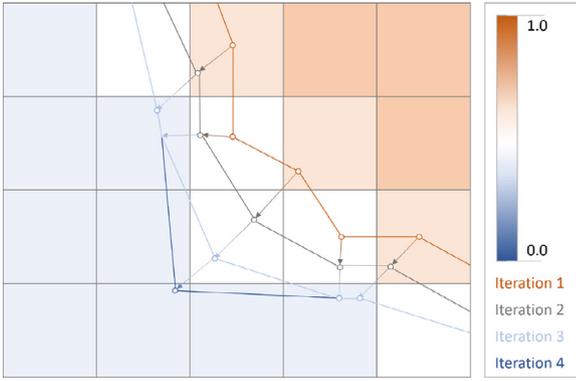


Fig. 3. Scheme of geometry optimization based on the underlying uncertainty measures. The points of the geometry are shifted along the uncertainty gradient until the position change does not exceed a user defined threshold.

uncertainty. Therefore, the resulting surface optimizes the surface uncertainty based on the underlying uncertainty model.

Fig. 3 shows the geometry optimization in detail. All points of the original geometry (red) are shifted along the uncertainty gradient. In the given example, this shift is performed twice. In the third iteration, solely 3 points are shifted, as their uncertainty gradient is still bigger, then the user-defined threshold. After the algorithm is performed, the points of the geometry are located at a point in space, where the uncertainty becomes a local minimum.

The benefit of this approach is that there is no need to insert new points. This allows an improvement of the surface generated from an image in low computational effort and without changing the topology of the underlying geometry.

4.3.3. Evaluation of surface

Depending on the selected parameter space, the presented algorithm outputs an optimized geometry where each geometric point is shifted thus it remains in a local minimum of surface uncertainty close to the starting point. As the presented algorithm outputs a set of optimized geometries based on the pre-defined parameter space, we require a mechanism that rates the quality of the optimized surfaces. Therefore, we utilize the remaining surface uncertainty to determine the quality of the resulting geometry. The uncertainty of a geometric surface $U(\mathbb{G})$ can be computed by:

$$U(\mathbb{G}) = \frac{\sum_{T \in \mathbb{G}} S(\mathbb{G}(T)) * U(\mathbb{G}(T))}{\sum_{T \in \mathbb{G}} U(\mathbb{G}(T))} \quad (2)$$

whereas, $S(\mathbb{G}(T))$ is the surface area of a triangle and $U(\mathbb{G}(T))$ is the

average length of the three uncertainty vectors assigned to the triangle points. The surface evaluation function takes the size of the surface into account. This ensures, that the remaining surface uncertainty is normalized based on the size of the considered geometry. Without doing this, it would be possible that small surfaces are rated better although they have a higher average uncertainty than a larger surface.

The proposed evaluation measure can be utilized to sort all optimized geometries. The result is a list of geometries sorted by their remaining surface uncertainty $U(\mathbb{G})$. Each element of this list contains the following items:

- (G) , the resulting geometry after the optimization process
- k , the utilized isovalue (further parameters, when using another surface extraction algorithm)
- ω_b , the weights for each uncertainty measure utilized for the optimization process
- ϵ , the stop criteria

The average time to compute such a list is highly depending on the runtime of the underlying surface extraction algorithm (which is usually depending on the number of pixels z , the number of tested isovalues k , the number of different weight assignments m and the required iterations to optimize a geometry i . This results in a runtime of $O(zkmi)$.

This list is utilized to show the user trustworthy geometries and allow him to explore the parameter space as well as the optimized geometries.

4.4. Visualization of optimization results

The output list of the geometry optimizations covers a large number of optimized geometries and their resulting surface uncertainty. Although, the geometry with the smallest surface error is the most interesting geometry, the remaining list can give important insight to the parameter space and its properties. In addition to that, the average surface uncertainty gives an overall impression for the quality of a surfaces. Often users are interested in specific parts of a surface and their uncertainty values. Therefore, the list of optimized geometry is embedded into an intuitive visual system, that allows the user to explore the list of optimized surfaces and inspect selected surfaces in detail.

Therefore, the visual system contains two linked views: a parallel coordinate view, showing for each isovalue a parallel coordinate plot containing the chosen random weights and the resulting surface uncertainty and a 3D plot of the user selected surfaces. The views and an example result can be reviewed in Fig. 4.

