rijksuniversiteit
groningen

University of Groningen

Salient edge preserving noise removal using surface skeletons

Master Thesis

Author:
H. R. Schubert

Supervisors:
Prof. dr. A. C. Telea
Prof. dr. N. Petkov

September 7, 2016

Abstract

Medial descriptors (surface and curve skeletons) have been used for various operations such as shape retrieval, shape processing, and shape manipulation. They offer efficient and effective ways to eliminate small-scale noise details from discrete datasets, which are robust and in some ways work better than curvature-based methods. Noise-removal using skeletons works by removing points which correspond to small-scale boundary details (which are generally caused by noise) by trimming the skeleton branches, and then reconstructing the shape from the simplified skeleton. A method has been proposed in 2D which removes skeleton branches that correspond only to small-scale noise details. It allows eliminating noise details, but keeps sharp (salient) features of the shape untouched. However, it does not directly generalize to 3D for complex topologies. In this thesis this method has been adapted and improved for 3D models, and can be successfully applied for a wide range of 3D shapes. The resulting pipeline allows quick and robust removal of noise, and, compared to a selection of tested smoothing methods, it removes more noise without distorting important features.

Contents

1 Introduction 7

2 Related work 10
- 2.1 Mathematical morphology . . . 10
  - 2.1.1 Connectivity . . . 10
- 2.2 Skeletons . . . 11
  - 2.2.1 CMD skeletons . . . 12
  - 2.2.2 RMA skeletons . . . 12
  - 2.2.3 Discrete case . . . 12
  - 2.2.4 Reconstruction . . . 13
  - 2.2.5 Surface skeletons . . . 13
- 2.3 Skeleton computation . . . 14
- 2.4 Skeleton regularization . . . 16
  - 2.4.1 Importance thresholding . . . 16
  - 2.4.2 Branch pruning . . . 17
- 2.5 Other denoising methods . . . 18

3 Saliency pipeline 20
- 3.1 Saliency metric . . . 20
  - 3.1.1 Pipeline . . . 21
  - 3.1.2 Choice of importance measure . . . 22
  - 3.1.3 Relation with mathematical morphology . . . 24
- 3.2 3D extension . . . 26

4 Derivative method 27
- 4.1 Corner and edge sensitivity . . . 27
- 4.2 Derivative metric . . . 27
  - 4.2.1 Choice of velocity field . . . 31
  - 4.2.2 Inverse mapping . . . 33
- 4.3 Velocity field experiment . . . 33
- 4.4 Qualitative comparison . . . 36
  - 4.4.1 Setup . . . 36
  - 4.4.2 Comparison . . . 37
- 4.5 Conclusion . . . 38

5 Saliency post-processing 42
- 5.1 Multi-axial features . . . 42
- 5.2 Solutions . . . 43
  - 5.2.1 Direct streamline filtering . . . 43
  - 5.2.2 Global streamline filtering . . . 45
- 5.3 Comparison . . . 46

5.3.1 Line experiment . . . 47
5.3.2 Conclusion . . . 48

6 EDT post-processing 51
6.1 Problem . . . 51
6.1.1 Skeleton smoothing versus EDT filtering . . . 51
6.2 Methods . . . 53
6.2.1 Kernel-based filtering methods . . . 53
6.2.2 Least squares projection . . . 54
6.2.3 Flat Projection . . . 55
6.3 Evaluation . . . 56
6.3.1 Implicit versus explicit kernel methods . . . 57
6.3.2 Comparison . . . 57
6.3.3 Conclusion . . . 58

7 Implementation 63
7.1 Skeleton extraction methods . . . 63
7.1.1 IMA skeleton . . . 63
7.1.2 Unified Skeleton Method . . . 65
7.1.3 Evaluation of skeleton methods . . . 67
7.2 Noise model . . . 69
7.3 Estimating geodesic distances . . . 71
7.3.1 Estimation . . . 71
7.3.2 Conclusion . . . 73
7.4 Reconstruction . . . 74
7.4.1 Reverse Euclidean distance transform . . . 74
7.4.2 Comparison . . . 75
7.5 Conclusion . . . 75

8 Results 77
8.1 Complete pipeline . . . 77
8.1.1 Adapted pipeline . . . 77
8.1.2 Results . . . 78
8.2 Comparison . . . 85
8.2.1 Results . . . 85
8.3 Conclusion . . . 86

9 Concluding remarks 90
9.1 Future work . . . 91

List of Figures

1.1 Example of naive smoothing using a Gaussian kernel, while the noise is reduced significantly, important features are blurred as well[61]. . 7

2.1 Illustration of the notation A_k^n based on contact of a curve with a circle [23]. 13
2.2 Illustration of the surface skeleton of an elliptical cylinder [21]: (a) the cylinder; (b) the cylinder and its surface skeleton; (c) the top and bottom sheets of the skeleton; (d) the central sheet which connects the bottom and top sheets; (e) labeled contact points of the central sheet. 15
2.3 An example of a surface skeleton of a horse model. The rainbow colored map shows the simplification metric of [29]. 15
2.4 This figure (from [57]) illustrates the various local characteristics of a point P which can be used as the basis for a pruning metric of the skeleton. Examples are the angle θ of the feature points P^+ and P^-, the chord distance (length of circular arc), the geodesic distance (distance over the boundary), or the area A. 17

3.1 An illustration which shows the difference between corners and cusps: in (a) both the importance ρ and the distance D steadily increases on a branch, while in (b) D increases, but ρ stays constant. 21
3.2 The saliency pipeline for 2D images [58]. In the first step the cusps are removed by removing the ligature branches of the foreground skeleton, which is then reconstructed. Respectively, the dents of reconstructed shape are removed by trimming ligature branches of the background skeleton. The reconstruction of the of the background skeleton gives the final result where both cusps and dents are removed. 23
3.3 An example showing why the operator δ is not increasing. The bump of A is removed as it corresponds to a ligature branch, however the rectangle B contained in A does not contain ligature branches, and is therefore kept intact. 24
3.4 A demonstration of the non-idempotent behavior of the saliency pipeline. Choosing the right threshold will remove only the cusp (I, J , K), but keep the cusp of (A, B, C) untouched. Since the circular arc of (A, B, C) is reduced, it will be removed after applying the pipeline again with the same threshold. 25

4.1 Shows equally-sized cusps on the boundary of a rectangle. Although these cusps all share the same importance, the shorter branch lengths inhibits the saliency pipeline to remove the cusps at the corners, as each branch has a maximum local thickness. Thus a different threshold on σ is required to remove each cusp. 28
4.2 Shows the detection ranges of the original saliency metric and the derivative of an importance measure. The shape is a circle with a singular bump on the surface; the corresponding skeleton is a A_1^2 curve segment with two A_3 endpoints. All skeleton points residing in the circle have the feature points F_1 and F_2. Detection of the derivative-based measure starts as soon as the importance measure stays constant, while the original saliency measure requires the ratio of ρ and D to be below c. Although c can be increased, within reason, to improve the detection range, setting it too high can also have the negative consequence of removing branches corresponding to features. 30
4.3 Shows the inverse mapping behavior of the boundary surface to the skeleton. Non-salient skeleton points with $\partial\rho(\mathbf{x}) = 0$ do not contribute to the reconstruction and therefore are absent from the inverse mapping. Small bumps coincide with long non-salient curve segments while larger bumps do not. 34
4.4 Comparison of the total error of different velocity fields as a function of t on the fandisk model. The result show that independent of the velocity field the error monotonically decreases as a function of the threshold, which implies robustness of the detection in terms of false positives. 35
4.5 Collection of models which are used throughout this thesis. 35
4.6 Cube model with convex 0-d noise. Skeletons are rainbow-colored using the importance metric ρ. With derivative filtering (f) more noise near edges is removed. 38
4.7 Bunny model with 1-d spherical creases. Skeletons are rainbow-colored using the importance metric ρ. When the derivative filtering is used, more noise is removed while maintaining core skeleton sheets. However some noise near areas with large amount of curvature remains. 39
4.8 Fandisk model with 1-d spherical creases. Skeletons are rainbow-colored using the importance metric ρ. 40
4.9 Chair model with 0-d spherical creases. Skeletons are rainbow-colored using the importance metric ρ. 41

5.1 An example of multi-axial features. The skeleton sheet corresponding to the bar intersects with the core skeleton, and can therefore not be removed with the original saliency pipeline or the adapted derivative pipeline. 42

5.2 Example of the importance metric introduced in equation (5.2). As the importance values are traced out from local roots or low-derivative nodes, the importance values are constant across branches. The ligature branches are separated by low-derivative nodes from the core skeleton and therefore have lower importance values. 44
5.3 An example of removal of intersecting sheets using the global streamline approach. 46
5.4 Removal of curvilinear elements on the cube model. 49
5.5 Removal of curvilinear elements on the fandisk model. 49
5.6 Removal of curvilinear elements on the chair model. 50

6.1 Demonstration of the small perturbations of the core skeleton and the EDT that persist in the reconstruction of the regularized skeleton. . 51
6.2 A comparison of how much filtering the EDT or skeleton affects the reconstruction. This demonstrates that regularizing the EDT has a far greater effect than smoothing the skeleton if the skeleton has been regularized. 52
6.3 Shows the effect of flat projection of the EDT on the inscribed circle and its feature points $\mathbf{a}_1$ and $\mathbf{a}_2$. The red circle shows the original radius, while the dashed orange circle shows the reduced radius where the circle no longer intersects with the segment $\overline{\mathbf{a}_1\mathbf{a}_2}$. 55
6.4 Comparison of reconstruction results of different EDT regularization methods on the chair model. 60
6.5 Comparison of reconstruction results of different EDT regularization methods on the fandisk model. 61
6.6 Comparison of reconstruction results of different EDT regularization methods on the hammer model. 61
6.7 Comparison of reconstruction results of different EDT regularization methods on the hammer model. 62

7.1 Comparison of IMA- and IE- skeletons, where the geodesic estimate is used for both methods. 66
7.2 Top: simplifications of the skeleton using USM with the maximum density $\lambda(\mathbf{x})$. Bottom: simplifications of the skeleton using the adapted IMA method (IE) with the geodesic importance $\rho(\mathbf{x})$ 68
7.3 Skeleton extraction using USM with $\lambda(\mathbf{x}) > 0$ on the fandisk model. The skeleton importance which is visualized is the geodesic metric $\rho(\mathbf{x})$. This shows that pruning the skeleton using ρ will cause holes to appear when using the USM skeleton. The cause is non-centered skeleton points. 68
7.4 Showcase of different noise parameters on a cube. 70
7.5 Visual comparison of the geodesic estimate $\phi(\mathbf{x})$, the graph geodesic $\rho(\mathbf{x})$ and the euclidean distance $E(\mathbf{x})$ on the horse model. The importance metrics are colored using the rainbow colormap, where the minimum and maximum are chosen independently for maximum contrast. 73

7.6 Difference of reconstruction speed using the naive evaluation and the REDT approach. 74

8.1 Bear model with 1-d spherical creases. 79
8.2 Chair model with 1-d spherical creases. 80
8.3 Cube model with 1-d spherical creases. 81
8.4 Shark model with 1-d spherical creases. 81
8.5 Fandisk model with 1-d spherical creases. 82
8.6 Pot model with 1-d spherical creases. 82
8.7 Bunny model with 1-d spherical creases. 83
8.8 Cat model with 1-d spherical creases. 83
8.9 Hammer model with 1-d spherical creases. 84
8.10 Comparison of skeleton-based smoothing to various other methods on the fandisk model. 88
8.11 Comparison of skeleton-based smoothing to various other methods on the cat model. 88
8.12 Comparison of skeleton-based smoothing to various other methods on the bear model. 89
8.13 Comparison of skeleton-based smoothing to various other methods on the cube model. 89

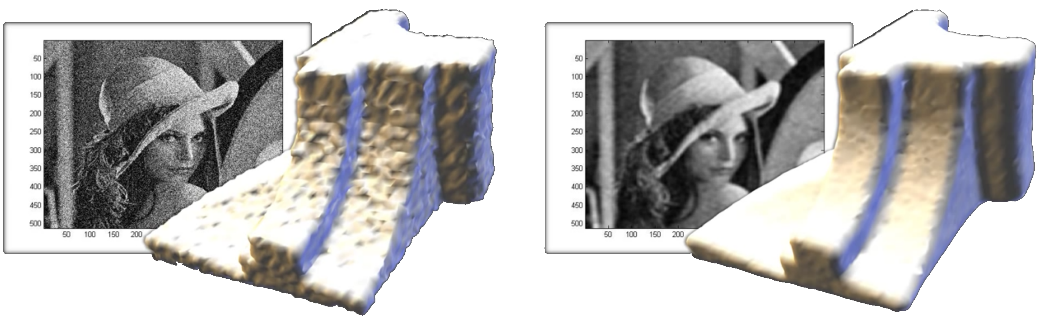
1 Introduction

With the proliferation of laser and optical scanners, 3D high resolution models are widely used in a variety of applications such as prototyping, computer-aided industrial designs, games and interactive virtual reality systems. Even with high quality scanners, however, the models inevitably have measurement noises from various sources [62, 54]. Similarly, shapes extracted from volume data (obtained by MRI or CT scanners, for instance) often contain significant amounts of noise, be it topological [63] or geometric [56], that needs to be removed before further processing.

Removing noise from a model while preserving its salient geometric details, or features, a process also called regularization, is however a challenging task. In many cases side-effects will occur during the regularization procedure, such as shape distortion and feature blurring, which greatly impairs the quality of the model and affect its use in some fine applications that rely on local properties of the model. Additionally, the situation may deteriorate when the extracted models contain additive noises from various sources, as some noise may be wrongly considered as features during optimization. To this end, a robust and effective optimization algorithm should remove noise, maintain features and avoid side-effects.

Naive denoising methods do not take geometrical features into account, and often smooth the surface homogeneously. Examples of such methods are low-pass filters, ball-erosions or least-squares filtering. These methods often succeed in removing noise, but also distort important edges. As is shown in figure 1.1, although most of the noise is successfully removed, the resulting feature blending is undesirable.

While a high number of feature preservation denoising approaches have been proposed [31, 62, 65, 6, 25, 26, 17, 48, 34], the research is far from mature and remains active due to its complexity. Most proposed methods are curvature based, and use local iterative approaches to perform denoising. These methods work well to remove small spurious noise, but also have inherent problems. The key problem of



(a) Original

(b) Smoothed

Figure 1.1: Example of naive smoothing using a Gaussian kernel, while the noise is reduced significantly, important features are blurred as well[61].

these methods is that they differentiate between noise (to be removed) and features (to be kept) using a *local* and *frequency-based* approach. Local methods can be used, indeed, to separate noise and features from smooth areas. However, to further separate noise from features, a scale (size) analysis is typically required. Such an analysis is not part of most local methods which, by construction, only analyze small neighborhoods of the shape. Generally, local iterative methods are good at detecting **local** geometric details, but they have difficulties differentiating noisy edges from **globally** important features.

It has been shown that feature preserving noise removal is also possible using skeletons [42]. A skeleton or medial axis can be seen as the locus of centers of n-spheres just touching the boundary of a 2D or 3D shape; it provides a complete descriptor of the shape. As skeletons contain both global and local information of the geometrical features, allow filtering and reconstruction, they are a good candidate for denoising. Noise-removal using skeletons works by removing points which correspond to small-scale boundary details (which are generally caused by noise) by trimming branches, and then reconstructing the shape from the simplified skeleton.

In 2D a salient edge preserving model has been shown to work quite well [58]. It is based on a saliency measure that relates feature size to local object size. It smooths out low-saliency convex features, and fills in low-saliency concave features. This is done by respectively inflating simplified skeletons of the mesh's foreground and background skeleton. Since global shape transitions correspond to high-saliency skeleton points, it can successfully distinguish noisy cusps and dents from salient edges. In contrast to other 2D shape smoothing methods, this method inherits several important properties from skeletons: the ability to distinguish between locally similar but globally different shape details, which in turn leads to the possibility to remove noise and keep salient features; and a simple and robust implementation, making use of only integral quantities, rather than differential ones, which adds robustness.

Unfortunately, the approach does not directly generalize to 3D. The problem is that simply thresholding the low salience skeleton points does not necessarily disjoint noisy skeleton components in 3D. Trying to remove complex noise with this method results in a (large) surface skeleton sheet which often partially connects with the core skeleton.

Hence, we state the key research question of this thesis: *Can we adapt the skeleton-based 2D noise removal method to handle 3D shapes, by using 3D surface skeletons?*

To this end, this thesis explores the problems related to saliency, and multiple improvements are proposed: i) derivative filtering, which allows improved detection of small scale noise near edges; ii) global importance filtering based on streamlines, which allows removing curvilinear noise structures from the surface, and iii) multiple smoothing methods which improve the quality of the reconstruction.

This thesis is organized as follows. First, a theoretical background of mathematical morphology and skeletons is given in chapter 2. Second, the core of the thesis given in chapters 3 to 6, where the full saliency pipeline is shown, the problems applying it to in 3D is discussed, and the solutions to mitigating these problems are given. Third, the implementation of this thesis, including the noise models which are used to test our results are discussed in chapter 7. Last, qualitative and quantitative results of the method are discussed in chapters 8 and 9, which also concludes this thesis.

2 Related work

Here we give a brief introduction to the important concepts of skeletons, as well as an overview of important related work. The notation of mathematical morphology is used in this thesis, and is introduced in section 2.1. Important concepts and definitions related to skeletons are given in section 2.2, and a brief overview of skeleton extraction methods is given in section 2.3. Related works based on skeleton regularization of 2D shapes are introduced in section 2.4. Finally, other feature-preserving denoising methods are discussed in section 2.5.

2.1 Mathematical morphology

The framework and notation used for skeletons have important connections in the field of mathematical morphology, and therefore, in this section, we recall some of the notation and terminology used in mathematical morphology. We primarily restrict ourselves to binary images; later we show the extension to grayscale images by means of threshold decomposition. For a more comprehensive discussion about these concepts, the reader may refer to [52]. In mathematical morphology, it is common to denote $\mathbb{E}$ as the universal set, i.e., the image domain, X as the set of foreground pixels (in the discrete case), and $\mathbb{E} \setminus X$ as the background. The power set of $\mathbb{E}$, the set of all subsets of $\mathbb{E}$, is denoted as $\mathcal{P}(\mathbb{E})$. In this thesis $A \in \mathcal{P}(\mathbb{E})$ and $A \subseteq \mathbb{E}$ is used interchangeably, depending on context.

An operator ψ is said to be a mapping $\psi : \mathcal{P}(\mathbb{E}) \to \mathcal{P}(\mathbb{E})$, meaning it operates on subsets of $\mathbb{E}$. The operator ψ is said to be:

1. *increasing*: if $X \subseteq Y$ implies that $\psi(X) \subseteq \psi(Y)$.
2. *extensive*: if $X \subseteq \psi(X)$.
3. *anti-extensive*: if $\psi(X) \subseteq X$.
4. *idempotent*: if $\psi(\psi(X)) = \psi(X)$

Examples of operators are dilation, and erosion. When an operator is both increasing and idempotent it is called a *filter*. An (algebraic) *opening* is a filter which is anti-extensive. Similarly, an (algebraic) *closing* is a filter which is extensive. An example of an algebraic opening (resp. closing) is an erosion followed by a dilation (resp. a dilation followed by an erosion). Formally, an operator is called an erosion iff the following property holds: $\cap\psi(X_i) = \psi(\cap(X_i))$. Similarly, an operator is a dilation iff it holds that: $\cup\psi(X_i) = \psi(\cup(X_i))$.

2.1.1 Connectivity

The elementary regions (pixels) of E are connected by means of a connectivity class $\mathcal{C}$:

Definition 1. *Let E be an arbitrary space. We call a connected class $\mathcal{C}$ a family in $\mathcal{P}(E)$ such that*

$$\emptyset \in \mathcal{C} \textit{ and } x \in E \textit{ implies } \{x\} \in \mathcal{C} \tag{2.1}$$

$$\textit{for any family } \{A_i\} \subseteq \mathcal{C},\ \cap A_i \neq \emptyset \textit{ implies } \cup A_i \in \mathcal{C} \tag{2.2}$$

Alternatively, we say that $\mathcal{C}$ defines a connectivity on E. The empty set, and all the singletons $\{x\}$ are connected. Furthermore the union of a family of connected sets is connected, if they have a non-empty intersection. We can explain the behavior of $\mathcal{C}$ by an example. Let the universal set E be the set of all integers, i.e. $E = \mathbb{Z}$, and let $\mathcal{C}$ define an adjacency relation on E so that every direct neighbor is connected. Then $\{1, 2\} \in \mathcal{C}$, and $\{2, 3\} \in \mathcal{C}$, and thus rule 2 implies that $\{1, 2, 3\} \in \mathcal{C}$. However $\{1, 3\} \notin \mathcal{C}$, as the numbers 1 and 3 are not direct neighbors in E. When $E = \mathbb{Z}^2$ common examples of connectivity classes are the four- and eight- adjacency relationships.

Every set $X \in E$ can be written as a union of connected sets that are pairwise disjoint and of maximal extent, so that $A \subseteq X$, $A \in \mathcal{C}$ of maximal extent implies that no set $B \subseteq X$, $B \in \mathcal{C}$ exists where $A \subset B$. These sets are called connected components, or grains, commonly denoted as $A \Subset \mathcal{C}$. Visually they correspond to the collection of all pixels of a connected region. These connected components can be retrieved by means of a connectivity opening:

$$\Gamma_x(X) = \bigcup\{A_i \in \mathcal{C} | x \in A_i, A_i \subset X\} \tag{2.3}$$

It retrieves the connected component corresponding to the pixel x. The operator Γ_x is anti-extensive, increasing, and idempotent and thus is an algebraic opening. In conclusion, the family of connected openings $\{\Gamma_x | x \in E\}$ uniquely characterize the connectivity class $\mathcal{C}$, and proving certain properties of the connectivity opening [52], proves that the related family is a valid connectivity class.

2.2 Skeletons

In this section we briefly describe the skeleton, or equivalently the Medial Axis, and its relation to its (Euclidean) distance transform. The definitions are first given for the continuous case, i.e., $\mathbb{E} = \mathbb{R}^d$; later the generalization to the discrete case is discussed. Although the skeleton can be defined on arbitrary distance metrics, we only consider Euclidean skeletons. For any vector $\mathbf{x} \in \mathbb{R}^d$ the length is denoted by $||\mathbf{x}|| = \sqrt{\sum x_i^2}$. Let $B = \mathbb{R}^d \backslash X$ be the background. Then the Euclidean distance transform D of B is the function that assigns to every point $\mathbf{x} \in \mathbb{R}^d$ the distance to its nearest background point. Formally it is defined as:

$$D(\mathbf{x}, B) = \min\{||\mathbf{x} - \mathbf{y}|| \mid \mathbf{y} \in B\} \tag{2.4}$$

The feature transform $\mathcal{F}$ is defined as the set-valued function that assigns $\mathbf{x} \in \mathbb{R}^d$ to its closest boundary points. Formally, it is defined as:

$$\mathcal{F}(\mathbf{x}, B) = \{\mathbf{y} \in B \mid ||\mathbf{x} - \mathbf{y}|| = D(\mathbf{x}, B)\} \tag{2.5}$$

There are two common skeleton definitions, the Real Medial Axis (RMA) and Center of Maximal Disk (CMD) skeletons [27]. Both are explained in the following sections.

2.2.1 CMD skeletons

Let the following set-valued function describe a d-sphere: $S(\mathbf{x}, r) = \{\mathbf{y} \in \mathbb{R}^d \mid ||\mathbf{x} - \mathbf{y}|| < r\}$. For every point $\mathbf{x} \in \mathbb{R}^d$ we can form the largest open sphere $D(\mathbf{x}, r)$ that is disjoint to B; this is defined as the inscribed sphere of $\mathbf{x}$ (commonly referred to as inscribed disk). If an inscribed sphere of point $\mathbf{p}$ is not contained in any other inscribed sphere of B, it is called a maximal sphere. The set of centers of these maximal spheres are the skeleton points. CMD skeletons together with the sphere radii are complete shape descriptors, meaning they can be used to construct the shape of the original domain. However, they are not guaranteed to be connected, and thus they are less useful for processing purposes.

2.2.2 RMA skeletons

The medial axis of an object is the set of all points having more than one closest point on the object's boundary. Formally a point $\mathbf{x} \in X$ is a skeleton point ($\mathbf{x} \in \mathcal{S}(X)$) iff $|\mathcal{F}(\mathbf{x}, B)| \geq 2$ for an object background $B \subseteq \mathbb{E}$. In the 3D case RMA skeletons are also called surface skeletons. There is also a special case of points where $|\mathcal{F}(\mathbf{x}, B)| \geq 3$, these are called curve skeleton points (although definitions may vary).