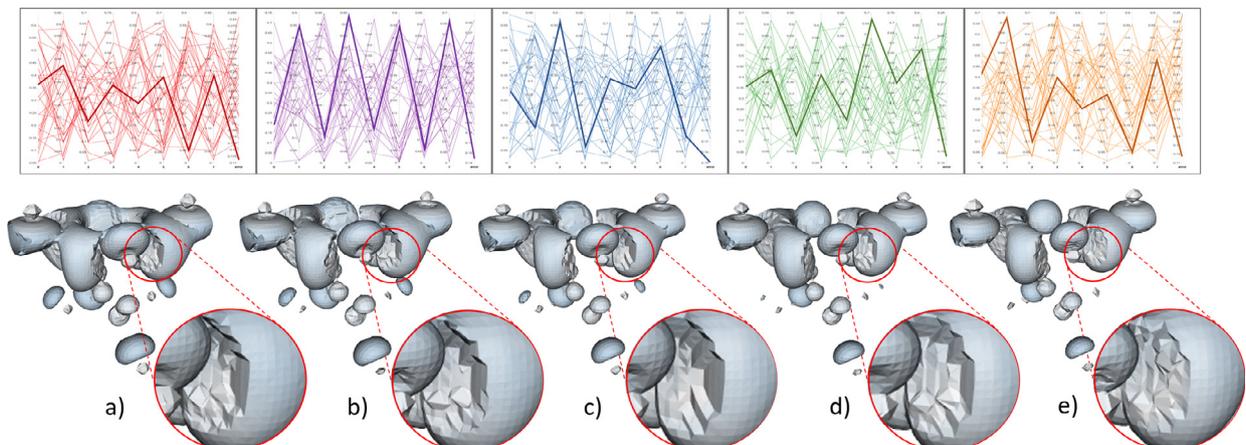


Fig. 4. Visualization of resulting surfaces. Top: Selection Widget containing parallel coordinates. Middle: Selected Surfaces that can be reviewed further. Down: Closeup for comparison of algorithm outputs.

The parallel coordinate view offers different parallel coordinate plots, one for each isovalue in the parameter space. Each of these plots contains all weights for the surface optimization as well as the resulting surface uncertainty. Users can interactively select surfaces based on their parameter settings. Based on the user defined selection, the surface view shows the 3D plots of the resulting surfaces. The surfaces are aligned under the parallel coordinate plot of the respective starting isovalue for an intuitive analysis.

To allow users to analyze the resulting surface in detail, the remaining surface uncertainty per triangle is color coded on the surface. The color ranges from blue ($U(G(T)) = 0$) over white ($U(G(T)) = 0.5$) to red ($U(G(T)) = 1$). With this color coding the user is enabled to differentiate the quality of a surface depending on the remaining surface uncertainty at specific locations. As red is a signal color the user is directly guided to uncertain areas on the surface.

Fig. 4 shows the user interface of the presented system. The top row shows the parallel coordinate view with 5 different isovalues, the utilized weights for optimization (30 for each isovalue) and the resulting surface uncertainty. For each isovalue (80, 90, 100, 110 and 120) the user selected the best optimization result from the performed optimizations. They are indicated by a thicker line. In the middle view, the selected surface visualizations can be reviewed. In this example, the isovalues are very distinct, leading to highly different surfaces, as shown in the lower part of Fig. 4. Overall, the surface selected in the orange parallel coordinate view has the lowest remaining surface error. In the closeup it can be observed, that in this optimization run, the algorithm outputs the best surface for the concave area of the geometry. This can be easily detected with the proposed visual system.

Based on the presented workflow, users are enabled to extract accurate and trustworthy surfaces from image data and review them.

5. Results

In the following section the presented approach is used to create extract accurate and reliable surfaces from real world datasets in the medical and mechanical engineering area. The presented approach was implemented using C++ with the vtk [31,32] and Qt [33] libraries.

5.1. Sphere example

Fig. 5 shows the extraction of a sphere surface for a synthetic dataset. The dataset had a size of $70 \times 70 \times 70$. The data was generated by setting all voxels within the radius of a sphere with the grayscale of 100. On all other locations, the grayscale value was set to 0. We ran 30 different optimizations with 5 different isovalues. The parallel coordinate view of all optimization runs can be found in Fig. 5. The best optimization result has an average surface uncertainty of 0.11 with an isovalue of 52.

Fig. 5(b) shows the isosurface extraction that is outputted by the marching cubes algorithm starting with an isovalue of 52. The closeup shows an often occurring effect when using marching cubes: the staircase effect. In this effect, the resulting surface builds several levels instead of a smooth curved surface. In many applications, this output needs to be post-processed to remove this effect.

Based on the marching cube output, the presented methodology is able to adjust the surface points, thus the surface becomes more accurate. Fig. 5(c) shows that the staircase effect is removed when using the presented methodology. Overall, the optimized surface is smoother and closer to the original sphere shape, then the result based on the marching cubes algorithm.

This result shows, that the underlying model of uncertainty used for the optimization process is able to optimize the geometry outputted by surface extraction algorithms. The interface is easy to use so that the user can identify the best optimization result very easy and explore the resulting surfaces. The inspection of selected surfaces is intuitive, as all zooming and panning operations are consequently propagated to all visible geometries.

5.2. Engine example

The next example is the surface extraction of an engine from a Computed Tomography scan. In mechanical engineering, these datasets are generated to automatically detect defects of working pieces after they are manufactured. Therefore, it is very important to obtain an accurate and reliable surface of the manufactured object to estimate its quality.

The original image data has a size of $256 \times 256 \times 256$ and is a freely available dataset [34]. The upper row shows the resulting surface extraction from a standard marching cubes algorithm, whereas the lower row shows the optimization results of the presented methodology. We tried five different isovalues (78–82) and ran 100 optimizations per isovalue. The shown result in the Fig. 6 had the lowest resulting average surface error with 0.241108.

Fig. 6 compares the output of the marching cubes algorithm (top) with the output of the presented algorithm (down) in this paper for the dataset containing the engine. Part (a) and (b) show the closeup of a hole in the engine's side. It can be observed, that the output of the original marching cubes algorithm is edged, which is an effect of the linear interpolation occurring in the original algorithm. On the other hand, our algorithm results in a smooth round hole shape. Part (c) and (d) show a closeup of a round recess on top of the engine. Here, the effect of smoothing out the results from the original marching cubes algorithm become clearer. The entire circular shape is captured more accurately in the results of the presented methodology.

Part (e) and (f) show the closeup of a tube accessing the interior of the engine. The tube is a thin structure, which is usually hard to extract by a classic marching cubes approach. As the results show, the tube looks edged and very thick. Contrary to this, the presented approach is able to smooth and thin the surface representation of the tube. The resulting tube is almost round, although no new points are introduced into the geometry.

The results show, that round and thin structures can be reproduced more accurately while using the presented approach of the paper in comparison to the classical marching cubes approach. This can help users from the mechanical engineering area to inspect work pieces and compare them to the targeted tolerances.

5.3. Foot example

The third example is a Computed Tomography Scan of a human foot. The original dataset has a size of $256 \times 256 \times 256$ pixels and is freely available [34]. It shows an entire foot with all soft tissues and all single bones of the foot skeleton. In medical image processing surfaces are an important technique to estimate the size of organs or tumors or locations and conditions of structures such as the shown bones in the examined dataset.

We performed 5 different isovalues (88–92) and tried 50 different weights per isovalue which is a total of 250 optimizations. Each optimization procedure took about 20 s on a normal desktop computer (Intel Core i7, 2.6 GHz). In total the program took about an hour to compute all optimizations. This time consumption can be minimized straightforward, as our methodology can be run in multiple threads very easily. Each optimization is independently from each other and can theoretically be run on its own core.

Fig. 7 shows the resulting surfaces from the marching cubes algorithm (left) and the presented approach of this paper (right). Closeup (a) and (d) show a closeup of the big toes bone. In the original surface extraction algorithm, the geometry contained several staircases, that are not correct. In the optimized geometry, the staircases are smoothed out and therefore, the overall uncertainty is minimized.

Closeup (b) shows staircases as well as a feature that points out of the bone. The goal was to minimize the staircase structure while preserving the feature. Closeup (e) shows, that our algorithm was able to smooth the overall appearance of the bone, while preserving the

feature. This is very important especially in medical applications, as anomalies in the human body need to be preserved so that the medical doctor can identify and examine them.