2.2.3 Discrete case

Regard $\mathbb{Z}^d$ as a grid embedded in $\mathbb{R}^d$. The elements of $\mathbb{Z}^d$ are called grid points. For the special cases $d = 2$ or $d = 3$ the elements are called respectively pixels or voxels. The Euclidean distance transform can be evaluated in a discrete manner by finding for each discrete grid point $\mathbf{x} \in \mathbb{R}^3$ its closest background point and computing its distance to this point.

In the continuous domain CMD and RMA skeletons are equivalent. However, there are no straight-forward definitions for the RMA skeletons in the discrete case. The most straight-forward approach to generalize the CMD skeletons to the discrete case, is to take for $\mathbf{x} \in \mathbb{Z}^d$ the inscribed sphere defined by $S(\mathbf{x}, r) \cap \mathbb{Z}^d$. Similarly as in the continuous case, the set of centers of the maximum inscribed spheres are then the skeleton points.

The RMA skeleton is harder to generalize according to its definition. Assume that X is discrete so that $X \subseteq \mathbb{Z}^d$, then there is no guarantee that $\mathcal{S}(X) \subseteq \mathbb{Z}^d$. In fact, the intersection $\mathcal{S}(X) \cap Z^d$ can be empty. Thus a fully reconstructible RMA skeletons of discrete images according the exact definition requires a mapping $\mathcal{S} : \mathcal{P}(\mathbb{Z}^d) \to \mathcal{P}(\mathbb{R}^d)$ [27]. However it is often desirable to have skeletons that map to the integer domain. Many algorithms exist which approximate RMA skeletons in the integer domain using various approximation schemes [27, 51, 29]. These approximate skeletons are often not fully reconstructible, but do contain similar features compared to real RMA skeletons. Most of these algorithms can recover connected surface skeletons, which is why they are intrinsically more useful for processing purposes than CMD skeletons.

Similarly as extracting RMA skeletons, the feature transform in the discrete case needs some adaptations, as the extracted skeleton in the integer domain is not guaranteed to be centered, i.e., the true real medial axis is a subset of $\mathcal{P}(\mathbb{R}^d)$ and not $\mathcal{P}(\mathbb{Z}^d)$. This means that the restricted feature transform to $\mathbb{Z}^d$ may well be everywhere single-valued, so that the consideration of neighboring points is needed. There are multiple solutions, but an approach that has worked well is the extended feature transform [49]:

$$\bar{\mathcal{F}}(\mathbf{x}) = \bigcup_{\mathbf{y}\in X \wedge \{\mathbf{x},\mathbf{y}\}\in\mathcal{C}} \mathcal{F}(\mathbf{y}) \tag{2.6}$$

It is the union of the features of $\mathbf{x}$ and of all the connected neighbors of $\mathbf{x}$. This extended feature transform is used in this thesis.

2.2.4 Reconstruction

An important concept in skeleton algorithms is reconstruction. Since skeletons together with the sphere radii are complete shape descriptors, they can be used to reconstruct the original shape. The reconstruction is given by the union of all spheres centered on the points comprising the skeleton, each with a radius given by the distance transform $D(\mathbf{x}, B)$. Let $\mathcal{S}$ be the skeleton of X then we have that:

$$X = R(\mathcal{S}) = \bigcup_{\mathbf{x}\in\mathcal{S}} \{\mathbf{y} \mid y \in D(\mathbf{x}, B)\} \tag{2.7}$$

Note that naively computing the reconstruction according to this definition leads to an $O(n^2)$ algorithm; better approaches exist, as is shown in chapter 7.

2.2.5 Surface skeletons

Skeletons of 2D geometry have a relatively simple structure: they are curve segments bounded by either an end-point corresponding to a curvature extremum on the boundary of the shape, or by a junction point where three branches meet [23]. Surface skeletons, also called medial surfaces, have a more complex structure, and require the notion of contact points to fully describe its shape[1].

[1] The notion of contact points are also a powerful descriptor of 2D skeletons as is shown in [22].

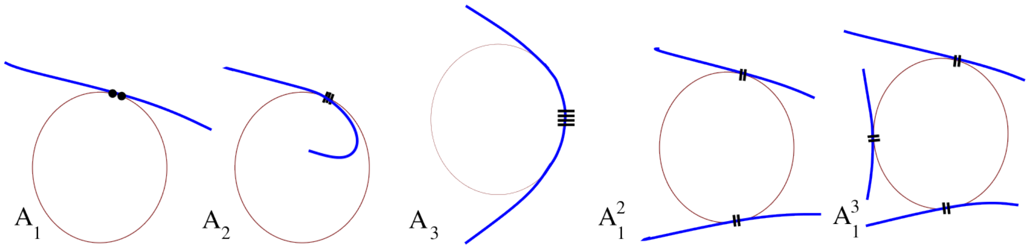


Figure 2.1: Illustration of the notation A_k^n based on contact of a curve with a circle [23].

The classification of surface skeleton points is based on the notion of contact with spheres, i.e., the loci of sphere osculating sources [22]. Let A_k^n denote a circle (in 2D) or a sphere (in 3D) osculating a boundary element at n distinct points and $k+1$-degree of contact. Here, degree of contact is defined as the order of shared derivatives. An example is shown in figure 2.1: $k = 1$ denotes regular tangency; $k = 2$ denotes a sphere of curvature for a surface patch; $k = 3$ denotes a sphere of curvature at a ridge point, and is of maximum contact. When $k > 1$ the parameter n is omitted, as there are infinitely many contact points. It is noteworthy that even degrees of contact do not occur at inscribes spheres of skeleton points, as they cannot be tangent to maximum inscribed spheres (i.e., they intersect with the boundary).

By classifying each skeletal point by its order of contact, it can be shown that generically the surface skeleton consists of five types of points which are then organized into sheets, curves, and points [21]:

1. **Sheets** (manifolds with boundary) which are the locus of bitangent spheres with regular tangency A_1^2.

2. **Curves**, which come in two types: i) the intersection curve of three sheets and the locus of centers of tri-tangent spheres A_1^3, and, ii) the boundary of sheets which are the locus of centers of spheres whose radius equals the larger principal curvature, i.e., higher order contact A_3 points.

3. **Points**, which come in two types: i) centers of quad-tangent spheres, and, ii) centers of spheres with at-least one regular tangency and one higher order tangency (A_1^{1+}, A_3).

Note that A_1^3 points are commonly referred to as junction points, as they lie on the intersection of three or more sheets. An example of a surface skeleton of a elliptical cylinder is shown in figure 2.2. The skeleton consists of three sheets formed by A_1^2 points, as is shown in (b), (c) and (d). For the upper sheet the maximum spheres are close to the flat top of the cylinder and tangent both to it and to the curved surface. For the flat sheet in (d) the spheres are tangent to the front and back curved surfaces of the cylinder.

The curve skeleton (cs) points A_1^3 are shown in (d), they lie at the intersection the central and bottom sheet, and likewise at the intersection of the top and central sheet. There are also A_3 cs points which lie at the sides of the central sheet, they are tangent to the ellipsoidal surface of the cylinder.

The singular points A_1^{1+}, A_3 lie at the corners of the central sheet. They are A_1 tangent to the bottom and top of the cylinder, and A_3 tangent to the ellipsoidal surface of cylinder. An example of a more complicated skeleton shape is shown in figure 2.3.

2.3 Skeleton computation

In the past decades, various skeleton extraction methods have been proposed. Generally, they can be coarsely classified into six types: the thinning algorithms [33, 37, 1, 36], the discrete domain algorithms based on the Voronoi diagrams [46, 7, 19, 43],

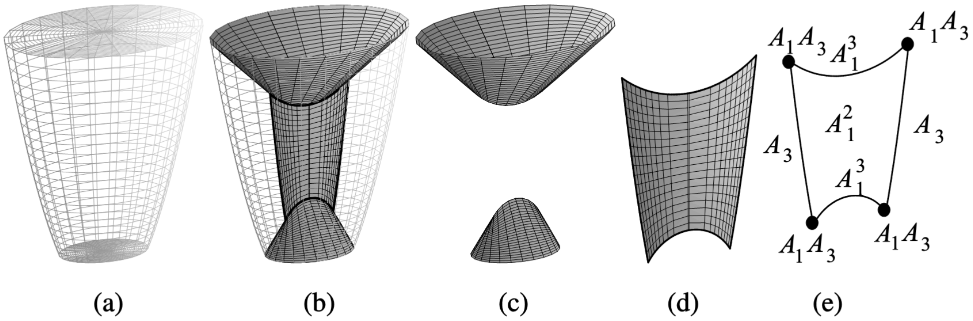
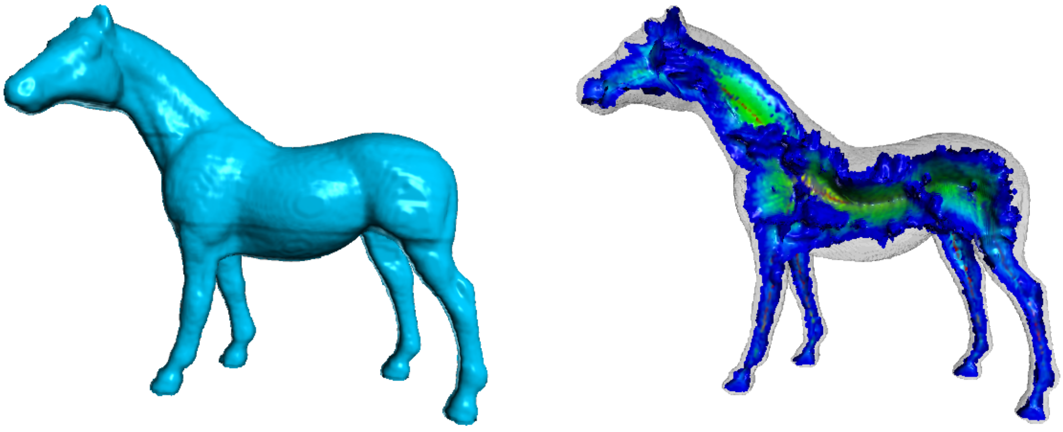


Figure 2.2: Illustration of the surface skeleton of an elliptical cylinder [21]: (a) the cylinder; (b) the cylinder and its surface skeleton; (c) the top and bottom sheets of the skeleton; (d) the central sheet which connects the bottom and top sheets; (e) labeled contact points of the central sheet.



(a) Horse model

(b) Unsimplified surface skeleton.

Figure 2.3: An example of a surface skeleton of a horse model. The rainbow colored map shows the simplification metric of [29].

the algorithms based on distance transforms or related fields [20, 11, 2, 15, 16, 3, 35], iterative shrinking of the object contour [32, 24, 64], boundary collapse methods [51, 29], and the algorithms based on mathematical morphology [18, 30, 27]. Only a small list of the mentioned algorithms are also directly applicable to 3D shapes [36, 19, 43, 51, 29, 27]. Most algorithms in 3D either output curve skeletons [36, 43]: a set of connected 1D curves centered with respect to the object boundary, or output surface skeletons [19, 27]: a set of 2D manifolds or so called sheets. Some algorithms can output both curve and surface skeletons [51, 29]. These algorithms rely on a global collapse measure, which after thresholding outputs hierarchical skeletons of different scale levels. In contrast to 2D and surface skeletons, 3D curve skeletons admit many definitions, and as such a formal relationship between the surface skeleton and the curve skeleton is not unanimously accepted. Curve skeletons only capture the topology of the shape, and not its geometry, and as a consequence cannot be reconstructed to its original shape, and therefore only surface skeletons are useful for our work.

For a more comprehensive discussion about skeleton computation, the reader may refer to the recent state of the art report [55].

2.4 Skeleton regularization

Skeleton pruning is a critical step in skeleton processing and analysis. Although the skeleton has many desirable characteristics as a shape descriptor, it is very sensitive to noise on the shape boundary [42]. The goal of skeleton pruning is to remove the unnecessary skeleton points or branches arising from noise or perturbations on the boundary [4].

Skeleton pruning methods can be classified in two important categories:

- Importance thresholding: This kind of method is based on assigning significance values to skeleton points and removes insignificant points according to a given threshold. [51, 29, 59, 58, 53]

- Branch pruning: These methods measure the of each branch as a whole and prunes the skeleton branch-by-branch [5, 42, 41, 57].

2.4.1 Importance thresholding

Importance thresholding approaches rely on a salience measure that is defined for each skeleton point, by a mapping $\rho : \mathcal{S}(X) \mapsto \mathbb{R}^+$, using local characteristics of the skeleton point, e.g. speed, radius, lengths, area, thickness, etc. Examples of useful local characteristics are shown in figure 2.4. These metrics are the angle θ of the feature points P^+ and P^-, the chord distance (length of circular arc), the geodesic distance (distance over the boundary), or the area.

A few important methods based on importance thresholding need to be mentioned: Ogniewicz and Kübler [47] introduced several length based significance measures for a given skeleton point, such as the length of the chord between two feature points (the points of the maximal sphere centered at the skeleton point, which are

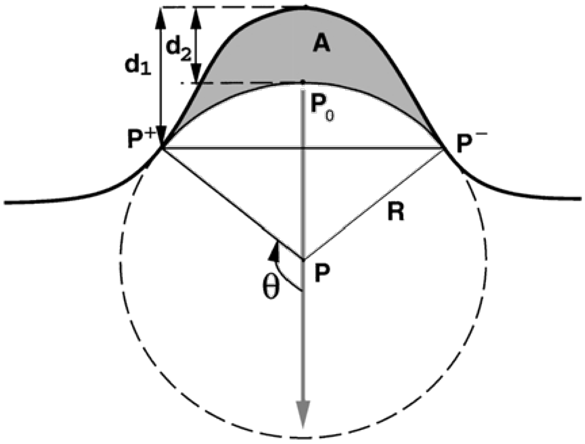


Figure 2.4: This figure (from [57]) illustrates the various local characteristics of a point P which can be used as the basis for a pruning metric of the skeleton. Examples are the angle θ of the feature points P^+ and P^-, the chord distance (length of circular arc), the geodesic distance (distance over the boundary), or the area A.

tangent to the boundary) and the length of the shortest boundary segment between two feature points.

Shen et al. [53] introduced a measure called bending potential ratio (BPR), in which the decision regarding whether a skeletal point should be pruned or not is based on the bending potential of the boundary segment that corresponds to the branch. Reniers et al. [51] introduce a method for surface skeletons which uses a global importance measure that assign to each point on the skeleton a part of the object surface, called the collapse. The size of the collapse is used as a uniform importance measure for the curve and surface skeleton. Jalba et al. [29] presented a similar method which uses conservative mass transport process which assigns the collapsed mass of the boundary to each skeleton point, improving in some ways on the result of [51]. The discussed importance thresholding methods share a common disadvantage: while they do prune noisy branches, they also shorten branches of coarse-scale corners.

As introduced before, the salient edge preserving model by Telea [58] has been shown to work quite well for 2D models. It is based on a saliency measure that relates feature size to local object size. Any importance measure can be used for the feature size, including the methods explained above. Unlike the previously discussed methods, it does not shorten branches which correspond to salient shape features and preserves coarse-scale corners.

2.4.2 Branch pruning

The branch pruning methods do not shorten important skeleton branches, they only remove branches as a whole, and therefore they are better in preserving sharper features. Tek and Kimia [57] proposed the earliest branch removal method by using a splice transform (data structure containing the symmetry set, which consists of both the foreground and background skeleton). They iteratively remove terminal branches from the splice transform based on the order their saliency measure, which is defined as the area difference between the smoothed and original shape divided by the branch length. Just as in this thesis their end-goal is smoothing the boundary. Bai et al. [5] proposed an effective pruning method, which can be integrated into a skeleton extracting process. The main idea is partitioning the object's contour into segments by discrete curve evolution (DCE). A large downside of this method is that it only works with Genus 0 shapes (shapes without holes). Another branch-removal method, proposed by Liu et al. [41], uses a combination of two metrics: (i) the reconstruction contribution, which is measured by the residual shape area, the area that can only be reconstructed by a branch, relative to the whole shape area, and (ii) its visual contribution (VC), which is measured by the length of residual part of the skeleton branch, the part that is not contained in the maximal sphere centered at the adjacent branch point. The same authors later improved this method by combining other significance measures using information fusion [42].

Although branch methods have shown to be successful in 2D, they do not generalize, at all, to surface skeletons. They rely on a rooted graph structure of line segments that only exists for 2D skeletons. The closest structure for surface skeletons are Medial Axis hypergraphs [21]. Their increased complexity, and its large number of transitions between skeleton points makes it much harder to work with than regular branch pruning methods. Therefore our focus is on importance thresholding methods, as they are easier to generalize to the 3D case.

2.5 Other denoising methods

There is a large variety of feature-preserving smoothing methods which are not based on skeletons. A few of these are explored here.

A selection of methods use a saliency map of the surface of the mesh that assigns high importance values to important features such as strong edges. The saliency map is then used in the weighting of the optimization procedure of the method so that important features are not distorted. An early feature preserving method that uses this model is 'Anisotropic Smoothing of Point Sets' [34]. For the saliency mapping they use directional and principal curvatures as well as the Weingarten map that is also known as the shape operator. They use anisotropic geometric mean curvature flow for their optimization procedure. There are other methods which also based on a saliency-based mapping [26, 12]. The large problem of these methods is that they are based on curvature-based measures based on first-order or second-order derivatives, which are only local and unstable when a large amount of noise is present.

There are also other methods which use robust statistics or bilateral filtering. They rely on a similarity measure or empirical influence function that changes the dependence of the optimization procedure on the value of the points, which makes them more robust to outliers. This is important, as feature preserving smoothing can be seen as estimating a surface in the presence of outliers. An important early work based on this technique is 'Non-iterative, feature-preserving mesh smoothing' by Jones et al. [31]. They use bilateral filtering, where the tangent planes based on filtered normals are used as a basis for their similarity measure. Another important work based on this technique is 'Preserving point set surfaces based on non-linear kernel regression' by Öztireli et al. [48]. They use robust statistics (M-estimators) to create a implicit least squares procedure that has good edge-preserving qualities. The main advantage of their method is that is performs well with a low number of samples. There are a large variety of methods that are also based on robust statistics or bilateral filters [17, 62, 39, 28]. The downside of these methods is that they rely on local neighborhoods of points, so they are unable to differentiate globally important edges from local (possibly noisy) geometric details. Another disadvantage is that these methods need a certain finite kernel size of to estimate curvature. If the size is set too small, then one gets poor curvature estimates, which become useless. However when the kernel size is set too large, the curvature estimates are stable, but are not localized in the input shape, but in the filtered version thereof, which results in poor localization of the features of the shape.

3 Saliency pipeline

Here the saliency pipeline as introduced in [58] is explained. This important previous work is detailed, as it is used as the basis of the proposed framework for surface skeleton filtering. The framework is useful because it offers a way to distinguish between locally similar but globally different small-scale details, thus, a way to classify perturbations into features and noise, which is ultimately what is needed for the denoising process. The saliency pipeline is explained in section 3.1, where also some new terminology and notation is introduced to unify related work. The relation of the saliency pipeline to mathematical morphology is given in section 3.1.3. In section 3.2 some of the problems of implementing this framework for surface problems are introduced.

3.1 Saliency metric

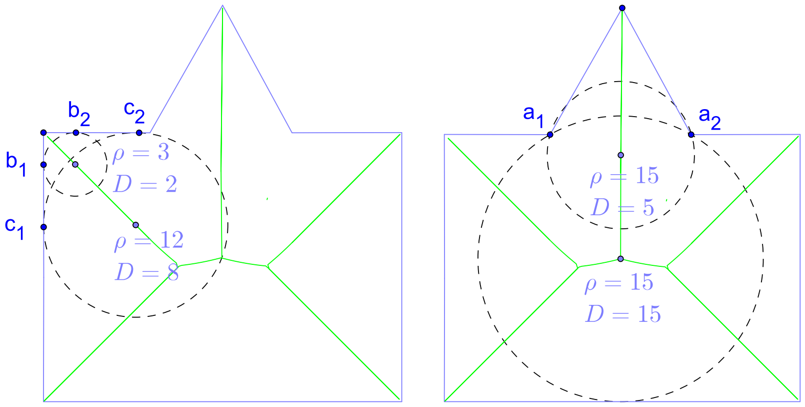
A framework is proposed by Telea [58] which performs filtering using a pruning metric on the foreground and background skeleton. It aims to remove shape perturbations, generally classified as noise, without smoothing corners. The shape perturbations can be divided into cusps and dents, and have different properties than important features such as corners. Here, cusps (resp. dents) are defined as small scale convex (resp. concave) protuberances of the shape boundary, that are surrounded by relatively flat or low-curvature areas. In the proposed framework the cusps are removed in the foreground skeleton, while the dents are removed in the background skeleton.

At its core it uses the following saliency metric:

$$\sigma(\mathbf{x}) = \frac{\rho(\mathbf{x})}{D(\mathbf{x})} \tag{3.1}$$

Where $\rho(\mathbf{x})$ is a local pruning metric proportional to the size of the features (originally taken as the geodesic distance of the feature points), and $D(\mathbf{x})$ is the Euclidean distance transform. The pruning metric ρ should be proportional to the size, so that longer features are more salient than shorter ones. The effect is that it allows differentiating between cusps and corners, as is explained below.

Along the cusp branches, ρ increases until the branches leave the cusp and enter the rump of the shape. After that, ρ stays constant on that branch until the branch joins the core skeleton (because ρ is proportional to the size of the cusp). These skeleton segments are called ligature branches. The effect is that further along the so-called ligature-branch the distance transform increases, while ρ stays constant. Thus we have that in the limit $\lim_{D(\mathbf{x})\to\infty} \sigma(\mathbf{x}) = 0$ for ligature branches. On globally important features such as corners ρ keeps increasing on the connected branch until it connects with the core skeleton. This allows us to differentiate between corners and cusps, as ligature branches (related to cusps) contain small $\sigma(\mathbf{x})$ values, while



(a) corner

(b) cusp

Figure 3.1: An illustration which shows the difference between corners and cusps: in (a) both the importance ρ and the distance D steadily increases on a branch, while in (b) D increases, but ρ stays constant.

corner branches have constant $\sigma(\mathbf{x})$ over the entire range of the branch. An example is shown in figure 3.1.

Following equation (3.1) we have a constant value of $2/\tan(\alpha)$ for points along the skeleton branch of an angular cusp of aperture angle α [58]. It has a theoretical limit of $P(M)/\phi$. Here, $P(M)$ is the perimeter in 2D or the maximum perimeter of the surface M in 3D, and ϕ is defined as the minimum local thickness so that $\phi = \min_{\mathbf{x}\in\mathcal{S}} D(\mathbf{x})$.

3.1.1 Pipeline

The pruning measure $\sigma(\mathbf{x})$ can be used to regularize the skeleton, and the processed skeleton can then be used to reconstruct the model where most of the noise is removed, while retaining the important features. The pipeline is divided into two stages, where in the first stage convex noise is removed using the foreground skeleton, and in the second stage concave noise is removed using the background skeleton. Thus in the first stage of the pipeline the cusps on the surface (or perimeter in 2D) are removed. This is done by the following steps:

1. Compute the foreground skeleton of the noisy binary image.
2. Prune skeleton based on saliency measure $\sigma(\mathbf{x})$.
3. Reconstruct the simplified skeleton; which will result in the original image where small cusps are removed.

In the second step the skeleton is simplified by using the following threshold rule $\sigma(\mathbf{x}) \geq c$ where $c \in \mathbb{R}$. There is no guarantee that the output is a connected skeleton. As $\lim_{D(\mathbf{x})\to\infty} \sigma(\mathbf{x}) = 0$, the tips of the ligature branches have larger values, which will be retained after thresholding. Thus this threshold rule will result in images that contain the core skeleton and disconnected tips of the ligature branches. The full removal of the ligature branches is performed by selecting the component with the largest importance value. The reasoning behind this is that $\rho(\mathbf{x})$ has the largest importance value located in the center of the 2D skeleton, and is therefore very unlikely to be located at the disconnected fragments (which should be far away from the center and close to and including the branch tips).

Definition 2. *Let $\mathbf{y} \in \mathbb{R}^n$ be the skeleton point with the largest importance value so that $\mathbf{y} = \operatorname{argmax}_{\mathbf{x} \in \mathcal{S}} \rho(\mathbf{x})$, and let $\Gamma(X)$ be the connectivity opening as defined in chapter 2, then the skeleton simplification is given by:*

$$\omega_\sigma(\mathcal{S}) = \Gamma_{\mathbf{y}}(\{\mathbf{x} \in \mathcal{S} \mid \sigma(\mathbf{x}) \geq c\}) \tag{3.2}$$

Since the core skeleton should contain the largest importance value, the operator $\omega_\sigma(\mathcal{S})$ removes ligature branches, but the core skeleton is kept in full. The simplified skeleton is reconstructed so that the output of the first stage is given by $R(\omega_\sigma(\mathcal{S}))$, which gives a processed image where the cusps are removed.