Closeup (c) and (f) show how the original marching cubes algorithm underestimated the size of a bone. Using the presented approach of this paper resolves this issue and leads to an overall smoother impression of the bone.

6. Discussion

In order to discuss the presented approach, we performed a user study and examine if the defined requirements are fulfilled.

6.1. User evaluation

As an important goal of the presented approach is to gain a higher user acceptance, we conducted a user evaluation of the presented results. The goal was to identify if users would preference the geometries generated by the presented approach. Therefore, we showed geometries extracted from a standard marching cubes algorithm in comparison to a geometry extracted by the presented approach. The evaluation was blinded, so the users did not know which geometry was generated from which approach. Two examples were shown: the sphere example, shown in Fig. 5 and the foot example as shown in Fig. 7. We removed the closeups that are visible in the Figures to obtain user feedback that is not influenced. For the sphere example we asked the users to select the geometry that appears to be closer to a perfect sphere and for the foot example we asked to identify the geometry that expresses the geometry of bones better. The results can be found in Fig. 8. Bar chart 1 shows the number of users that voted for the geometry of the presented approach and 2 shows the number of users that voted for the classic isosurface extraction approach.

20 Persons participated in the study, 14 male and 6 female. Each of the participants works in the field of computer science. Each of them declared, that they are used to the concept of isosurfaces. The results show, that a majority of the users find that the extracted geometry of the presented approach better match the original object. In addition, we presented our methodology to a domain expert and he gave us a very positive feedback.

The most promising are:

- The method significantly enhances the exploration of a high dimensional uncertainty space
- The projection of errors onto geometries supports a reliable decision making process

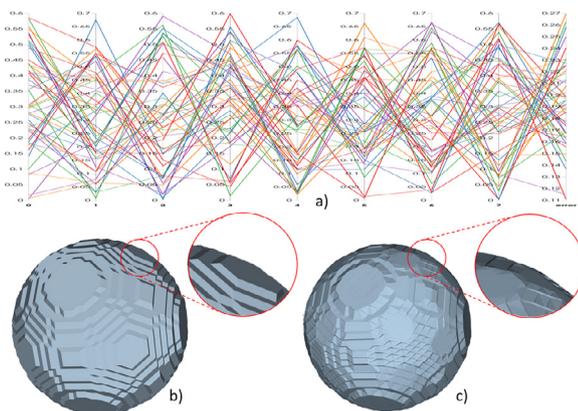


Fig. 5. Isosurface extraction of a three dimensional image containing the approximation of a sphere. a) Weights and isosurfaces (color-coded) for the run experiments. b) Surface extracted by a marching cubes algorithm. The closeup shows the staircase effect. c) Surface extracted with the presented method. The closeup shows that the staircase effect can be minimized.

- I was able to enhance my geometric representation with the presented method

6.2. Discussion of defined requirements

In the following we aim to discuss how the presented methods targets the requirements defined in Section 3.

Accuracy. As shown in the user evaluation, the majority of users think, that our method is able to represent the geometry of objects more accurate. This impression is enhanced by the given results. Nevertheless, the surface extraction of the presented method strongly depends on the starting isosurface generation. The utilized fundamental marching cubes algorithm used in the presented scenarios can output topological errors. The presented method is not restricted to the utilized isosurface extraction algorithm. In fact, an arbitrary surface extraction algorithm can be utilized. Newman and Yi presented a survey on marching cubes algorithms [35]. Depending on the purpose, the extracted surface needs to fulfill, improved versions of the marching cubes algorithm can be utilized.

Another possible effect is, that the geometry correction leads to points, that are shifted to the same location in space. Although in all tested cases never occurring, there is no guarantee, that the resulting surface after the optimization process does not contain self-intersections. Fortunately, there exist a large variety of geometry cleanup procedures, that can be applied to the presented optimization approach, when required [36,37].

In summary, the presented optimization methodology needs to be combined with a suitable geometry extraction as well as geometry cleaning approach. Depending on the purpose of the extracted surface, Table 1 gives suggestions on the algorithms that can be utilized.

Reliability. The optimization of isosurfaces is based on uncertainty-measures. The resulting surface is located at points in space where the uncertainty is minimal (locally). Therefore, the presented methodology outputs surfaces that are more reliable in comparison to the marching cubes algorithm.