Definition 3. *Let $X \in \mathcal{P}(\mathbb{E})$ be the set of foreground pixels that make up a single connected component, and R the reconstruction operator as defined in chapter 2 then we define the operator δ_c as:*

$$\delta_c(X) = R(\omega_\sigma(\mathcal{S}(X))) \tag{3.3}$$

In the second stage of the pipeline the same steps are repeated, but then for the background image of the reconstructed result. This way it is possible to detect ligature branches corresponding to dents, and remove concave noise. Using the dual structure for the operators the second stage can be defined by the background image:

Definition 4. *Let $X \in \mathcal{P}(\mathbb{E})$ be the set of foreground pixels that make up a single connected component, and R the reconstruction operator as defined in chapter 2 then we define the operator $\bar{\delta}_c$ as:*

$$\begin{aligned} \bar{\delta}_c(X) &= \mathbb{E} \setminus \delta_c(\mathbb{E} \setminus X) && (3.4) \\ &= \mathbb{E} \setminus R(\omega_\sigma(\mathcal{S}(\mathbb{E} \setminus X))) && (3.5) \end{aligned}$$

The assumption here is that the domain $\mathbb{E}$ is a bounded subset so that there exists a finite Lebesgue measure, otherwise the skeleton of the background image is ill-defined.

The two stages of the pipeline are combined by applying them in succession: $\bar{\delta}_c \delta_c(X)$. The result is that in the individual stages both the convex and concave noise are removed. The pipeline is further outlined in figure 3.2. In this example the cusps and dents of a rectangle are removed. The first stage defined by δ_c is shown in the first row, where in the second row $\bar{\delta}_c$ is shown.

length of the shortest path, over the shape boundary, between

3.1.2 Choice of importance measure

Although there is no strict requirement of ρ in the saliency pipeline, it is proven to work well with geodesic distances for 2D images, where the geodesic distance of the two most relevant features are used. In particular:

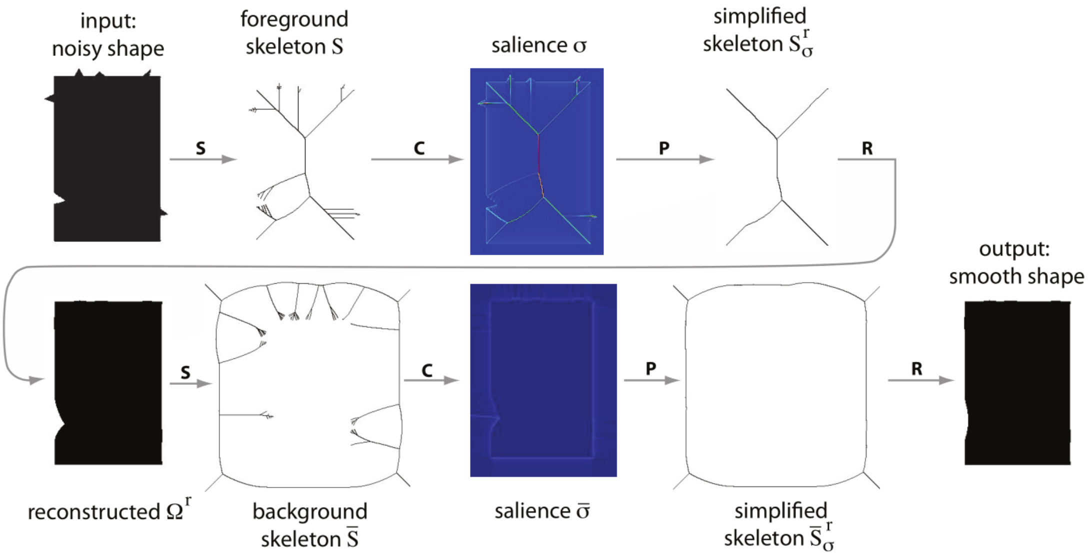


Figure 3.2: The saliency pipeline for 2D images [58]. In the first step the cusps are removed by removing the ligature branches of the foreground skeleton, which is then reconstructed. Respectively, the dents of reconstructed shape are removed by trimming ligature branches of the background skeleton. The reconstruction of the of the background skeleton gives the final result where both cusps and dents are removed.

Definition 5. *let $\mathcal{F}$ be the feature transform as defined equation* (2.5)*, and let g be the function that gives the length of the shortest path, over the shape boundary, between two feature points, then the relevant feature transform $\hat{\mathcal{F}} : \mathcal{S} \mapsto \mathbb{E} \times \mathbb{E}$ is defined by:*

$$\hat{\mathcal{F}}(\mathbf{x}) = \operatorname*{argmax}_{(\mathbf{a},\mathbf{b})}\{g(\mathbf{a},\mathbf{b}) \mid (\mathbf{a},\mathbf{b}) \in \mathcal{F}(\mathbf{x}) \times \mathcal{F}(\mathbf{x})\} \tag{3.6}$$

Then the importance measure ρ can readily be defined by $\rho(\mathbf{x}) = g(\hat{\mathcal{F}}(\mathbf{x}))$. To improve computation speed g is often replaced by the Euclidean distance or the angle between the features for $\hat{\mathcal{F}}$, where only the actual geodesic distance is used for ρ.

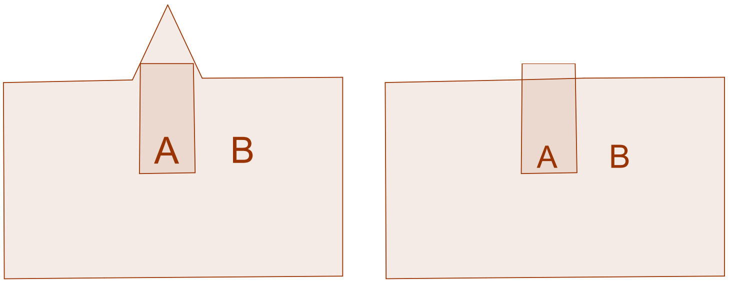
In 2D the geodesic distance has the special property that it is monotonically increasing for genus 0 shapes (shapes without holes). This means that the importance ρ is increasing as one orders skeleton points from the skeleton boundary to its center. Formally we have that:

Definition 6. *Let $\mathcal{S}$ be the skeleton of X; let ρ be an arbitrary importance measure on $\mathcal{S}$, and let $\mathbf{q} = \operatorname{argmax}_{\mathbf{y}} \rho(\mathbf{y})$ be the root of $\mathcal{S}$, then ρ is said to be a monotonically increasing so that for all $\mathbf{x} \in \mathcal{S}$ there exists an increasing path $P \in \mathcal{S}^k$ with $k \in \mathbb{N}^+$ the path length, where*

$$P_1 = \mathbf{x} \textit{ and } P_k = \mathbf{q} \tag{3.7}$$

$$\{P_i, P_{i+1}\} \in \mathcal{C} \textit{ for all } P_i \in P,\ i \neq k \tag{3.8}$$

$$a \leq b \textit{ implies } \rho(P_a) \leq \rho(P_b) \tag{3.9}$$



(a) $A \subseteq B$

(b) $\delta(A) \nsubseteq \delta(B)$

Figure 3.3: An example showing why the operator δ is not increasing. The bump of A is removed as it corresponds to a ligature branch, however the rectangle B contained in A does not contain ligature branches, and is therefore kept intact.

In the case where there exists multiple roots then a monotonically increasing path must exist for all $\mathbf{q}_i \in \operatorname{argmax}_{\mathbf{y}} \rho(\mathbf{y})$. This is equivalent to saying that for all thresholds $c \in \mathbb{R}$ on ρ the skeleton $\mathcal{S}$ stays connected. Therefore, thresholding ρ cannot break $\mathcal{S}$ into multiple connected components. This is an important property for saliency filtering, as otherwise it might also disconnect the core skeleton as well as the ligature branches at lower thresholds values.

3.1.3 Relation with mathematical morphology

As the saliency metric is a topological operator, it is interesting to see properties of the operator in a mathematical morphology setting, as it allows us to see its limitations, but also use this theoretical framework to create a grayscale extension. In particular, some research has been done whether the saliency operators are morphological filters (such as erosions, openings, or closings).

Lemma 7. *Let $X \in \mathcal{P}(\mathbb{E})$ be the finite set of foreground pixels that make up a single connected component, then $\delta_c(X)$ is anti-extensive so that $\delta_c(X) \subseteq X$.*

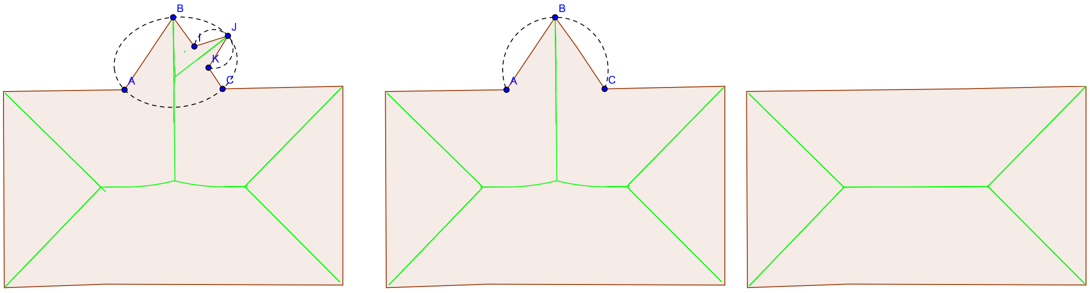
Proof. For any threshold c we have that $\omega_\sigma(X) \subseteq \mathcal{S}(X)$. Since R is an increasing operator, it follows that $R(\omega_\sigma(X)) \subseteq R(\mathcal{S}(X))$. Thus it holds that $\delta_c(X) \subseteq X$. □

Note that R is only increasing if the distance transform stays constant, which holds here as at both sides of the equation the distance transform is of the same X.

Lemma 8. *Let $X \in \mathcal{P}(\mathbb{E})$ be the finite set of foreground pixels that make up a single connected component, then $\delta_c(X)$ is eventually idempotent, so that there exists a $k \in \mathbb{N}$ so that $\delta_c(X)^k = \delta_c(X)^{k+1}$*

Proof. Since $\delta_c(X)$ is anti-extensive it holds that $\delta_c(X)^{k+1} \subseteq \delta_c(X)^k$. Either $\delta_c(X)^k$ is strictly anti-extensive so that there exists a k where $\delta_c(X)^k = \emptyset$ where it follows that $\delta_c(X)^{k+1} = \emptyset$, or $\delta_c(X)^k$ is not strictly anti-extensive, for which it must hold that there exists a k where $\delta_c(X)^k = \delta_c(X)^{k+1}$. □

In practical cases k is almost always one, as when all ligature branches are removed under a threshold, they are not reintroduced after reconstructing the shape.



(a) Two bumps (b) One bump ($k = 1$) (c) No bumps ($k = 2$)

Figure 3.4: A demonstration of the non-idempotent behavior of the saliency pipeline. Choosing the right threshold will remove only the cusp (I, J , K), but keep the cusp of (A, B, C) untouched. Since the circular arc of (A, B, C) is reduced, it will be removed after applying the pipeline again with the same threshold.

However, there exist pathological cases in the case of hierarchical ligature branches (a small cusp on another cusp). An example of such a case is shown in figure 3.4.

Although δ is anti-extensive and eventually idempotent, it is not increasing. An example where δ_c is not increasing is shown in figure 3.3. It follows directly that δ_c does not a constitute a classical filter, and is therefore also not an opening. Even though it is not increasing, it is still possible create a grayscale extension similar to attribute thinnings [8], which are also non-increasing.

Grayscale extension

Similar as attribute thinnings [8], the operator δ can be generalized to grayscale. This can be done through threshold superposition [44], where a gray-scale image $f : \mathbf{x} \mapsto \mathbb{N}$ is decomposed into binary images, based on thresholding f at all levels $h \in [0, N-1]$, where N is the number of graylevels of the image. Formally this can be defined as:

$$T_h(f) = \{\mathbf{x} \in \mathbb{E} | f(\mathbf{x}) \geq h\} \tag{3.10}$$

In equation (3.10) all the pixels corresponding to values higher than threshold h are selected. It is decreasing with respect to h so that when $k > h$ it follows that $T_k(f) \subseteq T_h(f)$. This is also referred to as hierarchical nesting [44]. Non-increasing filters can be generalized by evaluating the binary filter at every threshold-level h and combining the resulting connected components.

For an increasing binary filter ϕ, the grayscale extension is evaluated by:

$$\phi_f(\mathbf{x}) = \max\{h \mid \mathbf{x} \in \phi(T_h(f))\} \tag{3.11}$$

In attribute filter terminology this corresponds to the direct rule. Evaluating this for non-increasing filters can result in unwanted side-effects [60]. For example, a point $\mathbf{x}$ might be subject to removal on most threshold levels, but if there is a single high threshold value where no removal takes place, then the grayscale filter has little or no effect.

Another grayscale filtering rule which proved to be more successful with anti-extensive filters is the subtractive rule [60]. In the discrete case it is defined as:

$$\phi_f(\mathbf{x}) = \sum_{h=0}^{N-1} \begin{cases} 1 & \text{if } \mathbf{x} \in \phi(T_h) \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

Effectively, using this rule is equivalent to counting the number of threshold levels which contains $\mathbf{x}$, which is more robust if a filter is not increasing, but is equivalent to the direct rule for increasing filters. Thus by substituting ϕ by δ_c, the saliency filtering can also be applied to grayscale images. This has not yet been investigated in practice.

Although this work is not used in the current pipeline, these relations are useful for future work. In particular, the grayscale extension might be useful for denoising CT, MRI or ultrasound scans which generally output grayscale voxel-based volumes.

3.2 3D extension

Although originally defined for 2D skeletons, the saliency metric can also be applied to surface skeletons. The complete pipeline carries over without any large adjustments, except that $X \subseteq \mathbb{Z}^3$ instead of $\mathbb{Z}^2$. However, there are a number of problems which makes shape denoising problematic for most models:

- Low detection range problems: Typically a 3D (voxel) shape is sampled at lower resolutions, with respect to the level-of-detail of the shape, than typical 2D (binary pixel) shapes. Therefore, noise near edges is harder to distinguish from features, requiring an improved detection range.

- Multi-axial features: Let axes in this context be mutually perpendicular planes in a local coordinate system that are feature-aligned. In 3D, important features in one axis can overlap with noise in a different axis, so that the ligature sheet is orthogonal to the core skeleton. This causes complete intersection of the ligature sheets and the core skeleton. The result is that simply thresholding the low salience skeleton points does not necessarily disjoint noisy skeleton components in 3D. Trying to remove complex noise by using the connectivity operator thus results in a (large) surface skeleton sheet which often partially connects with the core skeleton.

- EDT ripples: After removing all ligature sheets, small noise on the surface still persist due to noise that persist as perturbations on the core skeleton and the EDT. In 2D this is barely noticeable, however with 3D models when using any modern rendering technique the small perturbations are much more visible due to light reflection.

To overcome these issues, the introduced saliency pipeline is adapted in multiple ways. In chapter 4 the detection range is improved by substituting the saliency measure for a derivative-based measure. In chapter 5 the multi-axial features are removed by replacing the connectivity opening $\Gamma_{\mathbf{y}}$ by an importance-driven selection operator. The EDT ripples are suppressed by filtering the EDT itself using a minification filter, as is shown in chapter 6.

4 Derivative method

The saliency pipeline has proven to work well on 2D images, however directly applying the same technique on volumes has proven to be difficult. One of the problems of this pipeline is decreased sensitivity of the saliency metric near edges. This problem is further clarified in section 4.1. A new metric similar to the saliency metric is proposed in section 4.2 as a solution.

4.1 Corner and edge sensitivity

Although the saliency metric can remove cusps and dents in most situations, there are problems when these appear near corners of objects. The noise near corners is not detected as noise, and is subsequently not removed after performing the saliency pipeline. The branches of cusps near corners have decreased branch lengths, which in turn means that there is a shorter detection range. The further along the ligature-branch the distance transform increases, while ρ stays constant, meaning that the ligature branch requires a certain length before the ratio ρ/D goes below a certain threshold. This becomes a problem with cusps that have shorter branch lengths, as is depicted in figure 4.1. A single threshold c can remove most of the cusps, but is unable to remove noise near corners. Increasing c to accommodate this can have the unwanted side-effect of removing core branches as well.

In the 3D case, the same problem caries over near edges, where ligature sheets can be too small to be reliably detected. Additionally, it is possible that curvilinear 2-dimensional noise structures appear on the surface. They correspond to planar ligature sheets in the skeleton, for which some parts may also be very close to the core skeleton. It is thus important to even detect small ligature parts, so that it is possible to disconnect the ligature sheets in full from the core skeleton.

4.2 Derivative metric

Ligature sheets have special properties that can be used to distinguish it from core skeleton parts. Cusps correspond to curve segments from which a subsegment all share the same feature points. In the 2D case, if we follow the skeleton from the cusp to the core the geodesic distance between the feature points keep increasing until it reaches a skeleton point which contains the outer points of the cusp as the feature points. The further along the ligature-branch the ρ stays constant; therefore the directional derivative of the importance metric along the skeleton is zero. This is an important fact which can be used to detect shorter ligature branches. One proposal to perform this is:

$$\overline{\sigma}(\mathbf{x}) = \nabla\rho(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) \tag{4.1}$$

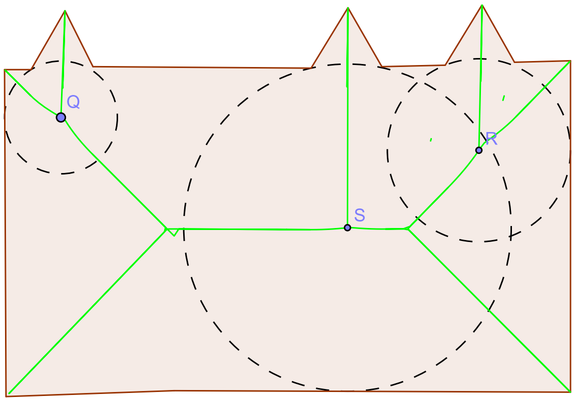


Figure 4.1: Shows equally-sized cusps on the boundary of a rectangle. Although these cusps all share the same importance, the shorter branch lengths inhibits the saliency pipeline to remove the cusps at the corners, as each branch has a maximum local thickness. Thus a different threshold on σ is required to remove each cusp.

where $\mathbf{v}(\mathbf{x})$ is a vector field which is tangent to $\mathcal{S}$. Unfortunately, this straightforward proposal is problematic when plateaus of importance exist which are not ligature branches or sheets. Take for example the center skeleton line of a rectangular box, although its importance is constant, it is not a ligature branch.

Improved detection can be attained by also including the distance transform in the equation. Ligature branches correspond to importance plateaus along skeleton sheets, but where the distance transform $D(\mathbf{x})$ also increases. Therefore a better proposal is the following equation:

$$\overline{\sigma}(\mathbf{x}) = \frac{\nabla\rho(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x})}{\nabla D(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x})} \tag{4.2}$$

Here the derivative of the importance is divided by the derivative of the distance transform. If both the importance and the distance transform only change slightly along the skeleton axis, it will not be detected as a ligature sheet, which is the desired effect. Only if the importance values stay stagnant and the distance transform varies largely a point will be detected as a ligature skeleton point.

Limit cases

Limit cases exists for which the computation of equation (4.2) requires some care. The most important case is when $\nabla D(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) \approx 0$ as a division by zero can occur. Recall that this does not occur with ligature sheets, as they increase in distance to the surface along its axes. Therefore, to mitigate this problem the following threshold rule is used:

$$\nabla D(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) \geq l \tag{4.3}$$

where l is a very small constant near zero to account for floating point imprecision. Points which do not satisfy this threshold are discarded as candidates for detection.

Detection

In the original saliency pipeline as explained in chapter 3 the detection is done by thresholding the saliency measure itself. In particular, the operator $\omega_\sigma(\mathcal{S})$ uses a threshold $c \in \mathbb{R}$ on σ to detect ligature sheets. Using the scheme on $\overline{\omega}$ is no longer meaningful, as it is invariant to the scale of the noise. Meaning, that the values of ligature branches will be near zero regardless of the size of the cusp or dent. To control for the size of the artifacts we wish to remove the following scheme is proposed:

Definition 9. *Let* $\mathbf{y} \in \mathbb{R}^n$ *be the skeleton point with the largest importance value so that* $\mathbf{y} = \operatorname{argmax}_{\mathbf{x} \in \mathcal{S}} \rho(\mathbf{x})$*, and let* $\Gamma(X)$ *be the connectivity opening as defined in chapter 2, then the derivative skeleton simplification is given by:*

$$\overline{\omega}_\sigma(\mathcal{S}) = \Gamma_{\mathbf{y}}(\{\mathbf{x} \in \mathcal{S} \mid \nabla D(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) > k \wedge \overline{\sigma}(\mathbf{x}) > k \wedge \rho(\mathbf{x}) \geq t\}) \tag{4.4}$$

Here, k is a small constant near zero to account for numerical precision and smooth cusp transitions, and $t \in \mathbb{R}$ is the threshold used for the original importance metric ρ. By introducing the importance threshold we can control for the size of artifacts we wish to remove. For example, in figure 4.1 we can remove all bumps using a single threshold.

The operators $\delta_c(X)$ and $\hat{\delta}_c(X)$ will work the same as in the previous pipeline, except that $\overline{\omega}_\sigma(\mathcal{S})$ is used instead of $\omega_\sigma(\mathcal{S})$. A small comparison of the behavior of the detection between the originally saliency pipeline is shown in figure 4.2. In this figure we can see that the detection range can be improved considerably by using the derivative-based measure.

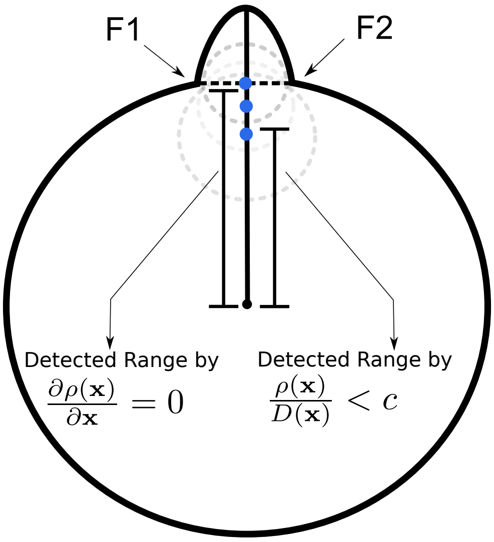

F1
F2
Detected Range by
$\frac{\partial \rho(\mathbf{x})}{\partial \mathbf{x}} = 0$
Detected Range by
$\frac{\rho(\mathbf{x})}{D(\mathbf{x})} < c$

Figure 4.2: Shows the detection ranges of the original saliency metric and the derivative of an importance measure. The shape is a circle with a singular bump on the surface; the corresponding skeleton is a A_1^2 curve segment with two A_3 endpoints. All skeleton points residing in the circle have the feature points F_1 and F_2. Detection of the derivative-based measure starts as soon as the importance measure stays constant, while the original saliency measure requires the ratio of ρ and D to be below c. Although c can be increased, within reason, to improve the detection range, setting it too high can also have the negative consequence of removing branches corresponding to features.

4.2.1 Choice of velocity field

The choice of velocity field $\mathbf{v}(\mathbf{x})$ is an important aspect when applying the proposed method on surface skeletons. For 2D skeletons the choice of velocity field is straight-forward, there are only two directions which are tangent to the skeleton field, for which the direction which is increasing in importance is the obvious choice. One proposal for the 3D case is:

$$\mathbf{v}(\mathbf{x}) = \nabla\rho(\mathbf{x}) \tag{4.5}$$

Substituting $\mathbf{v}(\mathbf{x})$ in equation (4.2) gives us:

$$\overline{\sigma}(\mathbf{x}) = \frac{|\nabla\rho(\mathbf{x})|^2}{\nabla D(\mathbf{x}) \cdot \nabla\rho(\mathbf{x})} \tag{4.6}$$

Note that choosing $\mathbf{v}(\mathbf{x}) = \nabla\rho(\mathbf{x})/|\nabla\rho(\mathbf{x})|$ will give the same result, as the length of the velocity field is a common factor in $\overline{\sigma}$. For surface skeletons, however, the choice is less obvious. Let M be the continuously differentiable manifold of the surface of the skeleton, and TM the tangent bundle of M. For 3D meshes the skeleton is locally planar, and therefore $T_{\mathbf{x}}M$ lie on a plane. This means that we have more freedom in the direction of $\mathbf{v}$ than on 2D skeletons. If we want to add the constraint to only allow directions increasing in importance, then $\mathbf{v}(\mathbf{x})$ is locally constrained on a half-plane for all $\mathbf{x}$. Therefore, we need additional constraints for $\mathbf{v}$ compared to the 1D case. The most straight-forward approach would be taking the gradient direction, as is defined in equation (4.5). Unfortunately this does not always give the desired effect. In ligature sheets corresponding to noisy cusps the importance is not constant across its axis, and therefore the gradient of these ligature sheets are not zero as in the 1D case. Although the directional derivative towards the skeleton root might be zero, the gradient orthogonal to this direction will likely not be. Therefore the gradient $\nabla\rho(\mathbf{x})$ is not a useful velocity field for surface skeletons.

Gradient of the EDT

Another straight-forward approach is to use the gradient of the Euclidean distance transform as the velocity field. Intuitively this would make sense as the gradient follows the interior of X, which often traces out (with gradient ascent) to a local maximum which is often close to the global root of importance. The proposal for V would be:

$$\mathbf{v}(\mathbf{x}) = \nabla D(\mathbf{x}) \tag{4.7}$$

However, there is no guarantee that the Euclidean distance transform is constrained on M (in 3-space). Indeed, the direction can be anywhere in X. As example, the ligature sheets corresponding to the spherical bumps on the surface have a gradient in the distance transform pointing partially to the interior of the ligature sheet, and thus are not constrained on M. Therefore naively evaluating equation (4.2) using finite differences can result in evaluating neighbors which have zero importance, and thus this method requires projection of the velocity field to M.

Unified skeleton model

Jalba et al. [29] introduced a unified multiscale framework for planar, surface, and curve skeletonization. In this thesis we refer this work as the Unified Skeleton Model (USM).

The USM method is a framework that detects all types of skeletons using a single model, and also produces a multiscale representation which allows to progressively simplify, or regularize, all skeleton types. It is a based on a contraction process which distributes density at unit length from the surface $\mathcal{M}$. The backbone of the method is described by a system of three PDE's:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{4.8}$$

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \phi = 0 \tag{4.9}$$

$$\mathbf{u} = \frac{\nabla \phi}{|\nabla \phi|} \tag{4.10}$$

Here t is the time parameter, ρ is the density as a function of t, and $\phi \in [-1, 1]$ is a field which is 1 inside the contracted shape, and -1 outside, and the field $\mathbf{u}$ gives the contraction direction. The boundary of the contracting shape is implicitly given by $Y_t = \{\mathbf{x} \in \mathbf{X} \mid \phi(\mathbf{x}, t) = 0\}$. The field $\mathbf{u}$ can be used as the velocity field for our method. And an added value for this, is that the USM takes several measures to minimize the amount of noise, which should improve the quality of any further computations that involve $\mathbf{u}$.

Feature-point driven velocity field

The feature points give important information of the inward flow from the boundary to the skeleton. In Reniers' work this is used to to find additional feature points using the normal of the tangent plane of the surface skeleton [49] (or rather he used a velocity field which is orthogonal to it). In a paper by Giblin and Kimia [21] a formal analysis of the feature points and the tangent plane of the surface skeleton is given. The most relevant result is repeated here.

In particular, let X be the continuously differentiable manifold of the surface of the model. Let Z be a Morge patch of M that is as part of the graph of a surface $z = f(x, y)$ parameterized by the xy plane at the origin, and tangent to M. The Euclidean distance transform can then be parametrized on this plane: i.e., $r(x, y) = D((x, y, f(x, y))^T)$. Let $(\overline{x}, \overline{y}, \overline{z})$ denote current coordinates in 3-space of a feature point ($\mathbf{a}^+ \in X$ or $\mathbf{a}^- \in X$). Then by construction the following equality holds:

$$(\overline{x} - x)^2 + (\overline{y} - y)^2 + (\overline{z} - z)^2 = r(x, y)^2 \tag{4.11}$$

The envelopes defined by the spheres around the origin point defines the surface boundary and the reconstruction of X (the continuously differentiable manifold of the surface of the model). The envelope of these spheres, as x and y vary, is given by equation (4.11) and its partial derivatives.

$$-2(\overline{x} - x) - 2(\overline{z} - f(x,y))\frac{\partial f(x,y)}{\partial x} = 2r(x,y)\frac{\partial r(x,y)}{\partial x} \tag{4.12}$$

$$-2(\overline{y} - y) - 2(\overline{z} - f(x,y))\frac{\partial f(x,y)}{\partial y} = 2r(x,y)\frac{\partial r(x,y)}{\partial y} \tag{4.13}$$

Evaluating at the origin ($x = y = f(x,y) = 0$) and assuming $f_x = f_y = 0$ gives the following simplified result:

$$\mathbf{a}^{\pm} = \left(-rr_x, -r_y r, \pm r\sqrt{1 - r_x^2 - r_y^2}\right)^T \tag{4.14}$$

In this result we see that the feature points $\mathbf{a}^{\pm}$ can be recovered solely by the local characteristics of the distance transform. If f_x and f_y are non-zero then one can rotate the Morge patch (or rather its coordinate system) so that the plane normal is parallel to the z-axis, which in case f_x and f_y are zero. Therefore the general solution is a rotation of equation (4.14), i.e., $\hat{\mathbf{a}}^{\pm} = R\mathbf{a}^{\pm}$, where R the rotation matrix.

Two important facts can be extracted from this result. First, it shows that the two features are on different sides of the tangent plane of the skeleton point, and therefore the tangent plane can indeed be used to find the second feature point by means of reflection. Second, the feature points can be projected on the tangent plane by simply taking the average of both feature points. Indeed we have that $1/2\mathbf{a}^{+} + 1/2\mathbf{a}^{-} = (-rr_x, -rr_y, 0)^T$, which is on to the xy plane. This result leads to the following proposal for the velocity field:

$$\mathbf{v}(\mathbf{x}) = R(rr_x, rr_y, 0)^T \tag{4.15}$$

or equivalently

$$\mathbf{v}(\mathbf{x}) = \mathbf{x} - 1/2\mathbf{a}^{+} - 1/2\mathbf{a}^{-} \tag{4.16}$$

Since $\mathbf{x}$ and $1/2\mathbf{a}^{+} - 1/2\mathbf{a}^{-}$ are on the tangent plane of $\mathbf{x}$, so is $\mathbf{v}(\mathbf{x})$. It follows that the proposed $\mathbf{v}(\mathbf{x})$ is always tangent to M. Since $R(rr_x, rr_y, 0)^T$ is in the increasing direction of the distance transform, it follows that it has similar properties as $\nabla D(\mathbf{x})$.

4.2.2 Inverse mapping

One important property of ligature branches or sheets is that they do not contribute in the reconstruction of the surface or boundary. For example the A_1^2 curve segment shown in figure 4.2 in the detected range by $\rho_{\mathbf{x}} = 0$ is redundant to the reconstruction. The points shown in this segment all have the same feature points in common; therefore inflating disks at these points all contribute to the same part of the boundary. This is an important property which could be useful to detecting ligature branches. An example is shown in figure 4.3.

4.3 Velocity field experiment

Since the quality of the detection is largely dependent on the choice of velocity field, the detection error of different fields are compared experimentally. This done

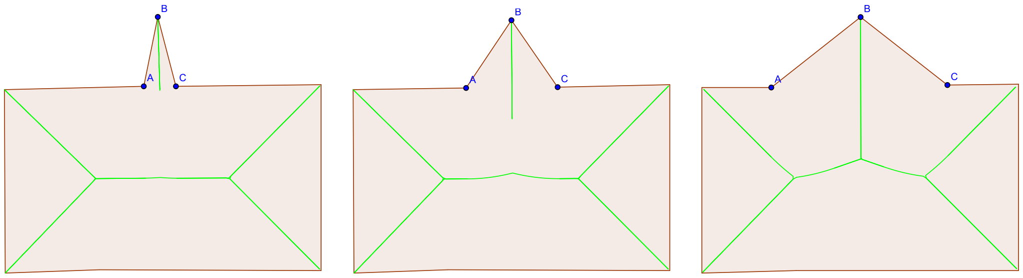
	$\boldsymbol{\nabla\rho}$ (4.6)		**Unified** (4.10)		**Feature** (4.16)		$\boldsymbol{\nabla D}$ (4.7)	
	$\upsilon(\mathcal{S}_{\mathbf{v}}, t)$	$\epsilon(\mathcal{S}_{\mathbf{v}}, t)$	$\upsilon(\mathcal{S}_{\mathbf{v}}, t)$	$\epsilon(\mathcal{S}_{\mathbf{v}}, t)$	$\upsilon(\mathcal{S}_{\mathbf{v}}, t)$	$\epsilon(\mathcal{S}_{\mathbf{v}}, t)$	$\upsilon(\mathcal{S}_{\mathbf{v}}, t)$	$\epsilon(\mathcal{S}_{\mathbf{v}}, t)$
cube	0.2506	0.2655	0.1565	0.1726	**0.1523**	**0.1686**	0.1556	0.1718
bunny	0.5094	0.5621	0.1840	0.3893	**0.1715**	**0.3820**	0.1689	0.3835
fandisk	0.5708	0.6057	0.2904	0.3741	**0.2751**	0.3684	0.2768	**0.3666**

Table 4.1: Estimated error of skeleton regularization of different velocity fields on a selection of models. These velocity fields are defined in section 4.2.1.

by regularizing the skeletons using the different definitions for the velocity field $\mathbf{v}$. For a selection of a few models convex noise is added. The models which are used in this experiment, and throughout this thesis are shown in figure 4.5. The regularized skeletons of these models using different velocity fields are compared to ground-truth skeletons. This allows us to qualitatively assess the viability of different velocity fields. Recall, the core idea of our work is that we want to remove noise from a shape by reconstructing a suitably regularized (simplified) version of its skeleton. Thus, the regularized skeleton (of the noisy shape) should be very similar to the skeleton of the shape without noise. We can hence test the quality of the regularization if we, avail of a shape without noise, by comparing its (clean) skeleton with the result of regularization on the skeleton of the same shape when noise was added.

On a original model X the foreground skeleton $\mathcal{S}$ is computed. Then a noisy version is created X' (by adding spherical creases), and its skeleton is regularized by the velocity field $\mathbf{v}$; define this regularized skeleton at threshold t as $\mathcal{S}_{\mathbf{v}(t)}$. The velocity fields $\nabla\rho$ and ∇D are approximated using central finite differences. For regularization the adjusted pipeline $\overline{\omega}_{\sigma}(\mathcal{S})$ is used as defined in equation (4.4). The importance metric is chosen as the geodesic measure $\rho(\mathbf{x}) = g(\hat{\mathcal{F}}(\mathbf{x}))$.

The errors between the original and the regularized skeleton are computed by



(a) Small bump (b) Medium bump (c) Large bump

Figure 4.3: Shows the inverse mapping behavior of the boundary surface to the skeleton. Non-salient skeleton points with $\partial\rho(\mathbf{x}) = 0$ do not contribute to the reconstruction and therefore are absent from the inverse mapping. Small bumps coincide with long non-salient curve segments while larger bumps do not.

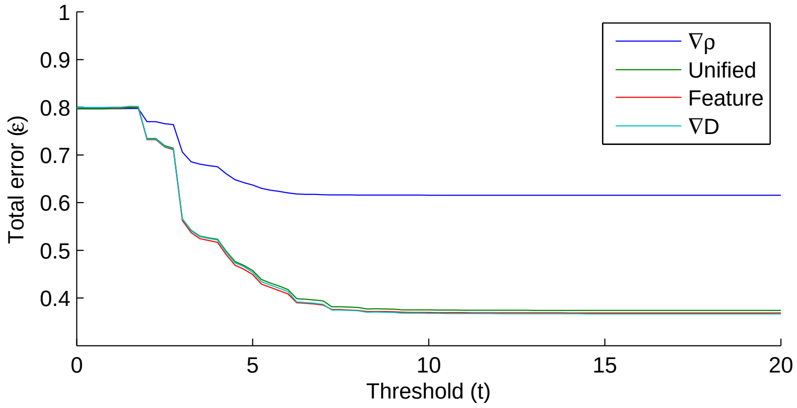
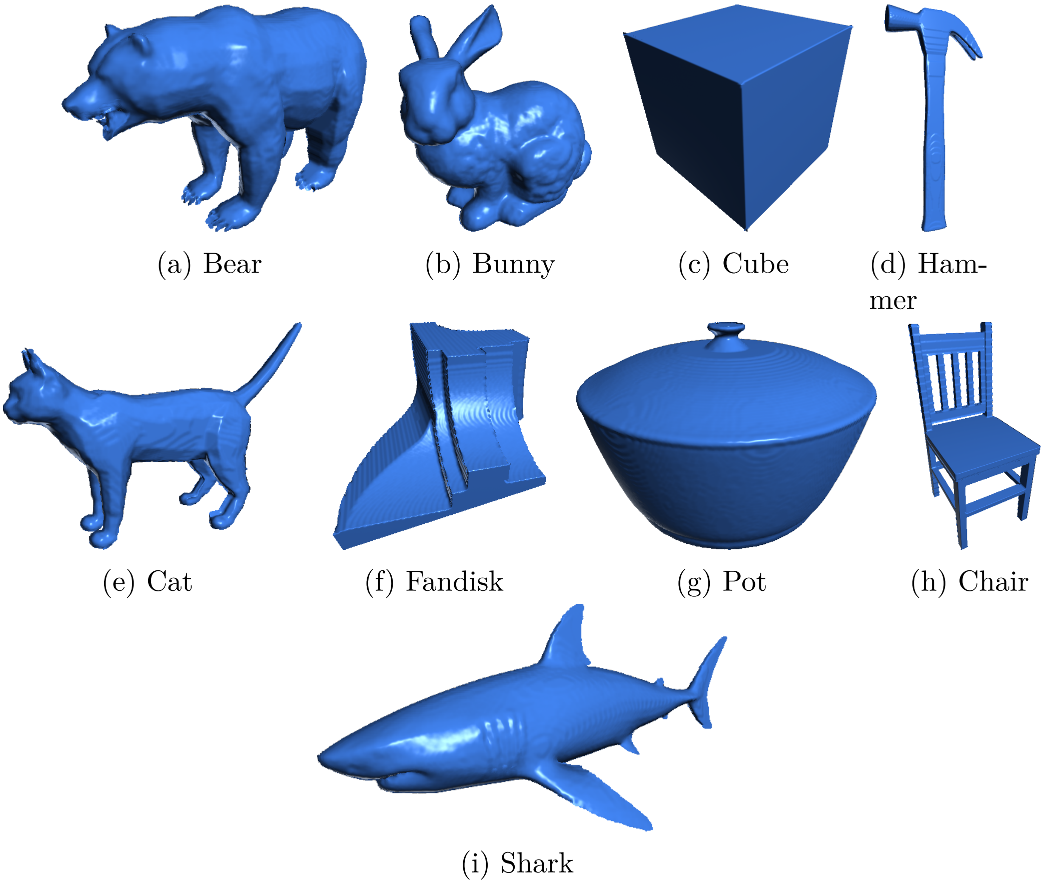


Figure 4.4: Comparison of the total error of different velocity fields as a function of t on the fandisk model. The result show that independent of the velocity field the error monotonically decreases as a function of the threshold, which implies robustness of the detection in terms of false positives.



(a) Bear (b) Bunny (c) Cube (d) Hammer

(e) Cat (f) Fandisk (g) Pot (h) Chair

(i) Shark

Figure 4.5: Collection of models which are used throughout this thesis.

the following two error measures:

$$\epsilon(\mathcal{S}_\mathbf{v}, t) = 1 - |\mathcal{S}_{\mathbf{v}(t)} \cap \mathcal{S}| / |\mathcal{S}_{\mathbf{v}(t)} \cup \mathcal{S}| \tag{4.17}$$

$$\upsilon(\mathcal{S}_\mathbf{v}, t) = |\mathcal{S}_{\mathbf{v}(t)} - \mathcal{S}| / |\mathcal{S}_{\mathbf{v}(t)}| \tag{4.18}$$

Respectively these are the negation of accuracy, and the false positive rate (type I error). Here, $\upsilon(\mathcal{S}_\mathbf{v}, t)$ gives the error of skeleton points contained in the regularized skeleton, but not in the original, which gives an indication of detection rate of ligature branches. The measure $\epsilon(\mathcal{S}_\mathbf{v}, t)$ gives the total error (type I error and type II error), i.e., all points in both sets that do not overlap divided by the total number of points. Both of these measures give a good indication of the performance of the velocity fields. If desired, a more fine-grained comparison could be done by computing, e.g., the Haussdorff distance between the two skeletons. However, for our purposes, the simpler expressions used here are deemed to be sufficient.

In figure 4.4 the total error $\epsilon(\mathcal{S}_\mathbf{v}, t)$ of the previously discussed velocity fields are compared as a function of the threshold t on the fandisk model. This figure shows that in all cases the total error decreases as a function of the threshold, which implies robustness of the measure in terms of false positives. Of the four velocity fields chosen the importance gradient stands out as the worst measure, while the other velocity fields give very similar results. This is not surprising, as the other fields (unified, feature and EDT) are all directly related to the Euclidean distance transform, and thus give similar gradient directions. Note that we can only compare the error relatively, as the assumption that $\mathcal{S} \subseteq \mathcal{S}_\mathbf{v}$ adds its own error to the measurements.

To give a better indication of the error the velocity fields are compared at the threshold $t = 20$ on multiple models (cube model, bunny model and the fandisk model). The result is shown in table 4.1. For each model, the lowest error for each of the two types is marked in bold. Here we can see that the feature-based approach has the lowest errors (by a small margin) on all but one model. On the fandisk model the gradient of the EDT performs slightly better. Again, we can see that the error of the gradient importance is significantly larger than other fields.

On the few models that were tested we can see that the feature-based method gives a slightly better result in terms of accuracy. However, if runtime performance is important the gradient of the Euclidean distance is a good choice, as it comes at a close second with no added performance penalty.

4.4 Qualitative comparison

In this section a qualitative comparison is shown of the derivative method compared to the original saliency metric. This is done by comparing the regularized skeletons, and by visually inspecting the reconstructions. The setup of the experiment is given in section 4.4.1, where the quantitative comparison is shown in section 4.4.2.

4.4.1 Setup

The effectiveness of the derivative based method is tested by performing the complete pipeline as is introduced in chapter 3 compared to the adjusted pipeline with the derivative-based metric. For the skeleton extraction the IMA-based method is used (as is introduced in section 7.1.1). In both cases the geodesic distance is used as the importance metric as introduced in section 3.1.2. In particular, the importance metric is chosen as $\rho(\mathbf{x}) = g(\hat{\mathcal{F}}(\mathbf{x}))$. Other performance metrics were experimented with, such as the unified model, but due to problems with its sensitivity to noise it could not be used as the importance metric. The reasoning of choice of the importance metric is further touched upon in chapter 7.

A small selection of models (cube, anvil, Stanford bunny, fandisk, chair) is used to make the comparison. In all of the models high amplitude noise is added, so that the differences in post processing is clearly visible. The noise model itself is clarified in chapter 7. The skeleton of the noisy model is created with the unified skeleton model [29]. For the saliency pipeline the threshold c is chosen as high as possible until side effects appear (when the core skeleton is degraded) for each model. For the derivative-based metric, the threshold is set to the maximum scale of the noise ($t = 20$) for all models. After the skeletons are simplified with both methods they are reconstructed, which allows us to judge effectiveness of the method by comparing how much noise is removed.

4.4.2 Comparison

The results are shown in figures 4.6 to 4.9. For each model the raw foreground surface skeleton is shown and the simplifications of the skeleton for both methods as well as their reconstructed results. Here we can see that the unprocessed skeletons are very large, and appear to have little structure that one would expect from a surface skeleton. This is because of large numbers of cusps created by the noise, which results in a large volume of ligature sheets which obstruct the core skeleton. Most of these ligature sheets are easy to remove with both methods. The most problematic ligature sheets are located near the edges. In these cases the ligature sheets are very small, which means that there is only a small range where the ligature sheet can be detected before connecting to the core skeleton.

In all examples the derivative filtering seems to filter more noise which are near edges. As example in figure 4.6 parts of ligature sheets are still connected to the core skeleton near strong edges when Saliency filtering is used, but most of them are removed with derivative-based filtering. This is something that one would expect because of improved detection range. The strongest difference is visible in figure 4.9. Large parts of the ligature tips are still connected to the core skeleton when the original saliency measure is used, but this problem does not exist with derivative-based filtering.

Yet the derivative filter does not remove all of the ligature sheets near edges. One example can be seen in figure 4.8, where ligature sheets are retained near the upper part of the fandisk. Although large parts of the ligature sheets are removed, the top parts of the cusps are retained. This is because these top parts directly connect with the core skeleton, so that the operator $\Gamma_{\mathbf{y}}$ cannot disconnect these ligature parts. To solve this other changes to the pipeline are proposed in chapter 5.

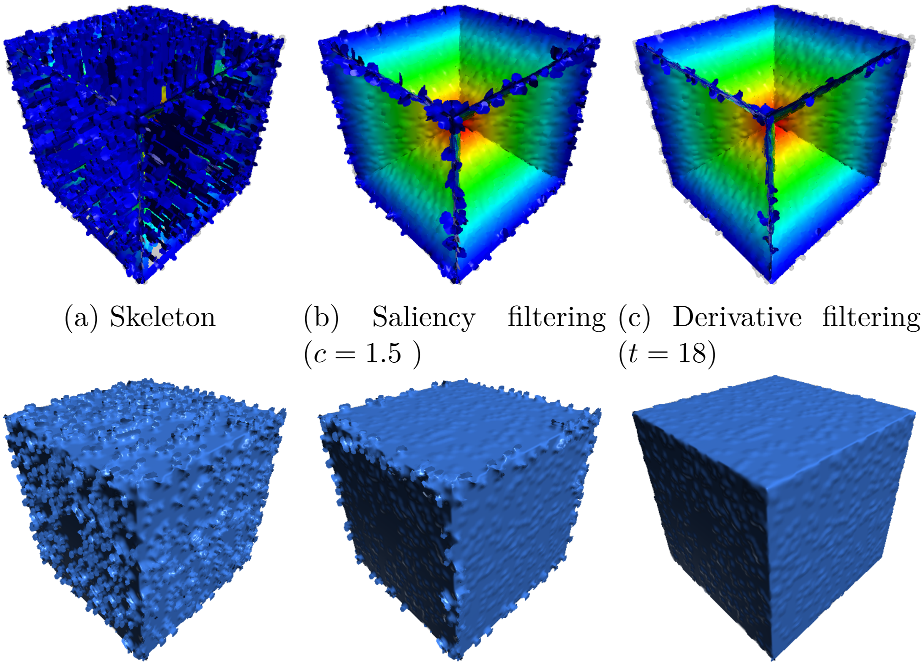
Even when the filtered skeleton is almost perfect, perturbations on the surface persist on the surface after reconstructing the filtered skeleton. This is especially visible on flat surfaces. This problem is caused by EDT ripples, for which several solutions are proposed in chapter 6.

4.5 Conclusion

In this chapter the problem of removing noise near edges with the existing saliency pipeline has been identified on surface skeletons. As a solution the derivative-based measure is proposed which has an improved detection range, in the sense that more noise is removed than the original salience measure proposed by Telea [58].

Variations of the method with different velocity fields were compared. From the tested velocity fields it was shown that the gradient of the importance performed the worst in terms of accuracy, while the EDT-gradient, the feature-based velocity field and the USM method all performed significantly better. From the small selection of models that were tested the feature-based method gave the best result with a small margin.

A comparison of the original saliency pipeline and the derivative-based pipeline has been made by visually comparing the regularized skeletons. Of the tested models, the proposed measure gives a significant improvement compared to the original saliency measure.