Fast to compute. As shown in Section 4, the complexity of the presented algorithm is $O(zkmi)$. A surface extraction including the optimization procedure took about 20–30 on a normal Laptop (Interl(R) Core(TM) i7-6700HQ CPU, 2.60 GHz), depending on the underlying dataset. Considering that users aim to tests thousands of parameter configurations, this leaves room for improvement.

Fortunately, this can be solved by using a faster implementation of the marching cubes algorithm using octrees [35]. This would decrease the runtime to $O(\log(z)kmi)$. In addition, the presented approach can be parallelised in multiple ways. First, the different optimization runs can be separated to different threads. Second, the optimization itself can be divided into different threads as well as each geometry points is optimized separately.

Comparative. The presented approach includes a visual framework that allows to compare different optimization option in one view. As the users stated, this helped them to understand how the underlying uncertainty space effects the resulting geometry. In addition, the presented rating functions helps users to identify the best options and quantify differences between geometries.

Uncertainty-awareness. The main goal of the presented approach was to achieve isosurface extraction that come with a high uncertainty awareness. This is accomplished in multiple ways. First, the space that is used for optimization is high-dimensional targeting to cover as much aspects of uncertainty as possible. Second, the presented approach tests a large variety of uncertainty-space weightings to see

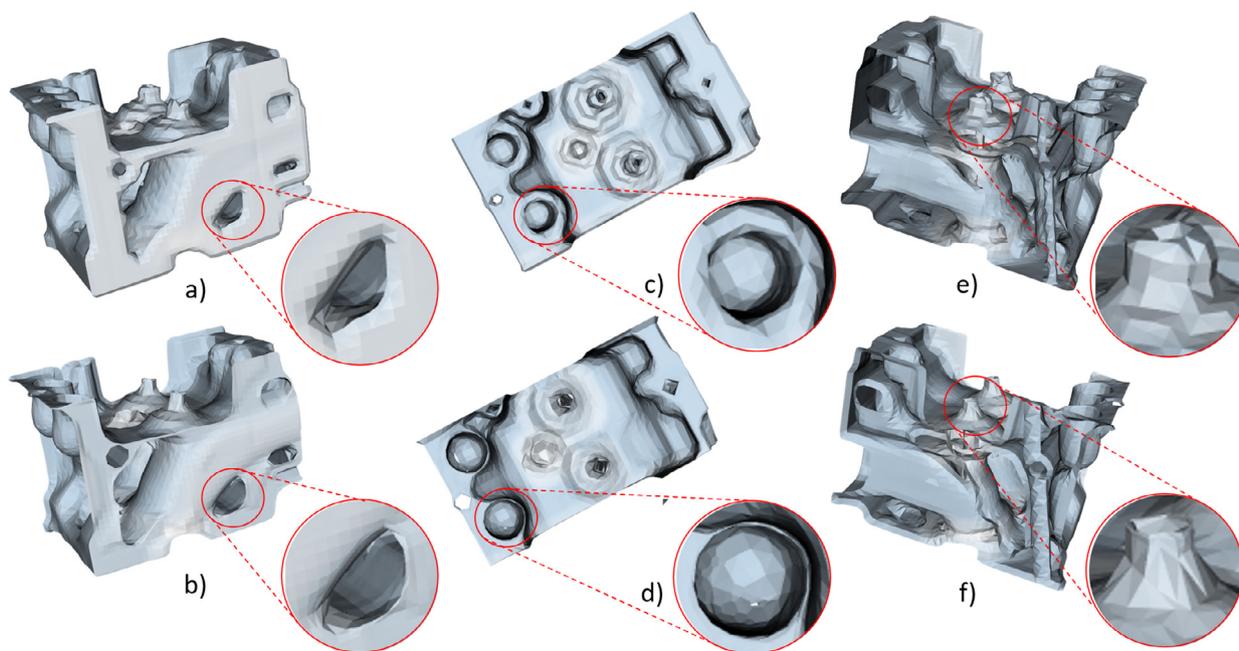


Fig. 6. Isosurface extraction of a engine from a CT scan. Upper row: results from a classic marching cubes algorithm. a, b) closeup of hole in engine border. c, d) Closeup of round recess on top of the engine. e, f) closeup of tube. For all cases our algorithm was able to enhance the surface representation.