(a) Skeleton

(b) Saliency filtering ($c = 1.5$)

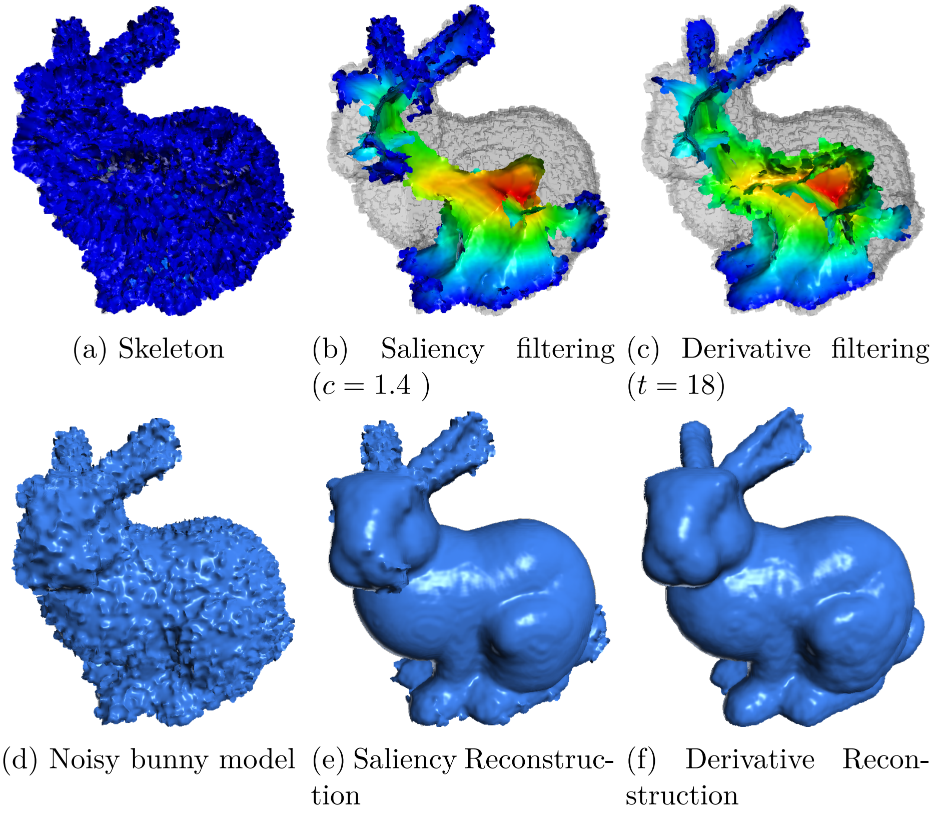
(c) Derivative filtering ($t = 18$)

(d) Cube model

(e) Saliency Reconstruction

(f) Derivative Reconstruction

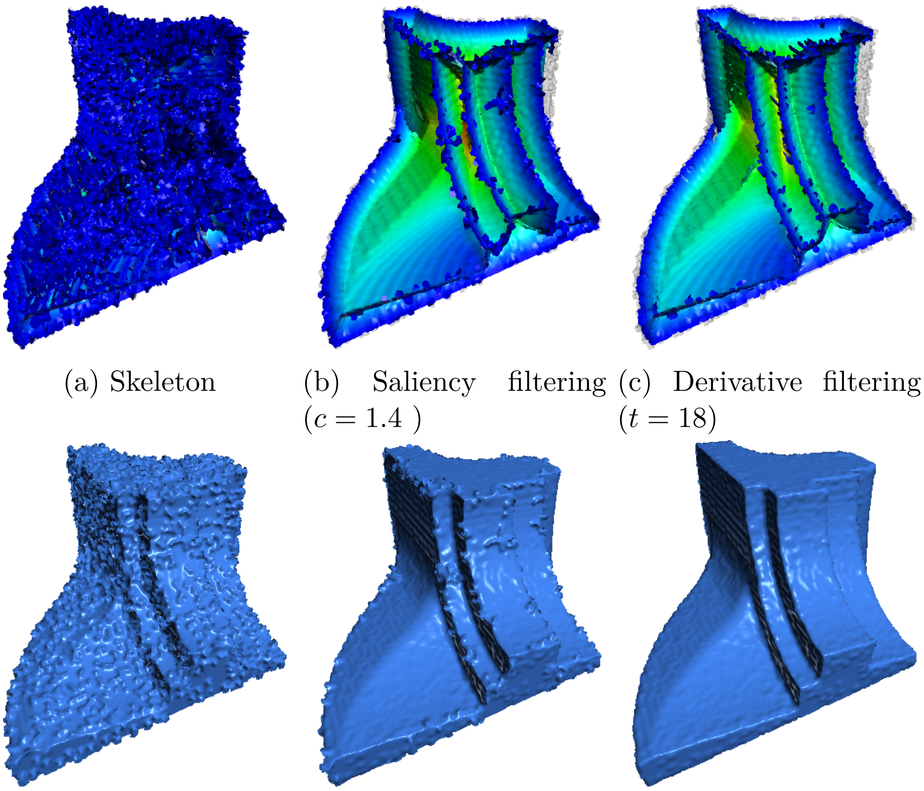
Figure 4.6: Cube model with convex 0-d noise. Skeletons are rainbow-colored using the importance metric ρ. With derivative filtering (f) more noise near edges is removed.



(a) Skeleton (b) Saliency filtering ($c = 1.4$) (c) Derivative filtering ($t = 18$)

(d) Noisy bunny model (e) Saliency Reconstruction (f) Derivative Reconstruction

Figure 4.7: Bunny model with 1-d spherical creases. Skeletons are rainbow-colored using the importance metric ρ. When the derivative filtering is used, more noise is removed while maintaining core skeleton sheets. However some noise near areas with large amount of curvature remains.



(a) Skeleton

(b) Saliency filtering ($c = 1.4$)

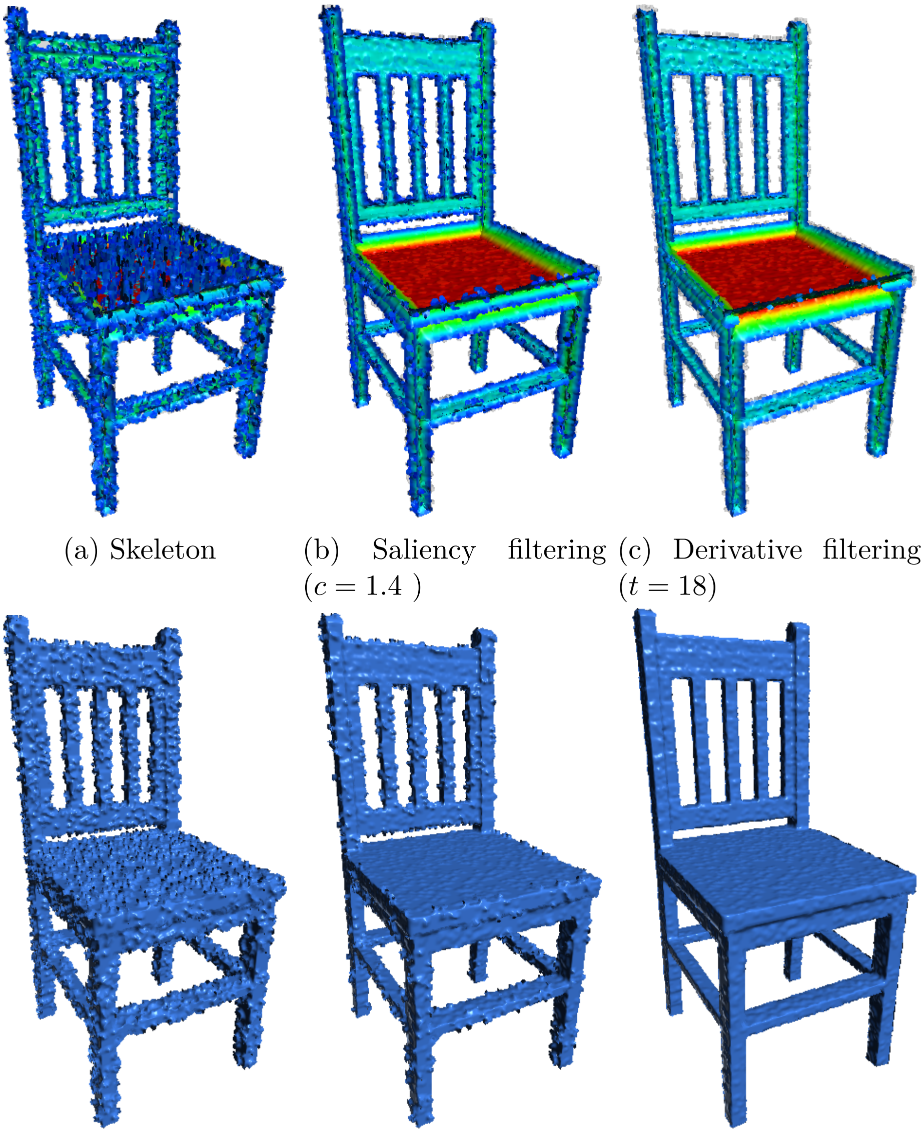
(c) Derivative filtering ($t = 18$)

(d) Fandisk model

(e) Saliency Reconstruction

(f) Derivative Reconstruction

Figure 4.8: Fandisk model with 1-d spherical creases. Skeletons are rainbow-colored using the importance metric ρ.



(a) Skeleton

(b) Saliency filtering ($c = 1.4$)

(c) Derivative filtering ($t = 18$)

(d) Chair model

(e) Saliency Reconstruction

(f) Derivative Reconstruction

Figure 4.9: Chair model with 0-d spherical creases. Skeletons are rainbow-colored using the importance metric ρ.

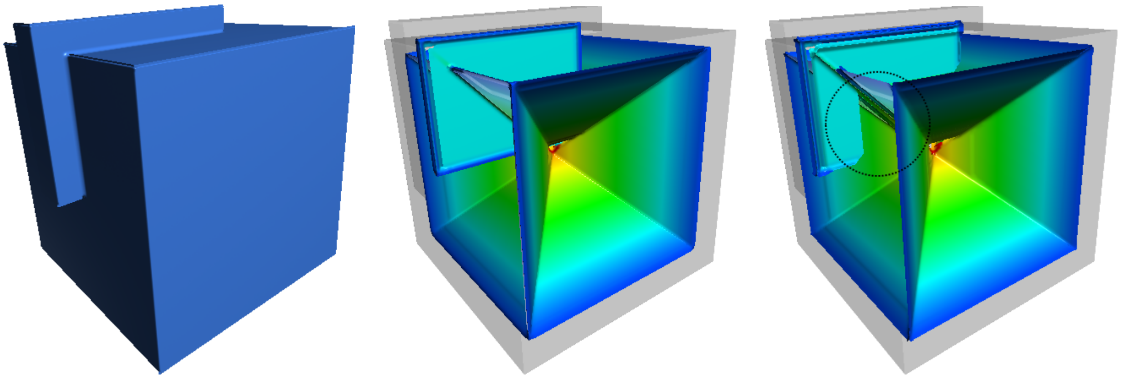
5 Saliency post-processing

In chapter 4, we have presented a pipeline which extends the original salience metric in [58] to remove noise on the surface of 3D shapes. However, as we have discussed there, and also visible in the examples presented at that point, our extended pipeline cannot fully remove all types of noise occurring on 3D shapes. In this chapter, we present several techniques that refine the noise detection and removal pipeline introduced in chapter 4, with the aim of capturing and removing additional noise configurations.

Unlike in the 2D case, it is possible to have so-called multi-axial features. These are multiple features or noises that overlap in different axes. When noise is conjoined with a feature in another axis it cannot be removed in the existing pipeline. Two solutions are presented which try to solve this problem. In section 5.1 the problems of multi-axial features are outlined and further clarified. Two related solutions are proposed in section 5.2 which are able to separate the multi-axial features. In in section 5.3, a qualitative and quantitative comparison is made between the two proposed methods and the derivative-based pipeline by adding long curvilinear noise stripes. This is done by comparing the regulated skeletons visually, as well as analyzing the error measurements. At the end of this chapter, a conclusion of is given in section 5.3.2.

5.1 Multi-axial features

Important features in one axis can overlap with noise in a different axis, so that the ligature sheet is orthogonal to the core skeleton. In this scenario, the ligature sheet corresponding to noise, can intersect with the core skeleton. See for example



(a) Cube and bar (b) Simplified skeleton (c) Saliency filtering ($c = 1.8$)

Figure 5.1: An example of multi-axial features. The skeleton sheet corresponding to the bar intersects with the core skeleton, and can therefore not be removed with the original saliency pipeline or the adapted derivative pipeline.

figure 5.1. An elongated bar is directly placed on a cube. The edges of the elongated bar are perpendicular to the edges of the sides of the cube. The bar serves as an exaggerated example of noise which is orthogonal to a feature (the cube's edge). The resulting skeleton of the elongated bar has a large rectangular sheet which intersects with the core skeleton of the cube. Suppose we would like to separate the elongated bar from the cube, then we would need to separate the rectangular sheet from the core skeleton. However, since they intersect it is not possible to separate them in full using any threshold of the classical saliency metric or even the improved derivative method.

In both cases the pipeline is dependent on the selection operator $\Gamma_{\mathbf{y}}$ to separate the core skeleton from the ligature sheets, however in this case the tips of the ligature sheets stay connected to the core skeleton. To separate these cases we require a different removal strategy. Multiple solutions are presented in section 5.2, and a comparison between the methods are given in section 5.3.

5.2 Solutions

In this chapter two methods are presented which can remove multi-axial features. The problems of multi-axial noises and features is that they leave a small set of points that stay connected between the core skeleton and ligature sheets, which prevents removal of ligature sheets in these cases. In both proposed methods, the solution is to create a new importance metric $\rho^+ \in \mathbb{R}$, based on the existing importance metric ρ and the derivative metric $\overline{\sigma}$. The idea is to create a monotonically increasing metric that assigns high importance values to the core skeleton (including the points with low ρ values), but low values to ligature sheets and endpoints connected to ligature sheets. Ideally, this would mean that thresholding ρ^+ results in feature-preserving hierarchical skeletons, gradually removing entire sheets in order of their size. This allows use to separate the intersecting ligature sheets, as they are assigned a local importance which is lower than the core skeleton.

5.2.1 Direct streamline filtering

From each point $\mathbf{x} \in \mathcal{S}$ a streamline or path can be traced out using the velocity field $\mathbf{v}(\mathbf{x})$ to a local root. Using local restrictions ($\overline{\sigma}$ and ρ) of this streamline we can assign an importance value to $\rho^+(\mathbf{x})$ that holds to the desired properties laid out above. The idea is trace the streamline until we reach a local root, or until a node which has a low derivative measure. Then the importance we assign to $\rho^+(\mathbf{x})$ is the importance of the last reached node.

Formally, the l-tuple $A = (\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_l)$ is called a path of length l if $\{\mathbf{a}_k, \mathbf{a}_{k+1}\} \subseteq \mathcal{C}$, for all $k \in [1, l-1]$. This is equivalent to saying that each subsequent pair of nodes should be connected according to its connectivity class (e.g. 26 connected in 3D). Given a path A in $\mathcal{S}$, we denote by $s(A)$ the set of its elements, so that $s(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_L) = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_l\}$. For our streamlines we add more restrictions to our paths:

Definition 10. *Let $\mathcal{S}$ be the foreground image and let $\mathbf{a}_1 \in \mathcal{S}$. Let $\mathbf{v}(\mathbf{x})$ be the*

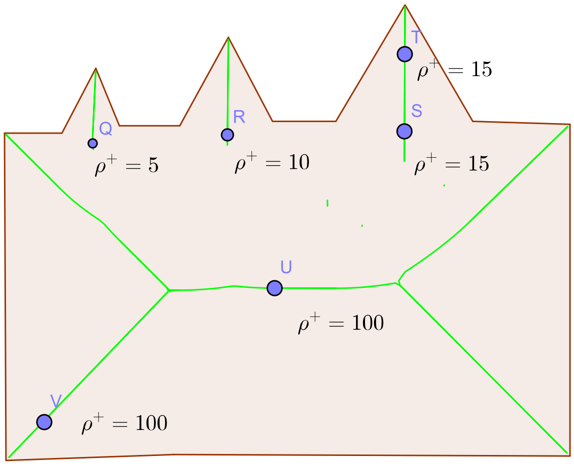


Figure 5.2: Example of the importance metric introduced in equation (5.2). As the importance values are traced out from local roots or low-derivative nodes, the importance values are constant across branches. The ligature branches are separated by low-derivative nodes from the core skeleton and therefore have lower importance values.

velocity field as defined in equation (4.16)*, then* $A(\mathbf{a}_1)$ *is said to be the path of* $\mathbf{a}_1$ *where its equation* $\mathbf{a}_{i+1}$ *is given as:*

$$\mathbf{a}_{k+1} = \underset{\mathbf{y}}{\operatorname{argmax}}\{\mathbf{v}(\mathbf{a}_k)\cdot(\mathbf{y}-\mathbf{a}_k) \mid \{\mathbf{a}_k,\mathbf{y}\} \subseteq \mathcal{C} \wedge \rho(\mathbf{y}) \geq \rho(\mathbf{a}_k) \wedge \overline{\sigma}(\mathbf{x}) > k\} \tag{5.1}$$

This means that subsequent nodes should have increasing ρ values, in addition to the connectivity requirement. From the candidates that are connected to $\mathbf{a}_k$ the node with lowest deviation from the velocity field $\mathbf{v}$ is chosen. If the set under the condition is empty, it means that $\mathbf{a}_k$ is a local root, or the derivative measure is below k. This means that the path largely follows the trajectory of a streamline of $\mathbf{v}$, but deviates when it crosses low-derivative nodes or the root. From this we can give the following definition for the importance field ρ^+:

Definition 11. *Let* $\mathbf{x} \in \mathcal{S}$*, and* $A(\mathbf{x})$ *the path of* $\mathbf{x}$*, then the importance of* ρ^+ *is given as:*

$$\rho^+(\mathbf{x}) = \max\{\rho(\mathbf{a}_k) \mid \mathbf{a}_k \in s(A(\mathbf{x}))\} \tag{5.2}$$
$$= \rho(\mathbf{a}_l) \tag{5.3}$$

Thus the new importance measure $\rho^+(\mathbf{x})$ gives the highest importance that is in the path of $\mathbf{x}$. If the path crosses a low-derivative node, the importance assigned will be the maximum ρ of the noise structure. If the path leads to the global root, the importance assigned will be of the core skeleton. An example is shown in figure 5.2. In this example the ligature branches have lower importance values. Therefore we can use a single threshold on ρ^+ to separate the ligature branches from the core skeleton, without using the selection operator $\Gamma_{\mathbf{y}}$. This is because the measure is monotonically increasing (definition 6) in this example.

We can also proof that the monotonically increasing property holds in the general case, independent of the velocity field, if the original importance ρ is also increasing.

Theorem 12. *Let ρ be a monotonically increasing measure (see definition 6), and let ρ^+ be the measure based on ρ as defined in equation* (5.2)*, then it follows that ρ^+ is also monotonically increasing.*

Proof. Let $\mathbf{q} = \operatorname{argmax}_{\mathbf{y}} \rho^+(\mathbf{y})$ be the root of $\mathcal{S}$. It must hold that at for all $\mathbf{x} \in \mathcal{S}$ there exists an increasing path $P \in \mathcal{S}^k$ with $k \in \mathbb{N}^+$ the path length, where $P_k = \mathbf{q}$. In the discrete case the following equality holds for any path $A(\mathbf{x})$:

$$\mathbf{a}_{k+1} = \underset{\mathbf{y}}{\operatorname{argmax}}\{\mathbf{v}(\mathbf{a}_k) \cdot (\mathbf{y} - \mathbf{a}_k) \mid \{\mathbf{a}_k, \mathbf{y}\} \subseteq \mathcal{C} \wedge \rho(\mathbf{y}) \geq \rho(\mathbf{a}_k) \wedge \overline{\sigma}(\mathbf{x}) > k\} \quad (5.4)$$

Suppose that the condition $\overline{\sigma}(\mathbf{x}) > k$ always holds, so that there are no ligature nodes. Then since ρ is monotonically increasing there must exist an increasing path to $\mathbf{q}$ for all $\mathbf{x} \in \mathcal{S}$. Since $\mathbf{a}_{k+1}$ is also monotonically increasing by design, we have that the following equality must hold $\mathbf{a}_l = \operatorname{argmax}_{\mathbf{y}}\{\mathbf{v}(\mathbf{a}_{l-1}) \cdot (\mathbf{y} - \mathbf{a}_{l-1}) \mid \rho(\mathbf{y}) \geq \rho(\mathbf{a}_{l-1})\} = \mathbf{q}$. Indeed, if the path of $A(\mathbf{x})$ does not cross derivative nodes it reaches $\mathbf{q}$. The elements $\mathbf{y} \in s(A(\mathbf{x}))$ all have the condition that $\rho^+(\mathbf{y}) = \rho(\mathbf{q})$. It follows directly that ρ^+ is monotonically increasing in paths with no ligature nodes.

If the condition $\overline{\sigma}(\mathbf{x}) > k$ does not always hold, a monotonically increasing path is chosen which does not cross any derivative nodes, if no such path exist the end-node $\mathbf{a}_l$ does not have to be $\mathbf{q}$. In this case the following equality still holds: $\rho^+(\mathbf{y}) = \rho(\mathbf{a}_l)$ for all elements $\mathbf{y} \in s(A(\mathbf{a}_l))$, so that $A(\mathbf{x})$ is monotonically increasing.

Since it holds that for all paths $A \in \mathcal{S}^k$ are monotonically increasing, it follows that ρ^+ is also monotonically increasing.

□

Note that the condition only holds if the importance measure ρ is also increasing.

5.2.2 Global streamline filtering

Directly evaluating equation (5.2) leads to an inefficient algorithm, as it requires tracing the velocity field for each $\mathbf{x} \in \mathcal{S}$ separately. This leads to an alternative formulation, which is very similar to the proposed streamlines, but is much faster to compute. Let $\mathcal{S}$ be a total ordered set on ρ of size l, so that $\mathbf{x}, \mathbf{y} \in \mathcal{S}$, $\mathbf{x} \leq \mathbf{y}$ implies that $\rho(\mathbf{x}) \leq \rho(\mathbf{y})$ and

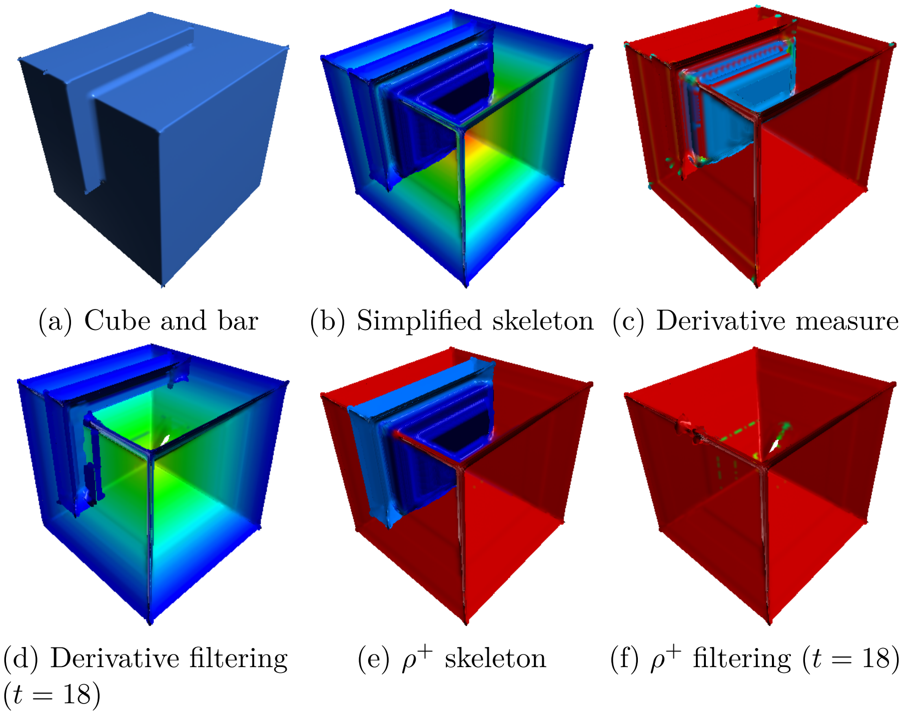
$$\mathcal{S}_1 \leq \mathcal{S}_2 \leq \mathcal{S}_3 \leq \cdots \leq \mathcal{S}_l \quad (5.5)$$

Define the new ρ^+ on the as the following recursion:

$$\rho^+(\mathcal{S}_i) = \begin{cases} \rho(\mathcal{S}_i) & \text{if } i = l \text{ or } \overline{\sigma}(\mathcal{S}_i) \leq k \\ \max(\{\rho^+(\mathcal{S}_j) \mid i < j \wedge \{\mathcal{S}_i, \mathcal{S}_j\} \subseteq \mathcal{C}\}) & \text{otherwise} \end{cases} \quad (5.6)$$

where $i, j \in \mathbb{N}^+$ and $i, j \leq l$. In other words the importance is the current ρ value at the root or low-derivative nodes. Otherwise, it is recursively defined as the largest assigned ρ^+ value of the neighboring nodes. The criterion $i < j$ ensures that we have a valid recursion, as new importance values can only be assigned in a strictly decreasing fashion.

The effect is very similar as the streamline filtering approach, where core skeleton parts are assigned the importance of $\rho(\mathbf{q}) = \rho(\mathcal{S}_l)$, but ligature sheets are assigned



(a) Cube and bar (b) Simplified skeleton (c) Derivative measure

(d) Derivative filtering ($t = 18$) (e) ρ^+ skeleton (f) ρ^+ filtering ($t = 18$)

Figure 5.3: An example of removal of intersecting sheets using the global streamline approach.

a local maximum. In fact, for the example in figure 5.2 both methods give the same output. For surface skeletons there are subtle differences, which are shown in section 5.3. The main advantage is that this formulation leads to a faster algorithm. We can compute a valid total ordered set $\mathcal{S}$ using radix sort, and then solve the recursion by assigning the importance values using equation (5.6) in reverse order starting from $\mathcal{S}_l$. This leads to an $O(l)$ algorithm, which is better than directly applying the streamline solution. Additionally, we do not need to trace a velocity field, which simplifies the implementation (although the velocity field is still needed for computing the derivative measure).

5.3 Comparison

Here we compare the two proposed methods by their ability to remove multi-axial features. First we can show that the cube and bar solution can be separated by both methods, as is shown in figure 5.3. Here, we can see that when derivative filtering is used, parts of rectangular sheets of the elongated bars cannot be removed, as they are connected to the core skeleton at the edges of the cube. However, when using the global streamline approach, the sheets belonging to the bar are assigned lower importance values than the core skeleton, and therefore they can be removed by thresholding ρ^+.

5.3.1 Line experiment

An experiment has been created to compare the following three methods to remove multi-axial noise:

1. The modified derivative pipeline introduced in chapter 4, where the operator $\Gamma_{\mathbf{y}}$ is used to separate the core skeleton from the ligature sheets.

2. The direct streamline filtering approach is introduced in section 5.2.1, where the importance is assigned by the paths that are traced to a local root. A threshold rule on the importance ρ^+ is used to separate the ligature sheets.

3. The global approach: similar as the streamline approach, but a recursive importance definition is used instead of velocity tracing. Just as the streamline approach the final step is to threshold the produced importance ρ^+ for separation.

On a selection of models 1D noise is added with long curvilinear stripes (using the parameters $p = 0.001, r = 3.0, h = 2, l = 50$). The noise model and its parameters are given in section 7.2. The long curvilinear stripes often cross the edges of the models (and thus produce intersecting sheets), and therefore are ideal for testing multi-axial separation. On all the methods the threshold $t = 18$ is used, as this is the maximum importance value of the ligature sheets of the curvilinear elements. The adapted IMA method (as given in section 7.1.1) is used to produce the skeletons, and the geodesic measure is used for the importance ρ. The three aforementioned methods are applied to regularize the skeletons, and their ability to remove the ligature sheets are compared both visually and by comparing the regularization errors. For the quantitative results, the same error metrics are used as introduced in section 4.3. In particular, $\epsilon(\mathcal{S}_{\mathbf{v}}, t)$ is the negation of accuracy, and gives an indication of the total error, and $\upsilon(\mathcal{S}_{\mathbf{v}}, t)$ is the type I error, which gives an indication of the false positive rate.

The qualitative results are shown in figures 5.4 to 5.6, and the regularization errors are shown in table 5.1. The skeletons are rainbow-colored by the importance metric of the used methods. Just as in chapter 4, low importance areas are colored blue, while high-importance areas are colored red. We can see that in case of the derivative metric the ligature sheets have low importance values, with exception of the tips of the ligature branches. Streamline-based methods assign low values to entire ligature sheets, including the tips. This advantage is visible in the resulting regularized skeletons, as the streamline methods remove much more ligature sheets corresponding to noise than the derivative-based pipeline. From the streamline methods, the direct method removes more noise than the global method, but unfortunately can also remove salient parts of the core skeletons. An example is shown in figure 5.6, the direct streamline method removes most of the noise, but the bars of the chair are removed as well. Both the global method and the derivative method keeps the core skeleton intact, where the global method removes more noise in this case. This is because the direct streamline method is more restrictive with the paths, and therefore is less likely to assign higher importance values if there are many low-derivative nodes (including false positives). This allows it to remove more

	Derivative(4.2)		**Streamlines**(5.2)		**Global**(5.6)	
	$\upsilon(\mathcal{S}_{\mathbf{v}}, t)$	$\epsilon(\mathcal{S}_{\mathbf{v}}, t)$	$\upsilon(\mathcal{S}_{\mathbf{v}}, t)$	$\epsilon(\mathcal{S}_{\mathbf{v}}, t)$	$\upsilon(\mathcal{S}_{\mathbf{v}}, t)$	$\epsilon(\mathcal{S}_{\mathbf{v}}, t)$
bear	0.0139	**0.6362**	**0.0136**	0.6393	0.0137	0.6370
bunny	0.0197	**0.6201**	**0.0192**	0.6219	0.0194	0.6203
cat	0.0185	**0.6268**	**0.0175**	0.6337	0.0180	0.6291
chair	0.1228	0.1911	0.1181	0.2213	**0.1181**	**0.1859**
fandisk	0.0327	**0.4538**	0.0330	0.4556	**0.0319**	0.4563
cube	0.1174	0.1264	0.1176	0.1258	**0.1152**	**0.1233**
pot	0.0085	**0.6791**	0.0084	0.6797	**0.0084**	0.6794

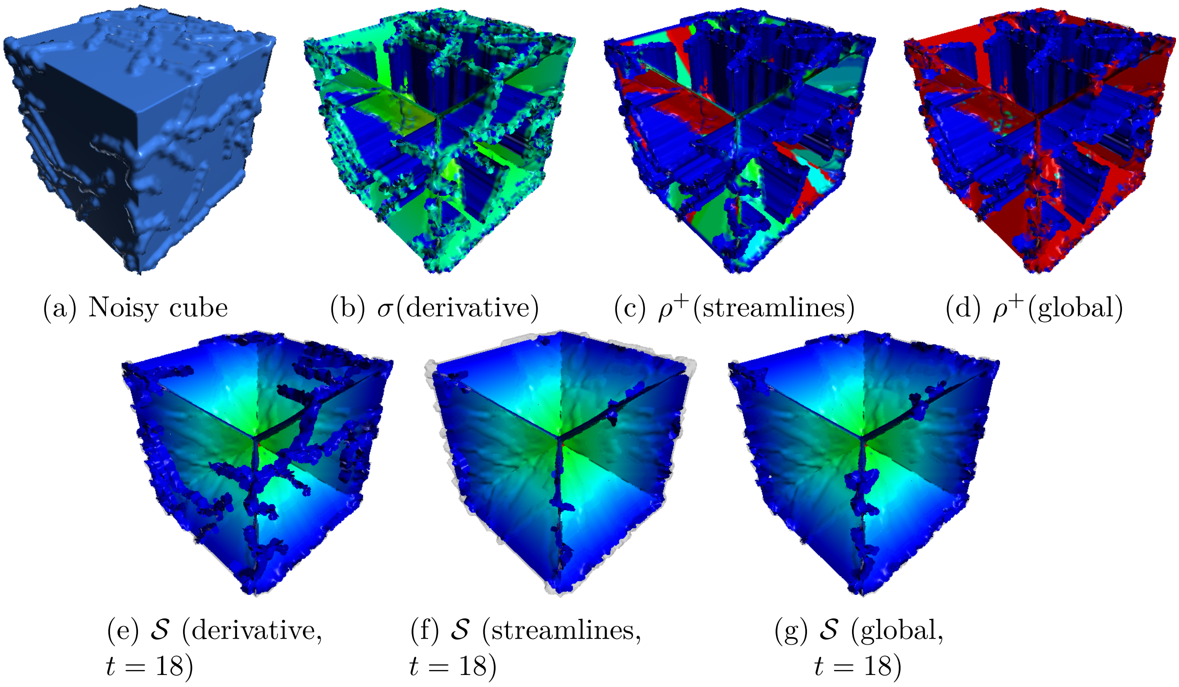
Table 5.1: Estimated error of different skeleton regularization methods on a selection of models with curvilinear noise. The lowest errors in their respective category and model are marked in bold.

noise, but also restricts its use. Overall, with the exception of the chair and cube model, the derivative model still has a lower regularization error, despite keeping some curvilinear noise intact. It is likely that this is because the derivative method is more conservative in removing core skeleton parts.

5.3.2 Conclusion

Two new methods have been proposed to remove multi-axial noise: direct streamline filtering and global streamline filtering. These methods create a new metric ρ^+ that assign high importance values to the core skeleton (including the point with low ρ values), but low values to ligature sheets and endpoints connected to ligature sheets. They have the advantage they no longer require the selection operator $\Gamma_{\mathbf{y}}$ as a step in the pipeline.

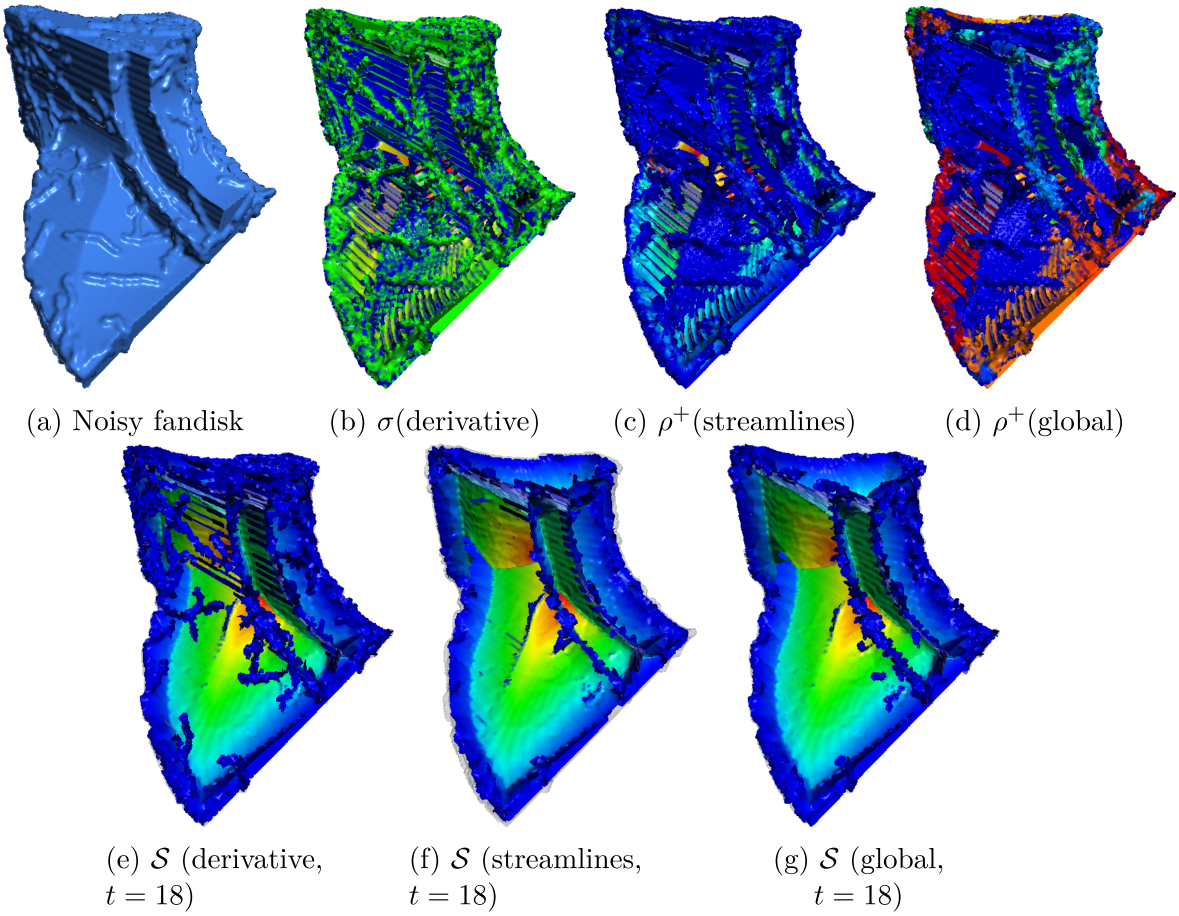
Both methods are successful in separating the sheets of the bar from the cube shown in the introduction. Although the direct streamline method removes more noise, it comes with the cost of affecting the core skeleton. From the tested examples the global streamline method seems to be more consistent than the direct streamline method, as well as removing more noise than thresholding the derivative measure. From the quantitative results we can see that the derivative measure still has a lower ϵ-error, despite removing less curvilinear noise. It is likely because the method is more conservative with removing core skeleton parts. However, we can see that for all the tested models the υ-error is considerably lower when the global streamline-based method is used, while the ϵ-error metric is only slightly worse than the derivative method. Since the global streamline method gives the best qualitative results and removes more noise, it is used for the final experiments which are presented further in chapter 8.



(a) Noisy cube (b) σ(derivative) (c) ρ^+(streamlines) (d) ρ^+(global)

(e) $\mathcal{S}$ (derivative, $t = 18$) (f) $\mathcal{S}$ (streamlines, $t = 18$) (g) $\mathcal{S}$ (global, $t = 18$)

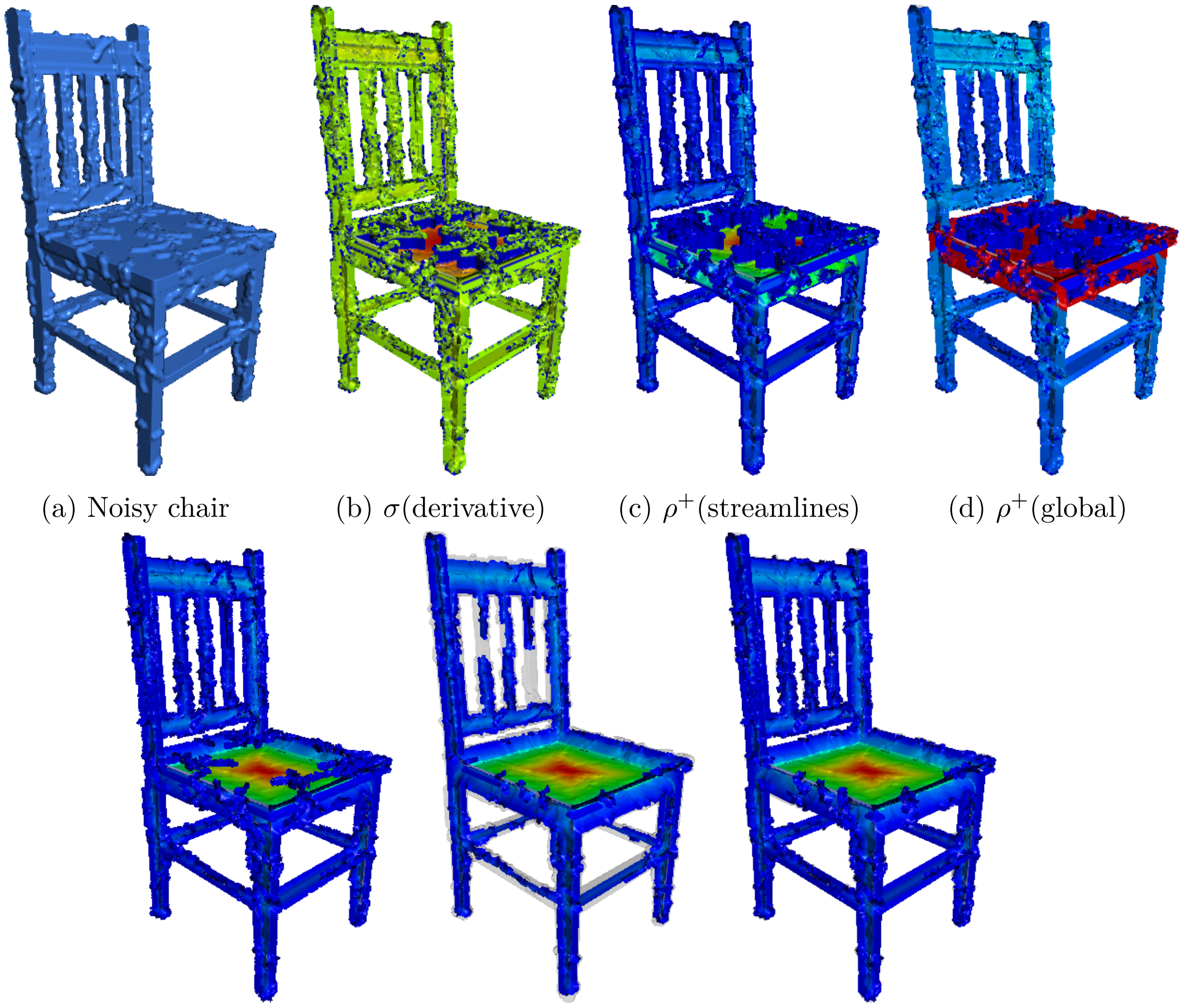
Figure 5.4: Removal of curvilinear elements on the cube model.



(a) Noisy fandisk (b) σ(derivative) (c) ρ^+(streamlines) (d) ρ^+(global)

(e) $\mathcal{S}$ (derivative, $t = 18$) (f) $\mathcal{S}$ (streamlines, $t = 18$) (g) $\mathcal{S}$ (global, $t = 18$)

Figure 5.5: Removal of curvilinear elements on the fandisk model.



(a) Noisy chair

(b) σ(derivative)

(c) ρ^+(streamlines)

(d) ρ^+(global)

(e) $\mathcal{S}$ (derivative, $t = 18$)

(f) $\mathcal{S}$ (streamlines, $t = 18$)

(g) $\mathcal{S}$ (global, $t = 18$)

Figure 5.6: Removal of curvilinear elements on the chair model.

6 EDT post-processing

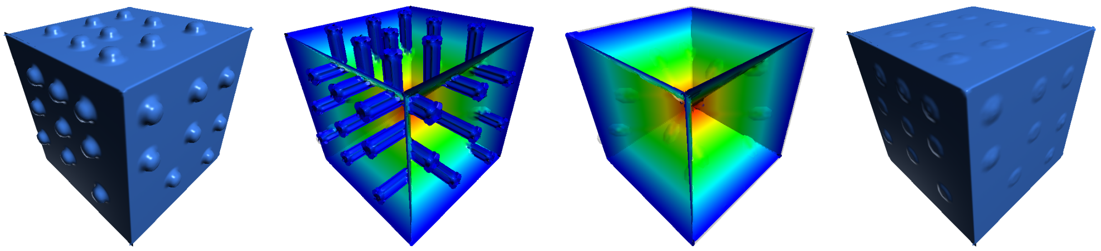
Even after removing all ligature sheets, small noise on the reconstructed surface still persists due to perturbations on the core skeleton and the EDT. Although these perturbations are of low amplitude, they are very visible on 3D models if any shading is used. Thus even after filtering the skeleton so that all ligature sheets are removed, it is not enough to ensure full removal of the noise in the reconstruction. The solution is to add a new step right before the reconstruction step(s) in the pipeline where we filter the Euclidean distance transform. The problem is further clarified in section 6.1. Several methods are proposed to remove the noise in section 6.2, each with their own advantages and disadvantages. In section 6.3 a comparison of the different methods is made and evaluated.

6.1 Problem

The problem is detailed in figure 6.1. After removal of the ligature sheets ‘ripples’ of the bumps persist in the reconstruction. This is because of valleys in the EDT of the core skeleton at parts that are closest to the bumps. In addition there is also slight bending of the core skeleton towards the bumps. The bending occurs since the skeleton is centered in the noisy shape. Thus, when the shape’s boundary has a bump (and there is no bump on the opposite side of the boundary), the skeleton needs to be slightly bended towards the bump so that it is centered. The result is that small ripples exist on the surface of the core skeleton as well as in the EDT, and they leave a mark in the reconstruction.

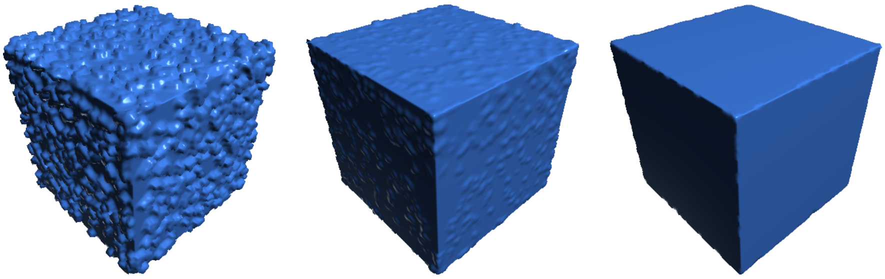
6.1.1 Skeleton smoothing versus EDT filtering

The immediate question is whether we are required to unbend the skeleton, filter the EDT values of the skeleton, or both. This question is partially answered in the following experiment. First we compute the original skeleton and EDT of a model.



(a) Bumpy cube (b) Skeleton (c) Regularized skeleton (d) Reconstruction

Figure 6.1: Demonstration of the small perturbations of the core skeleton and the EDT that persist in the reconstruction of the regularized skeleton.



(a) Noisy cube (b) Reconstruction $(\mathcal{S} + \hat{D})$ (c) Reconstruction $(\hat{\mathcal{S}} + D)$

Figure 6.2: A comparison of how much filtering the EDT or skeleton affects the reconstruction. This demonstrates that regularizing the EDT has a far greater effect than smoothing the skeleton if the skeleton has been regularized.

Then add noise to this model, and compute the noisy skeleton and the noisy EDT field. We then compare the two following scenarios:

1. Reconstruct the model with the original EDT field D but with the regularized noisy skeleton $\hat{\mathcal{S}}$.

2. Reconstruct the model with the noisy EDT field $\hat{D}$, but with the original skeleton $\mathcal{S}$.

The regularized skeleton of the noisy model is used so that is very similar to the original skeleton, apart from the slight bending of the skeleton due to noise. We can then test what method produces the least amount of noise: the noisy EDT with the original skeleton **or** the original EDT with the regularized noisy skeleton. This tests how much smoothing the skeleton should matter compared to filtering the EDT after skeleton post-processing.

This experiment has been performed on the cube model. The original skeleton $\mathcal{S}$ has been produced by using the IMA method, where no thresholds are used to regularize it. The noise which is applied to the cube model is created using the parameters $p = 0.02$, $r = 2$, $h = 3$, $l = 5$ using the method introduced in section 7.2. The noisy skeleton $\hat{\mathcal{S}}$ has been produced by using the IMA method on the noisy cube model, and then the global streamline filtering pipeline is applied with the threshold $t = 18$ to regularize the skeleton. Therefore this experiment simulates the same condition prior to the reconstruction step in the saliency pipeline, where we have a perturbed regulated skeleton and a noisy EDT field. The result is shown in figure 6.2. From this example the result is clear, at this noise scale level it is far more effective to post-process the EDT than smoothing the skeleton. As replacing the regulated skeleton by the original skeleton simulates the skeleton smoothing process, and replacing the EDT field by the original EDT field simulates the EDT post-processing step. Here we can see that using the regularized skeleton, but with the original EDT gives the better result. Furthermore, smoothing the skeleton is something far from easy, since it consists by a complex set of intersecting manifolds. Since only smoothing the EDT already gives quite good results, the methods introduced in section 6.2 are based on post-processing the EDT.

6.2 Methods

Here several methods are proposed that try to remove the EDT 'ripples'. These methods are subdivided into three categories:

1. Kernel-based filtering : Methods that apply a kernel on a local neighborhood of points to smooth the EDT, such as convolution filters, but also rank based filters.

2. Least squares projection: Use a local neighborhood to project points a on a local least-squares plane.

3. Flat projection: for each skeleton point, based on a local neighborhood of ligature sheets, explicitly try to find the removed parts of the surface and reduce the EDT so that enveloping spheres do not intersect with the removed parts, and therefore flattening the surface.

These methods are respectively explained in sections 6.2.1 to 6.2.3. The final evaluation and comparison of the methods are done in section 6.3.

6.2.1 Kernel-based filtering methods

Multiple kernel-based methods are proposed to filter the EDT. In this context we define a kernel $\mathcal{K}(S)$ as the mapping $\mathcal{K} : \mathcal{P}(D) \mapsto \mathbb{R}$, where, in this context, D is the multiset of the original distance transform values. The mapping operates on subsets of the distance transform, and outputs a filtered distance transform. The kernel $\mathcal{K}$ can be a rank-based filter such as the infimum, supremum, or median, but is can also be the local mean. A distinction is made of two kernel-based methods, i) implicit: which uses only the distance transform as input, explicit: which makes only use of the distance values of the regulated skeleton. The idea of the explicit method is that only the core skeleton points should contribute to the EDT filtering, as they should be less affected by the noise compared to ligature points.