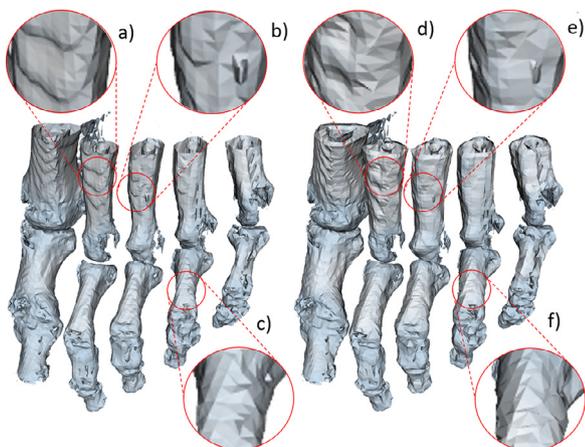


Fig. 7. Surface extraction of human foot with a classic marching cubes approach (left) and the presented approach in this paper (right). Closeups show how the presented approach is able to improve the surface while preserving important features.

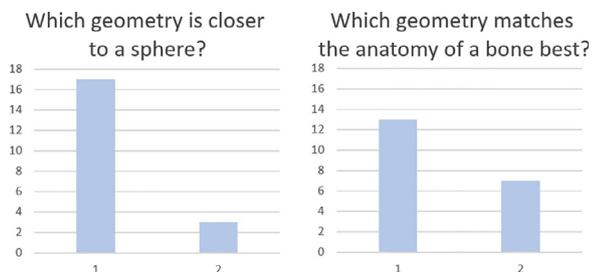


Fig. 8. User evaluation results of the geometries shown in Fig. 5 (left) and Fig. 7 (right). Bar chart 1 shows the number of users that voted for the geometry of the presented approach and 2 shows the number of users that voted for the classic isosurface extraction approach.

which uncertainty metrics has the best influence on the optimization procedure. Last, the surface representation is colorcoded to indicate users which areas on the surface are more trustworthy than others.

Table 1

Possible problems before and after the optimization procedure and their solutions.

Problem	Possible solution
Degenerated triangles	Li and Agathoklis [38]
Ambiguity in MC	Gelder and Wilhelms [39]
Internal ambiguity	Nilson and Hamann [40]
Isolated vertices	Can be ignored
Dangling edges	OFF and PLY File Format
Orientation	Attene and Falcidieno [41]
Holes and gaps	Barequet et al. [42]
Self-Intersections	Hachenberger et al. [43]

In summary, the presented method forms a novel surface extraction method that helps minimizing the surfaces uncertainty.

7. Conclusions

We presented a novel surface extraction methodology that is able to extract accurate and reliable surfaces by using an uncertainty model of the underlying image data. The algorithm is able to run multiple optimization procedures with different input parameters and evaluates them based on the remaining surface uncertainty. To allow a fast and intuitive examination of the optimization results, the presented methodology includes a visual system to explore and examine the resulting extracted surfaces. We showed the effectiveness of this approach by extracting surfaces from artificial as well as real world datasets originating from different domains.

As future work, the goal is to include the presented methodology in real world workflows as occurring in clinical daily routine. In addition, the goal is to provide users with a parallel computation structure to avoid long runtimes.

Acknowledgments

This research was funded by the German Research Foundation (DFG) within the IRTG 2057 “Physical Modeling for Virtual Manufacturing Systems and Processes”.