Definition 13. *Let X be the foreground image, and let D be the original distance transform, and let $\mathcal{K}$ be the kernel and $r \in \mathbb{R}$ the radius, then the implicit kernel method is given by:*

$$\bar{D}_{\mathcal{K}}(\mathbf{x}) = \begin{cases} \mathcal{K}(\{D(\mathbf{y}) \mid \mathbf{y} \in X \wedge ||\mathbf{y} - \mathbf{x}|| \leq r\}) & \textit{if } \mathbf{x} \in X \\ 0 & \textit{otherwise} \end{cases} \tag{6.1}$$

The condition $\mathbf{x} \in X$ is required to force anti-extensiveness, so that no new elements are added after filtering. Similarly the explicit kernel method is given by substituting the foreground image by the regularized skeleton $\mathcal{S}$, which is defined as $\hat{D}_{\mathcal{K}}(\mathbf{x})$. This enforces that only the distance values of skeleton points are used in the kernel-based filtering method. Using this framework, the following kernels are used:

1. Mean: $\bar{D}_E(\mathbf{x})$, where E gives the sample mean.

2. Median: $\bar{D}_{\text{med}}(\mathbf{x})$, where med gives the median of the multiset.
3. Constrained opening: the infimum $\bar{D}_{\text{inf}}(\mathbf{x})$ followed by the supremum $\bar{D}_{\text{sup}}(\mathbf{x})$.
4. Minification: infimum $\bar{D}_{\text{inf}}(\mathbf{x})$ followed by inflation correction. The operation is defined by:

$$\text{minf}(\mathbf{x}) = \begin{cases} \bar{D}_{\text{inf}}(\mathbf{x}) + r & \text{if } D(\mathbf{x})) \geq r \\ \bar{D}_{\text{inf}}(\mathbf{x}) & \text{otherwise} \end{cases}$$

All of the above kernels are implemented in both the implicit and explicit version (resp. $\bar{D}_{\mathcal{K}}(\mathbf{x})$ and $\hat{D}_{\mathcal{K}}(\mathbf{x})$). Using the infimum will remove almost all of the left-over convex noise, but will also shrink the model. Both the supremum and the inflation correction are steps which try to reduce the shrinkage. The infimum followed by the supremum (method 3) makes sense in a morphological setting, as the resulting operator will be an opening. The inflation correction (method 4) makes use of the following bound $||\nabla D(\mathbf{x})|| \leq 1$, thus in the worst case the distance field is reduced by $r||\nabla D(\mathbf{x})||$, which is bounded by r. To correct this the radius r is added if $D(\mathbf{x}) \geq r$.

6.2.2 Least squares projection

Least squares projection can be a helpful tool in denoising. The EDT is spatially related with the surface skeleton; it can be seen as a fourth coordinate. We can define the mapping $\mathcal{P} : \mathcal{S} \mapsto \mathbb{R}^4$ as

$$\mathcal{P}(\mathbf{x}) = (x_1,\, x_2,\, x_3,\, D(\mathbf{x}))^T \tag{6.2}$$

The idea is to fit for each skeleton point $\mathbf{x} \in \mathcal{S}$ the least-squares 2-dimensional plane in a local neighborhood of mapped skeleton points $\mathcal{P}(\mathbf{y})$, $\mathbf{y} \in \mathcal{S}$ where $||\mathbf{y} - \mathbf{x}|| \leq r$ for some radius $r \in \mathbb{R}$. This is done by finding the eigendecomposition of the local covariance matrix of these points, and then truncating the two lowest eigenvalues. The skeleton points and their EDT values are projected on their local least-squares plane to perform denoising.

In particular let P_{ij} be the i'th neighboring point ($i = 1, ..., n$) on the j'th coordinate ($j = 1, ..., 4$). These points can be arranged into n column vectors, each with 4 entries, with the 4×1 column vector giving the i'th point of all variables being denoted $\mathbf{P}_i$. Define $\boldsymbol{\mu}$ as the column-wise average of P so that $\boldsymbol{\mu} = \sum \mathbf{P}_i/n$. Let $\mathbf{Y}$ be a $4 \times n$ matrix with the same notation as $\mathbf{P}_i$, and define it as mean-centered version of $\mathbf{P}_i$ so that $\mathbf{Y}_i = \mathbf{P}_i - \boldsymbol{\mu}$ Then the covariance matrix of $\mathbf{P}$ is given by $\mathbf{Y}\mathbf{Y}^\top$. Since the covariance matrix is symmetric, the matrix is diagonalizable, and the eigenvectors can be normalized such that they are orthonormal:

$$\mathbf{Y}\mathbf{Y}^\top = \mathbf{W}\mathbf{D}\mathbf{W}^\top$$

where $\mathbf{W}$ is the square 4×4 matrix whose j'th column is the eigenvector of $\mathbf{Y}$, and $\mathbf{W}$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues, i.e., $W_{jj} = \lambda_j$. Keeping only the first 2 principal components, produced by using only the largest 2 eigenvectors, gives the truncated transformation

$$\hat{\mathbf{x}} = \mathbf{x}\mathbf{W}_2 \tag{6.3}$$

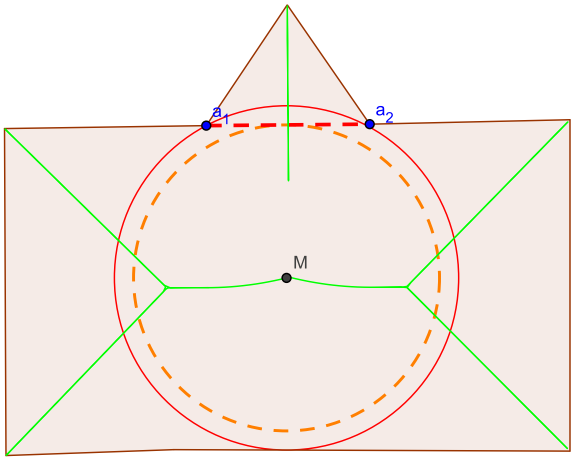


Figure 6.3: Shows the effect of flat projection of the EDT on the inscribed circle and its feature points $\mathbf{a}_1$ and $\mathbf{a}_2$. The red circle shows the original radius, while the dashed orange circle shows the reduced radius where the circle no longer intersects with the segment $\overline{\mathbf{a}_1\mathbf{a}_2}$.

where $\mathbf{W}_2$ is the truncated matrix containing the two largest eigenvectors, so that the projected vector $\hat{\mathbf{x}}$ now has only 2 columns. The least-squares projection filtering method is then given by transforming the truncated vector back, and taking its fourth coordinate, which is the projected EDT value. The complete transformation is given as:

$$\mathcal{D}_{ls}(\mathbf{x}) = (\mathbf{W}_2(\mathbf{x} - \boldsymbol{\mu})\mathbf{W}_2 + \boldsymbol{\mu})_4 \tag{6.4}$$

6.2.3 Flat Projection

The previously introduced methods try to regularize the EDT by generic smoothing methods using the local EDT and or the regularized skeleton points. However, we can also extract information from the ligature sheets of the noise we want to remove. In particular, the feature points of the ligature sheets give the parts of surface we wish to be deleted. The idea of the flat projection method is constrain the EDT by the line segments between the feature point pairs of the ligature sheets, so that the inscribed spheres do not intersect with these segments.

A demonstration is shown in figure 6.3. The current inscribed circle at point M intersects with a line segment between the feature points of a ligature branch, therefore a round bump will still be visible in the reconstruction. The radius of the dashed orange inscribed circle is reduced so that it no longer intersects with the segment. Note that in this particular case the reduced orange circle no longer touches the bottom of the rectangle, so that a dent will be visible in the reconstruction. This is because the core skeleton is not centered. However, one would find that in the average case just reducing te EDT is enough, as typical noise will be on both sides of the surface, so that the core skeleton centered on average.

This method is not explained in full detail, as its results seem to be poor in comparison with the other methods.

	$\bar{D}_E(\mathbf{x})$		$\text{minf}(\mathbf{x})$		$\bar{D}_{\text{sup}}\bar{D}_{\text{inf}}(\mathbf{x})$		$\bar{D}_{\text{med}}(\mathbf{x})$	
	Im.	Ex.	Im.	Ex.	Im.	Ex.	Im.	Ex.
bear	0.0325	**0.0129**	**0.0758**	0.1151	0.0481	**0.0145**	0.0359	**0.0134**
bunny	0.0272	**0.0161**	**0.0702**	0.1032	0.0366	**0.0154**	0.0305	**0.0171**
cat	0.0479	**0.0268**	**0.0862**	0.1162	0.0749	**0.0294**	0.0509	**0.0276**
chair	0.0516	**0.0489**	**0.0132**	0.1906	0.1015	**0.0453**	0.0736	**0.0499**
fandisk	0.0184	**0.0183**	**0.0216**	0.0955	0.0305	**0.0151**	0.0245	**0.0190**
hammer	0.0734	**0.0329**	**0.0557**	0.1835	0.1314	**0.0279**	0.0856	**0.0368**
cube	**0.0076**	0.0171	**0.0008**	0.0715	**0.0002**	0.0141	0.0242	**0.0170**
pot	0.0097	**0.0047**	**0.0277**	0.0306	0.0100	**0.0043**	0.0130	**0.0047**

Table 6.1: Comparison of the reconstruction errors of implicit $\bar{D}_{\mathcal{K}}(\mathbf{x})$ and explicit $\hat{D}_{\mathcal{K}}(\mathbf{x})$ kernel-based filtering methods. With the exception of the minification filter, explicit filters perform better than implicit methods.

6.3 Evaluation

To evaluate the different methods they are compared both visually and by comparing the reconstruction errors, which are computed by a 'difference' function (given below) of the original models and the reconstructed models.

On a selection of models 0D noise is added using the parameters ($p = 0.014, r = 2.1, h = 3, l = 1$). The noise parameters are set lower than other experiments so that we only evaluate the removal of noise ripples; not the removal of ligature sheets. The adapted IMA method is utilized to compute the surface skeletons. For all the methods the global streamline method is used for skeleton regularization, as explained in chapter 5. The threshold $t = 18$ is used as it removes most ligature sheets without any side-effects. All of the EDT regularization methods us a local neighborhood radius $r = 3$. This threshold was chosen as it gave the cleanest visual result with the least amount of feature distortion on most methods.

To compare the methods quantitatively the following error metric is used.

Definition 14. *Let $\mathcal{S}_{\mathbf{v}(t)}$ be the regularized skeleton as a function of t as defined in section 4.2.1, and let $\mathcal{X}$ the original model without noise. Then the reconstruction error is given as:*

$$\hat{\epsilon}(\mathcal{S}_{\mathbf{v}(t)}, t) = 1 - |R(\mathcal{S}_{\mathbf{v}(t)}) \cap \mathcal{X}| / |\mathcal{S}_{\mathbf{v}(t)} \cup \mathcal{X}| \tag{6.5}$$

This is almost identical to the error metric introduced in equation (4.18), but where the reconstructed models are compared to the original, instead of comparing the regularized skeletons to the original skeleton.

The rest of this chapter is organized as followed. First, a comparison is made of the implicit and explicit kernel methods in section 6.3.1. Second, all the relevant methods are compared both quantitatively and by comparing the models visually in section 6.3.2. Finally, a conclusion is given in section 6.3.3.

	$\text{minf}(\mathbf{x})$	$\hat{D}_E(\mathbf{x})$	$\hat{D}_{\text{sup}}\hat{D}_{\text{inf}}(\mathbf{x})$	$\hat{D}_{\text{med}}(\mathbf{x})$	$\mathcal{D}_{ls}(\mathbf{x})$	**Flat**
bear	0.0758	**0.0129**	0.0145	0.0134	0.0364	0.1354
bunny	0.0702	0.0161	**0.0154**	0.0171	0.0354	0.1092
cat	0.0862	**0.0268**	0.0294	0.0276	0.0496	0.1776
chair	**0.0132**	0.0489	0.0453	0.0499	0.0906	0.1107
fandisk	0.0216	0.0183	**0.0151**	0.0190	0.0456	0.0569
hammer	0.0557	0.0329	**0.0279**	0.0368	0.0606	0.1809
cube	**0.0008**	0.0171	0.0141	0.0170	0.0386	0.0095
pot	0.0277	0.0047	**0.0043**	0.0047	0.0202	0.0433

Table 6.2: Comparison of the reconstruction errors of the most relevant methods. The implicit mean, median opening, and the explicit min are omitted as they perform worse than their explicit (resp. implicit) counterparts.

6.3.1 Implicit versus explicit kernel methods

All of the proposed kernel methods can be implemented in a implicit manner, where all the neighboring distance values are used, or in a explicit manner, where only the distance values of the filtered skeleton points are utilized. This leads to the question which is the better approach. To answer this question the reconstruction errors of the explicit and implicit kernel methods are compared side-by-side in table 6.1. The kernels which are tested are $\hat{D}_E(\mathbf{x})$, $\text{minf}(\mathbf{x})$, $\hat{D}_{\text{sup}}\hat{D}_{\text{inf}}(\mathbf{x})$ and $\hat{D}_{\text{med}}(\mathbf{x})$. These are respectively the mean filter, the minification filter, the constrained opening, and the median filter. In the results we see that almost all kernel-based methods have lower reconstruction errors in their explicit mode than in their implicit mode. This is expected, as we only allow the core skeleton points to contribute to the EDT filtering, as they should be less affected by the noise compared to ligature points. The exception is the min filter, where the implicit version performs significantly better. This is because since more neighbors are evaluated, it is much more likely to find a better global minimum. Therefore the minification filter does not get any advantage of only using core skeleton points.. This is different in the constrained opening, as the supremum step is more sensitive to the ligature points, and therefore has an advantage in the explicit version. In the final comparison the explicit version is used for all the kernels, with the exception of the minification filter.

6.3.2 Comparison

To judge the effectiveness of the methods a comparison is made of all the relevant methods. The methods are compared quantitatively by comparing the reconstruction errors, and visually by comparing the protuberances of the surfaces of the reconstructed models. The best versions of the kernel-based methods (in terms of reconstruction error) are used from section 6.3.1, as well as the least-square projection from section 6.2.2, and the flat-projection method from section 6.2.3.

A table of the reconstruction errors of different models is shown in table 6.2. In terms of reconstruction error the $\hat{D}_{\text{sup}}\hat{D}_{\text{inf}}(\mathbf{x})$ method has the best performance for

4 out 8 models. The minification method performs better on the chair and cube model, while the mean kernel $\hat{D}_E(\mathbf{x})$ has a better performance on the bear and the cat model. Note that the bear and cat models are the smoothest two models, while the cube and chair have the strongest edges and are relatively flat. This is important as the ripple noise is much more visible on flat surfaces. The median-based kernel $\hat{D}_{\text{med}}(\mathbf{x})$ has very similar error values to the mean kernel, but performs slightly worse on all the tested models. The least-squares projection and the flat projection method have significantly worse errors than the kernel-based methods.

The visual comparisons of the reconstructed models with different filters are shown in figures 6.4 to 6.7. For each model, the noisy model, and the reconstructed models with and without EDT post-processing are shown. Here we can see that the minification gives the best visual result in terms of flatness in figures 6.4, 6.5 and 6.7. Especially in the chair, where the end-result is flat and has no visible perturbations on the surface. The fandisk result seems also to be noticeably cleaner, especially near the edges and at the bottom; the hammer result seems to be smoother as well. This is surprising as the error of the $\hat{D}_{\text{sup}}\hat{D}_{\text{inf}}(\mathbf{x})$ kernel is much better for this model, but its end-result is noticeably less clean. In terms of flatness the least-squares also gives reasonable results on the chair and the fandisk, but is does not seem to remove the noise near the edges.

For the pot model shown in figure 6.6, it is not immediately clear which has the best visual result. However, the $\hat{D}_{\text{sup}}\hat{D}_{\text{inf}}(\mathbf{x})$ kernel seems to remove the most noise under the top handle of the lid of the pot, while still maintaining the edges of the lid, while the minification filter reduced the edges of the lid in this case. The flat projection method seems to reduce most of the noise on the pot model and the fandisk model, but also shrinks the model, as well as destroying important features.

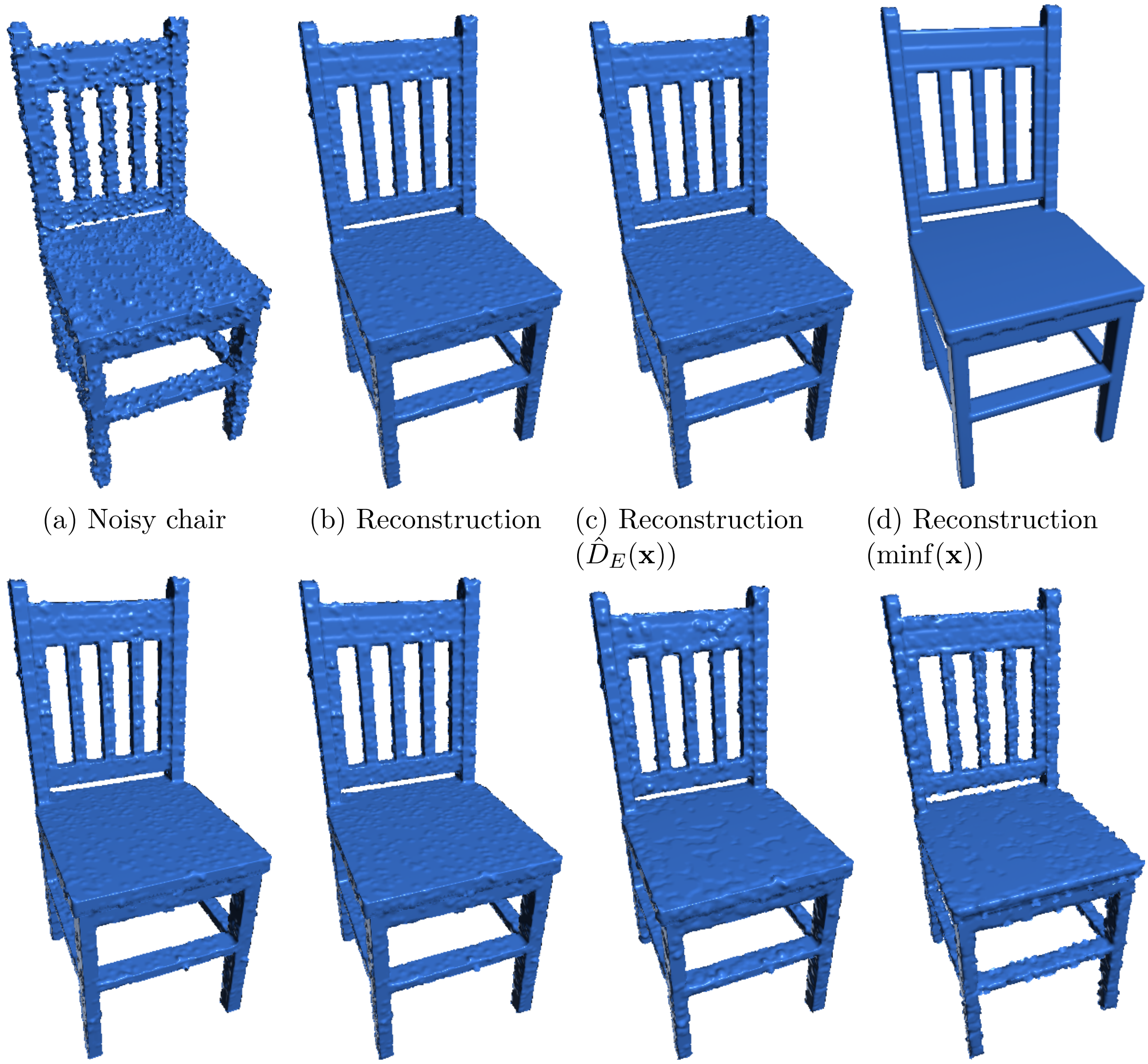
Comparing the visual results and the reconstruction errors the minification filter seems to give the best overall results. The error is the smallest on flat surfaces, which is where the 'EDT ripples' are the most visible. Additionally, the visual results are better than the $\hat{D}_{\text{sup}}\hat{D}_{\text{inf}}(\mathbf{x})$ kernel, despite having better reconstruction errors on four models.

6.3.3 Conclusion

After removing all ligature sheets, small noise structures on the surface still persist due to perturbations on the core skeleton and the EDT. An experiment on the fandisk has shown that that regularizing the EDT before reconstruction is the best option to remove the perturbations.

Several filters were proposed to remove the EDT ripples. The kernel-based filters: the mean, median, minification and the constrained opening as well as least-squares projection and flat projection. The kernel-based filter can be used in a implicit mode, where only the distance transform is used as input, or in explicit mode which only uses the distance values of the regularized skeleton. In an experiment where the reconstruction errors are compared of the implicit and explicit mode for each respective filter, it has been shown that the explicit version have lower errors, with the exception of the minification filter which proofs to be better in its implicit version.

From all the tested methods the EDT minification method gave the best visual results. Especially on models with flat surfaces, where in the reconstruction the perturbations no longer visible. Although the minification removes noise, it also has the side-effect of reducing sharp corners. Creating a filter that preserves these corners is a question of future work.



(a) Noisy chair

(b) Reconstruction

(c) Reconstruction ($\hat{D}_E(\mathbf{x})$)

(d) Reconstruction ($\mathrm{minf}(\mathbf{x})$)

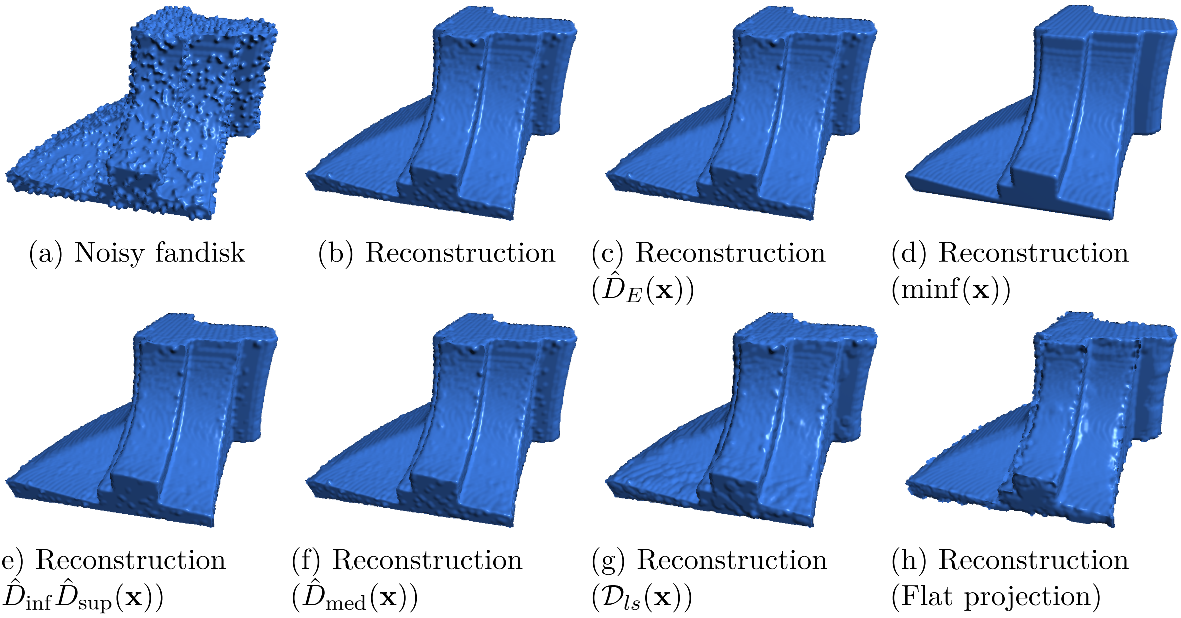
(e) Reconstruction ($\hat{D}_{\mathrm{inf}}\hat{D}_{\mathrm{sup}}(\mathbf{x})$)

(f) Reconstruction ($\hat{D}_{\mathrm{med}}(\mathbf{x})$)

(g) Reconstruction ($\mathcal{D}_{ls}(\mathbf{x})$)

(h) Reconstruction (Flat projection)

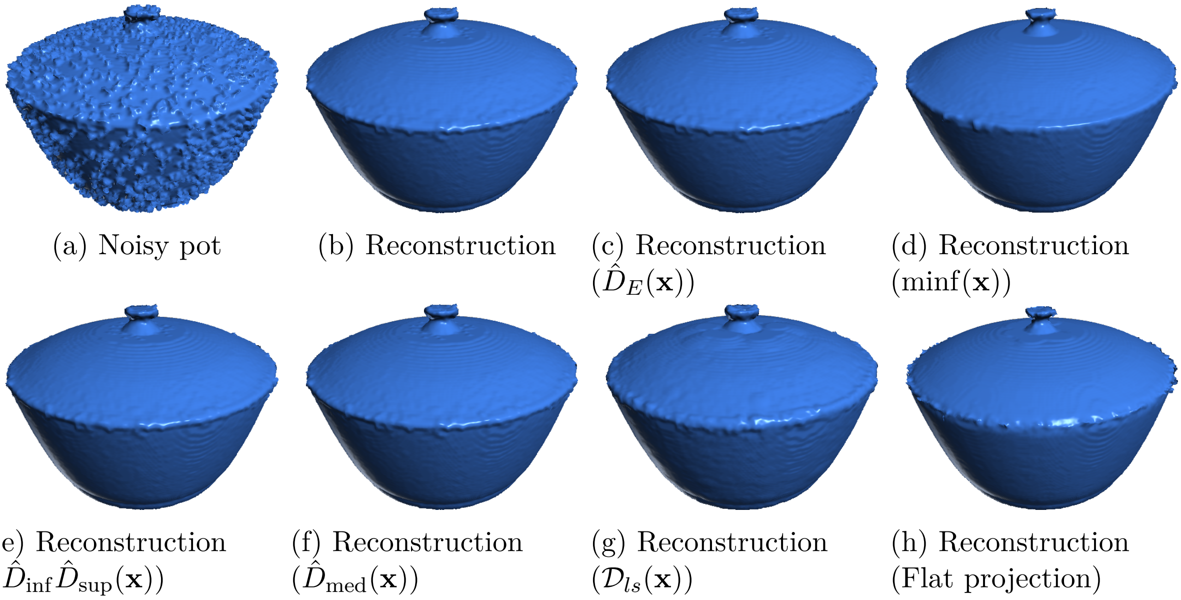
Figure 6.4: Comparison of reconstruction results of different EDT regularization methods on the chair model.