References

- [1] J.C. Russ, *Image Processing Handbook*, 4th ed., CRC Press, Inc., 2002.
- [2] I.N. Bankman (Ed.), *Handbook of Medical Image Processing and Analysis*, Academic Press, 2009.
- [3] M. Bertram, *Multiresolution Modeling for Scientific Visualization*, (2000) Ph.D. thesis. AA10802446.
- [4] Front matter, in: C.D. Hansen, C.R. Johnson (Eds.), *Visualization Handbook*, Butterworth-Heinemann, Burlington, 2005, iii- doi:10.1016/B978-0-12-387582-2.50051-4.
- [5] T.S. Newman, H. Yi, A survey of the marching cubes algorithm. *Comput. Graph.* 30 (5) (2006) 854–879.
- [6] W.E. Lorensen, H.E. Cline, Marching cubes: A high resolution 3d surface construction algorithm, *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87*, ACM, 1987, pp. 163–169, <https://doi.org/10.1145/37401.37422>.
- [7] M. Glanzig, M.M. Malik, M.E. Gröller, Locally adaptive marching cubes through isovalue variation, in: V. Skala (Ed.), *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2009, pp. 33–40.
- [8] S. Schaefer, J. Warren, Dual marching cubes: primal contouring of dual grids, *Comput. Graphics Forum* (2005), <https://doi.org/10.1111/j.1467-8659.2005.00843.x>.
- [9] V.S. Lempitsky, Y. Boykov, Global optimization for shape fitting, 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18–23 June 2007, Minneapolis, Minnesota, USA, (2007), <https://doi.org/10.1109/CVPR.2007.383293>.
- [10] C.A. Dietrich, J.L. Comba, C.E. Scheidegger, J. Schreiner, L.P. Nedel, C.T. Silva, Edge transformations for improving mesh quality of marching cubes, *IEEE Trans. Vis. Comput. Graph* 15 (2008) 150–159, <https://doi.org/10.1109/TVCG.2008.60>.
- [11] A. Lopes, K. Brodlie, Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing, *IEEE Trans. Vis. Comput. Graph* 9 (1) (2003) 16–29, <https://doi.org/10.1109/TVCG.2003.1175094>.
- [12] T. Athawale, A. Entezari, Uncertainty quantification in linear interpolation for isosurface extraction, *IEEE Trans. Vis. Comput. Graph* 19 (12) (2013) 2723–2732, <https://doi.org/10.1109/TVCG.2013.208>.
- [13] G.-P. Bonneau, H.-C. Hege, C.R. Johnson, M.M. Oliveira, K. Potter, P. Rheingans, T. Schultz, *Overview and State-of-the-Art of Uncertainty Visualization*, Springer London, pp. 3–27. doi:10.1007/978-1-4471-6497-5_1.
- [14] T. Athawale, E. Sakhaee, A. Entezari, Isosurface visualization of data with non-parametric models for uncertainty, *IEEE Trans. Vis. Comput. Graph* 22 (1) (2016) 777–786, <https://doi.org/10.1109/TVCG.2015.2467958>.
- [15] P. Drapikowski, Surface modeling uncertainty estimation and visualization, volume 32, (2008), pp. 134–139, <https://doi.org/10.1016/j.compmedimag.2007.10.006>.
- [16] P.J. Rhodes, R.S. Laramée, R.D. Bergeron, T.M. Sparr, Uncertainty visualization methods in isosurface volume rendering, *Eurographics 2003, Short Papers*, (2003), pp. 83–88.
- [17] Y. He, M. Mirzargar, S. Hudson, R.M. Kirby, R.T. Whitaker, An uncertainty visualization technique using possibility theory: possibilistic marching cubes, *Int. J. Uncertain. Quantif.* 5 (5) (2015) 433–451.
- [18] A. Townsend, L. Pagani, L. Blunt, P.J. Scott, X. Jiang, Factors affecting the accuracy of areal surface texture data extraction from x-ray ct, *CIRP Ann.* 66 (1) (2017) 547–550, <https://doi.org/10.1016/j.cirp.2017.04.074>.
- [19] H. Masuda, I. Tanaka, M. Enomoto, Reliable surface extraction from point-clouds using scanner-dependent parameters, *Comput. Aided Des. Appl.* 10 (2) (2013) 265–277, <https://doi.org/10.3722/cadaps.2013.265-277>.
- [20] C.L. Bajaj, V. Pascucci, D.R. Schikore, Fast isocontouring for improved interactivity, *Proceedings of the 1996 Symposium on Volume Visualization, VVS '96*, IEEE Press, Piscataway, NJ, USA, 1996, pp. 39–ff <http://dl.acm.org/citation.cfm?id=236226.236231>.
- [21] A. Brambilla, P. Angelelli, Andreassen, H. Hauser, Comparative visualization of multiple time surfaces by planar surface reformation, 2016 IEEE Pacific Visualization Symposium (PacificVis), (2016), pp. 88–95, <https://doi.org/10.1109/PACIFICVIS.2016.7465255>.
- [22] C.D. Correa, Y.H. Chan, K.L. Ma, A framework for uncertainty-aware visual analytics, 2009 IEEE Symposium on Visual Analytics Science and Technology, (2009), pp. 51–58, <https://doi.org/10.1109/VAST.2009.5332611>.
- [23] W.E. Lorensen, H.E. Cline, Marching cubes: A high resolution 3d surface construction algorithm, *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87*, ACM, 1987, pp. 163–169, <https://doi.org/10.1145/37401.37422>.
- [24] A. George, S.J. Livingston, A survey on full reference image quality assessment, *Int. J. Res. Eng. Technol.* (2013) 303–307.
- [25] J.R. Fienup, Invariant error metrics for image reconstruction, *Appl. Opt.* 36 (32) (1997) 8352–8357.
- [26] R.K. Mantiuk, A. Tomaszewska, R. Mantiuk, Comparison of four subjective methods for image quality assessment, *Comput. Graph. Forum* 31 (8) (2012) 2478–2491.
- [27] Q. Tian, Q. Xue, N. Sebe, T. Huang, Error metric analysis and its applications, *Proc. SPIE - Int. Soc. Opt. Eng.* 5601 (2004) 46–57.
- [28] I. Irum, M. Sharif, M. Raza, S. Mohsin, A nonlinear hybrid filter for salt and pepper noise removal from color images, *J. Appl. Res. Technol.* 13 (1) (2015) 79–85.
- [29] K.H. Thung, P. Raveendran, A survey of image quality measures, 2009 International Conference for Technical Postgraduates (TECHPOS), (2009), pp. 1–4.
- [30] J.V. L. R. Gopikakumari, Article: iem: a new image enhancement metric for contrast and sharpness measurements, *Int. J. Comput. Appl.* 79 (9) (2013) 1–9. Full text available.
- [31] W. Schroeder, K. Martin, B. Lorensen, *The visualization toolkit—An object-Oriented Approach to 3D graphics*, fourth ed., Kitware, Inc., 2006.
- [32] H.J. Johnson, M. McCormick, L. Ibanez, *The itk Software Guide*, Kitware, Inc., 2013.
- [33] D. Molkentin, *The book of Qt 4: The Art of Building Qt Applications*, Open Source Press San Francisco, Munich, 2007. Index. <http://opac.inria.fr/record=b1134099>.
- [34] The volume library, (<http://www.informatik.uni-erlangen.de/External/vollib/>) Accessed: 2016-03-09.
- [35] T.S. Newman, H. Yi, A survey of the marching cubes algorithm. *Comput. Graphics* 30 (5) (2006) 854–879 <http://dblp.uni-trier.de/db/journals/cg/cg30.html#NewmanY06>.
- [36] Front Matter, Published By Imperial College Press And Distributed By World Scientific Publishing CO., pp. i–xvi. doi:10.1142/9781860946813_fmatter.
- [37] M. Attene, M. Campen, L. Kobbelt, Polygon mesh repairing: an application perspective, *ACM Comput. Surv.* 45 (2) (2013) 15:1–15:33, <https://doi.org/10.1145/2431211.2431214>.
- [38] J. Li, P. Agathoklis, A case study of isosurface generation in 3d visualization, *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, volume 2, (1993), pp. 622–625.
- [39] A. van Gelder, J. Wilhelms, Topological considerations in isosurface generation, *ACM Trans. Graph.* 13 (4) (1994) 337–375, <https://doi.org/10.1145/195826.195828>.
- [40] N. G. H. B., The asymptotic decider: resolving the ambiguity in marching cubes, *Visualization* (1991) p. 83–91.
- [41] M. Attene, B. Falcidieno, Remesh: An interactive environment to edit and repair triangle meshes, *IEEE International Conference on Shape Modeling and Applications 2006 (SM'06)*, (2006), 41–41.
- [42] G. Barequet, S. Kumar, C.A. Duncan, Rsvp: a geometric toolkit for controlled repair of solid models, *IEEE Trans. Visual. Comput. Graphics* 4 (1998) 162–177, <https://doi.org/10.1109/2945.694983>.
- [43] P. Hachenberger, L. Kettner, K. Mehlhorn, Boolean operations on 3d selective nef complexes: data structure, algorithms, optimized implementation and experiments, *Comput. Geom.* 38 (1) (2007) 64–99. Special Issue on CGAL. doi:10.1016/j.comgeo.2006.11.009.