(a) Noisy fandisk (b) Reconstruction (c) Reconstruction ($\hat{D}_E(\mathbf{x})$) (d) Reconstruction ($\mathrm{minf}(\mathbf{x})$)

(e) Reconstruction ($\hat{D}_{\mathrm{inf}}\hat{D}_{\mathrm{sup}}(\mathbf{x})$) (f) Reconstruction ($\hat{D}_{\mathrm{med}}(\mathbf{x})$) (g) Reconstruction ($\mathcal{D}_{ls}(\mathbf{x})$) (h) Reconstruction (Flat projection)

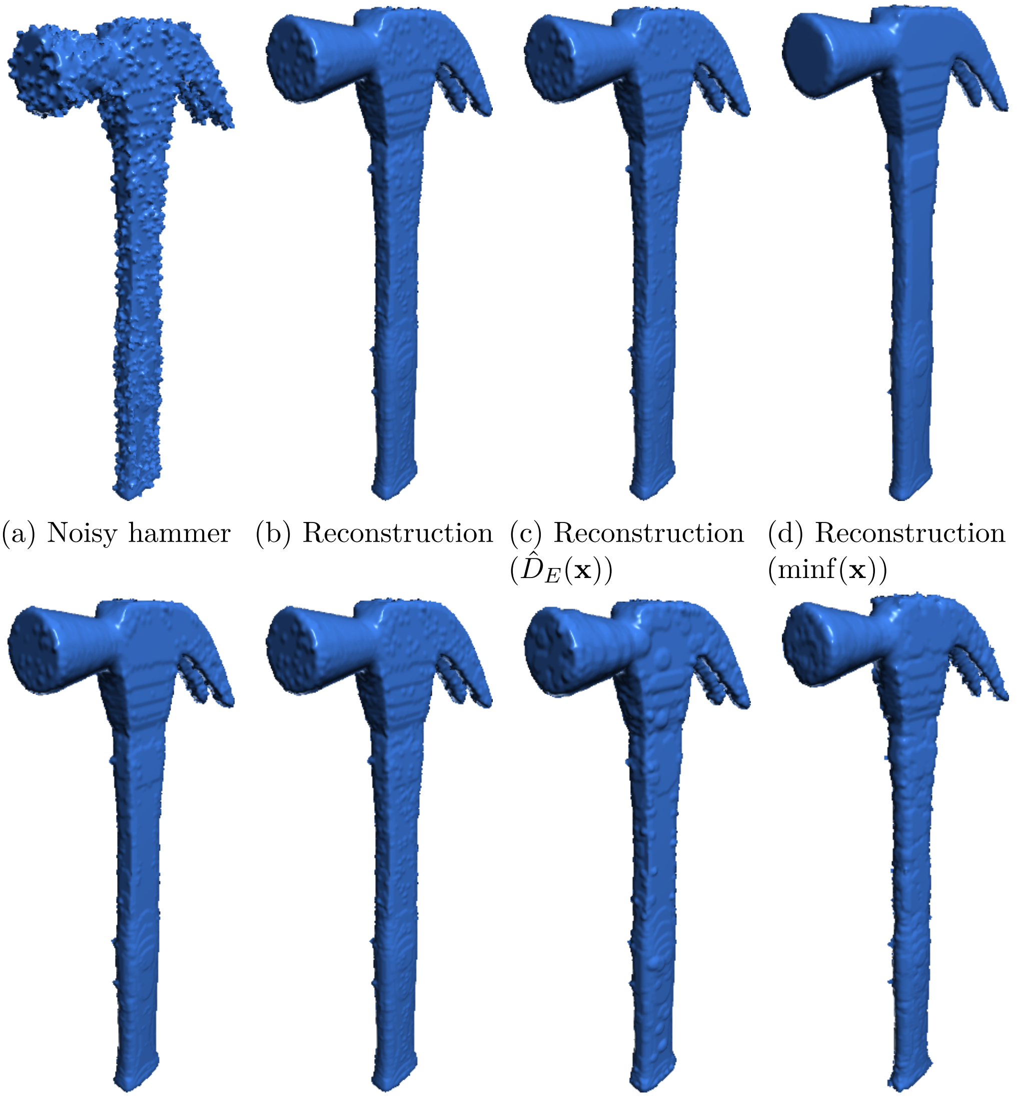
Figure 6.5: Comparison of reconstruction results of different EDT regularization methods on the fandisk model.



(a) Noisy pot (b) Reconstruction (c) Reconstruction ($\hat{D}_E(\mathbf{x})$) (d) Reconstruction ($\mathrm{minf}(\mathbf{x})$)

(e) Reconstruction ($\hat{D}_{\mathrm{inf}}\hat{D}_{\mathrm{sup}}(\mathbf{x})$) (f) Reconstruction ($\hat{D}_{\mathrm{med}}(\mathbf{x})$) (g) Reconstruction ($\mathcal{D}_{ls}(\mathbf{x})$) (h) Reconstruction (Flat projection)

Figure 6.6: Comparison of reconstruction results of different EDT regularization methods on the hammer model.



(a) Noisy hammer

(b) Reconstruction

(c) Reconstruction ($\hat{D}_E(\mathbf{x})$)

(d) Reconstruction ($\mathrm{minf}(\mathbf{x})$)

(e) Reconstruction ($\hat{D}_{\mathrm{inf}}\hat{D}_{\mathrm{sup}}(\mathbf{x})$)

(f) Reconstruction ($\hat{D}_{\mathrm{med}}(\mathbf{x})$)

(g) Reconstruction ($\mathcal{D}_{ls}(\mathbf{x})$)

(h) Reconstruction (Flat projection)

Figure 6.7: Comparison of reconstruction results of different EDT regularization methods on the hammer model.

7 Implementation

The theoretical framework presented here can be implemented in several ways. In this chapter the important implementation details of the pipeline are explained, and a justification is given of the several implementation choices. The skeleton extraction method has a large influence on the overall competency of the pipeline, and is therefore explained in section 7.1. The noise model which is used to generate the noisy shapes is given in section 7.2. Graph geodesic distances are used as the importance measure of the salience measure, and an estimation algorithm that quickly computes this is given in section 7.3. In the final step of the pipeline the reconstruction of the regularized skeleton is created. The approach which was used in this thesis is shown in section 7.4.

7.1 Skeleton extraction methods

Two skeleton extraction methods were implemented and integrated in the pipeline. This includes the IMA-based extraction method [27] and the mass-transport method from [29]. These methods are respectively introduced in section 7.1.2 and section 7.1.1. A short evaluation of the methods, and the justification of the choice of the final method is given in section 7.1.3.

7.1.1 IMA skeleton

The Integer Medial Axis (IMA) skeleton method is a general algorithm that computes 2D and 3D euclidean skeletons in linear time. The algorithm for computing the IMA skeleton is based on the feature transform, using a modification of a linear time algorithm for Euclidean distance transforms.

Recall that the feature transform is defined as $\mathcal{F}(\mathbf{x}, B) = \{\mathbf{y} \in B \mid ||\mathbf{x} - \mathbf{y}|| = D(\mathbf{x}, B)\}$. Thus it is the set-valued function that assigns $\mathbf{x} \in \mathbb{R}^d$ to its closest boundary points $\mathbf{y} \in B$. It is possible to compute $\mathcal{F}$ in linear time [27], but it is computationally cheaper and sufficient to compute, for every point $\mathbf{x}$, just a single feature transform point $f(\mathbf{x})$. So, the function $f(\mathbf{x})$ is incompletely specified by $f(\mathbf{x}) \in \mathcal{F}(\mathbf{x})$. In the implementation by Hesselink and Roerdink [27] a single feature point is selected based on lexical ordering. Using this incompletely specified feature transform the definition of the IMA skeleton is given as follows:

Definition 15. *Let E be the set of unit vectors so that $E = \{\mathbf{e} \in \mathbb{Z}^d \mid ||\mathbf{e}|| = 1\}$. Let X be the foreground image, and B the background. Then the IMA skeleton is given by the points $\mathbf{x} \in X$ where for some $\mathbf{e} \in E$ it holds that:*

$$||f(\mathbf{x}+\mathbf{e}) - f(\mathbf{x})|| \geq 1 \tag{7.1}$$

$$||\mathbf{m} - f(\mathbf{x}+\mathbf{e})|| \leq ||\mathbf{m} - f(\mathbf{x})|| \quad \textit{where } \mathbf{m} = \mathbf{x} + 1/2\mathbf{e} \tag{7.2}$$

The first condition makes sure that the maximum Euclidean distance of the neighboring feature points is larger than one. This can be seen as a small pruning threshold on the skeleton based on the euclidean distance metric. The second condition is introduced to get one point, rather than two, and specifically the point that is closest to the perpendicular bisector of the line segment from $f(\mathbf{x})$ and $f(\mathbf{x}+\mathbf{e})$. If $||\mathbf{m} - f(\mathbf{x}+\mathbf{e})|| = ||\mathbf{m} - f(\mathbf{x})||$, then both points are included.

The method also includes some simple pruning metrics, which are based on local attributes of a skeleton point and its feature points such as the bisector angle, the euclidean distance and the distance transform. These metrics are not discussed further, as they suffer from shortening branches of coarse features. In our pipeline the IMA method is only used in the skeleton extraction process, where the skeleton regularization is done by the adapted saliency pipeline.

Implicit Euclidean

Instead of extracting local features and computing the Euclidean distances of these feature points, the distances themselves can also be estimated numerically. This can be done using the intrinsic relationship of the feature points and the distance transform, which is shown in section 4.2.1. Recall that on a local coordinate structure, where the skeleton surface lies on the xy plane at the origin, the feature points of a skeleton point $\mathbf{x} \in \mathcal{S}$ are given by:

$$\mathbf{a}^{\pm} = \left(-rr_x, -r_y r, \pm r\sqrt{1 - r_x^2 - r_y^2}\right)^{\top} \tag{7.3}$$

where r is the radius so that $r = D(\mathbf{x})$. The distance between the feature points is equal to

$$||\mathbf{a}^{+} - \mathbf{a}^{-}|| = r\sqrt{1 - r_x^2 - r_y^2} \tag{7.4}$$

The general solution is given by the rotation of the local coordinate system of the Morge patch, but the Euclidean distance is invariant under rotation. If $\mathbf{x}$ does not lie on the xy plane we have that:

$$\begin{aligned} ||\mathbf{a}^{+} - \mathbf{a}^{-}|| &= r\sqrt{1 - r_x^2 - r_y^2 - r_z^2} && (7.5) \\ &\leq D(\mathbf{x})\sqrt{1 - ||\nabla D(\mathbf{x})||^2} && (7.6) \end{aligned}$$

Define this upper bound as:

$$\mathcal{E}(\mathbf{x}) = D(\mathbf{x})\sqrt{1 - ||\nabla D(\mathbf{x})||^2} \tag{7.7}$$

In the discrete case, i.e., $X \subseteq \mathbb{Z}^d$, the gradient $\nabla D(\mathbf{x})$ can be estimated using the central differences. The condition $\mathcal{E}(\mathbf{x}) \geq 1$ implies that $||\mathbf{a}^{+} - \mathbf{a}^{-}|| \geq 1$ (but not the other way around). Thus by applying the threshold rule $\mathcal{E}(\mathbf{x}) \geq 1$, we can estimate the IMA skeleton without feature extraction. The advantage is that we no longer need the second condition of equation (7.2), as as we do not use the neighboring points to estimate the feature points, preventing the duplicate pairs to appear. Additionally, if we already have computed the distance transform, this adaption is easier to implement and faster to compute. This method is used in the final results in chapter 8.

Comparison

In figure 7.1 a non-exhaustive comparison of IMA skeletons and the Implicit Euclidean (IE) skeletons is shown. In each row the methods are compared on different models with different simplification thresholds. The regularized skeletons are created by directly thresholding the geodesic measure for both methods. Note that these skeletons are extremely similar. The justification of using IE over the IMA method is the easier implementation.

7.1.2 Unified Skeleton Method

Jalba et al. [29] introduced an unified multiscale framework for planar, surface, and curve skeletonization. In this thesis we refer this work as the Unified Skeleton Model (USM). In this chapter USM method is briefly introduced.

The USM method is a framework that detects all types of skeletons using a single model, and also produces a multiscale representation which allows to progressively simplify, or regularize, all skeleton types. The backbone of the method is a contraction process which is described by a system of three PDE's:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{7.8}$$

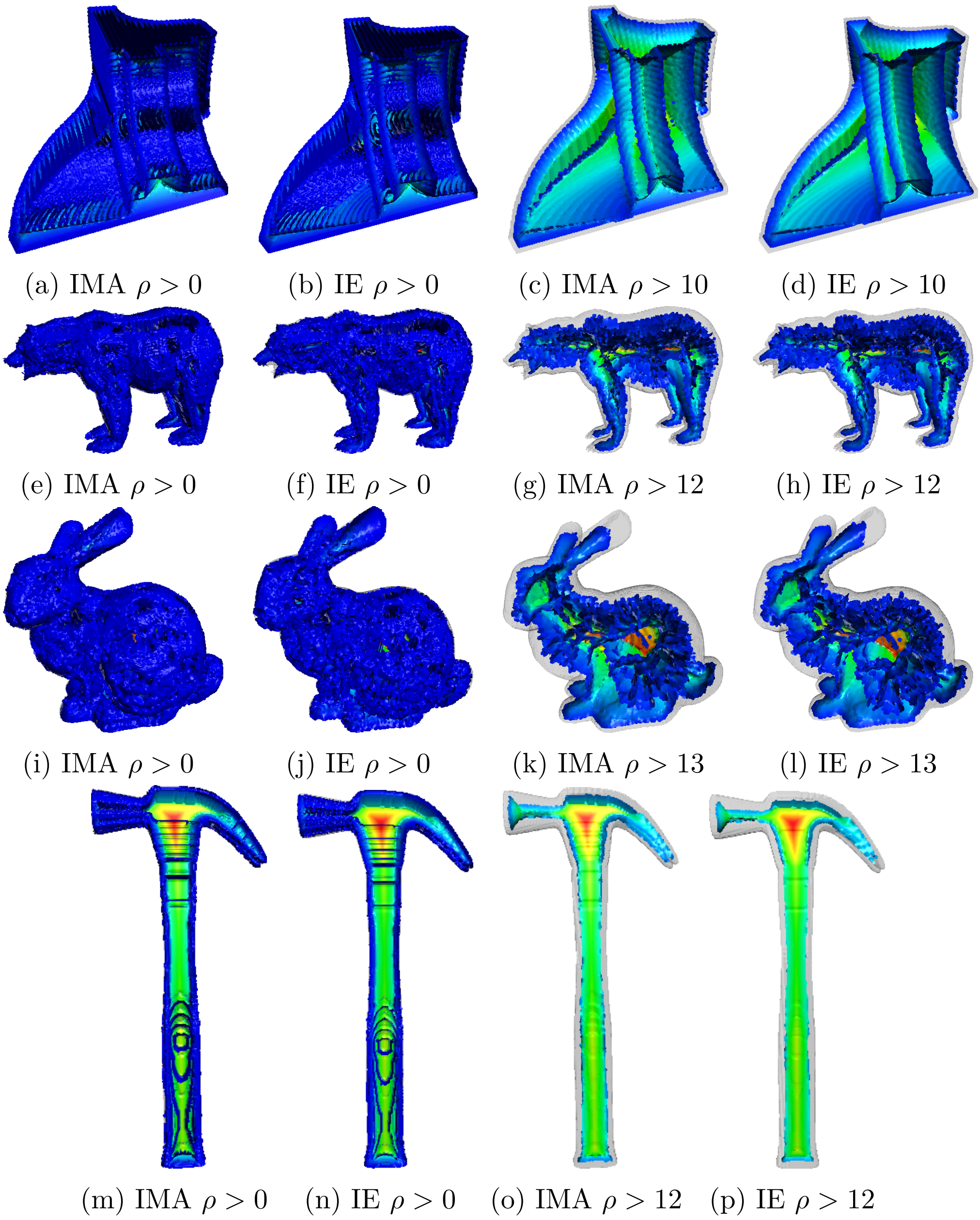
$$\frac{\partial \phi}{\partial t} + \nabla \cdot \phi = 0 \tag{7.9}$$

$$\mathbf{u} = \frac{\nabla \phi}{|\nabla \phi|} \tag{7.10}$$

Here t is the time parameter, ρ (only in this context) is the density as a function of t, and $\phi \in [-1, 1]$ is a field which is 1 inside the contracted shape, and -1 outside, and the field $\mathbf{u}$ gives the contraction direction. The boundary of the contracting shape is implicitly given by $Y_t = \{\mathbf{x} \in \mathbf{X} \mid \phi(\mathbf{x}, t) = 0\}$. The importance measure of the method is evaluated by:

$$\lambda(\mathbf{x}) = \max_{t>0} \rho(\mathbf{x}, t) \tag{7.11}$$

Thus the skeleton importance $\lambda(\mathbf{x})$ is given as the maximum density that has reached a certain location. Intuitively, the model describes a conservative advection process where mass, uniformly spread on X, flows on shortest paths from X to its surface skeleton $\mathcal{S}$; then, along $\mathcal{S}$ on shortest paths to the curve skeleton; and finally along the curve skeleton on shortest paths to the global root $\mathbf{q}$.



(a) IMA $\rho > 0$ (b) IE $\rho > 0$ (c) IMA $\rho > 10$ (d) IE $\rho > 10$

(e) IMA $\rho > 0$ (f) IE $\rho > 0$ (g) IMA $\rho > 12$ (h) IE $\rho > 12$

(i) IMA $\rho > 0$ (j) IE $\rho > 0$ (k) IMA $\rho > 13$ (l) IE $\rho > 13$

(m) IMA $\rho > 0$ (n) IE $\rho > 0$ (o) IMA $\rho > 12$ (p) IE $\rho > 12$

Figure 7.1: Comparison of IMA- and IE- skeletons, where the geodesic estimate is used for both methods.

7.1.3 Evaluation of skeleton methods

Here, the IMA method (IE) is compared to USM, and explained why the IMA method is used in our pipeline.

Noise problems of USM

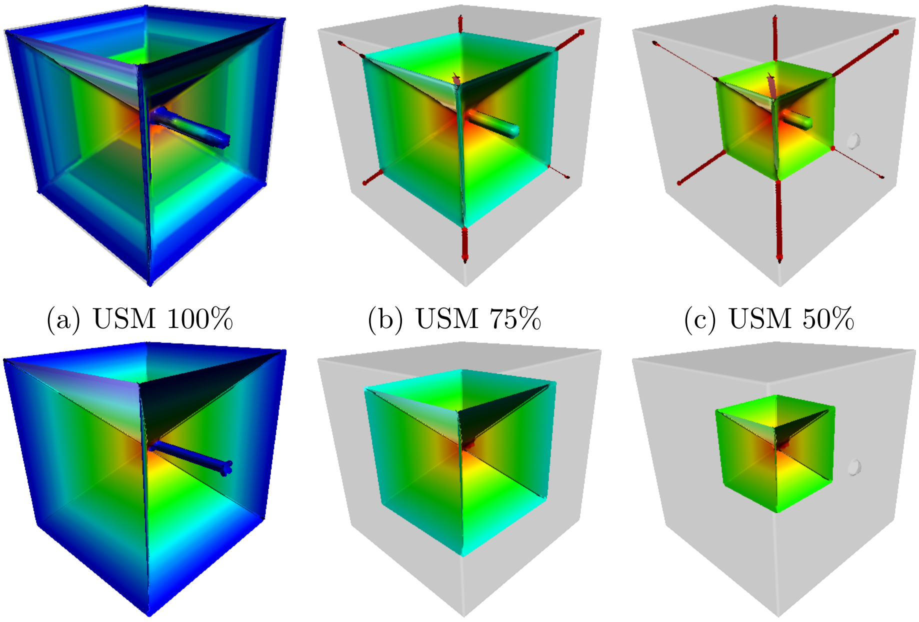
Since USM is a field method it requires smoothing prior to the skeleton computation to achieve good results. This is done by smoothing $D(\mathbf{x})$ using a linear smoothing kernel, and then thresholding the result to create a smooth input volume. Unfortunately, this causes feature blurring of the edges we want to preserve in our denoising process. When no smoothing is used, or when the noise is larger than the smoothing kernel size, the method has problems with separating ligature branches from the core skeleton. This is largely visible in figure 7.2, where increasing the importance threshold on $\lambda(\mathbf{x})$ is unable to remove the ligature sheet of the singular bump from the skeleton. However, using the IMA method (IE) with the geodesic importance $\rho(\mathbf{x})$ we are able to remove the bump after regularizing the skeleton. Note that the saliency pipeline is not used for the simplifications, and the comparison is shown by directly thresholding the respective importance measures.

USM as an extraction method

Instead of using the importance measure of the USM, one could use USM for only extracting the skeleton. Meaning, we extract the skeleton using the threshold rule $\lambda(\mathbf{x}) > 0$, and then use the geodesic importance $\rho(\mathbf{x})$ for further processing in the pipeline. Although this works for some models, the problem is that the USM method is not guaranteed to extract centered skeletons, which makes finding corresponding features more difficult. An example is shown in figure 7.3. In this example, the true graph geodesic $\rho(\mathbf{x})$ is used. At the noncentered points we are unable to extract two salient feature points, and therefore the geodesic distances are shorter than expected. Note that this can be mitigated by improving the extended feature transform [49], but this may introduce accuracy problems, which in turn makes calculating reliable derivative estimates harder, which is why it is not considered.

Conclusion

For the use-cases of this thesis, IMA (IE) is more practical, as the skeleton points are centered, which allows an easier implementation of the geodesic distances. Additionally, the importance measure $\lambda(\mathbf{x})$, when applied to a non-smoothed model, has trouble differentiating ligature sheets from core skeleton parts. Rather than adapting the USM method in various ways to accommodate these problems, it was chosen to use the IMA-based method combination with the geodesic method.



(a) USM 100%

(b) USM 75%

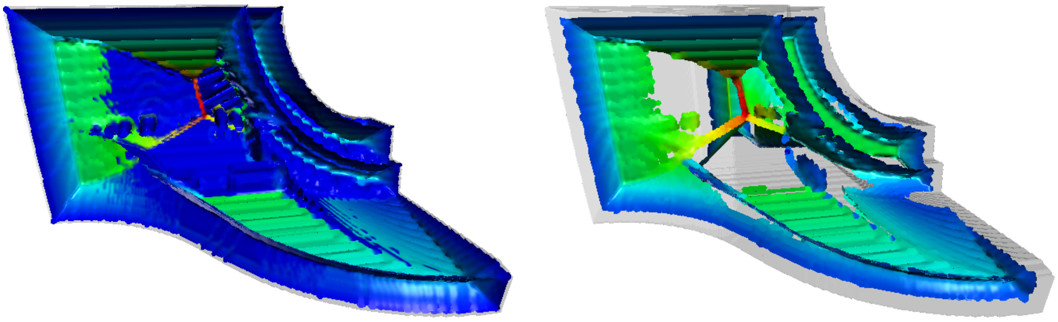
(c) USM 50%

(d) IMA 100%

(e) IMA 75%

(f) IMA 50%

Figure 7.2: Top: simplifications of the skeleton using USM with the maximum density $\lambda(\mathbf{x})$. Bottom: simplifications of the skeleton using the adapted IMA method (IE) with the geodesic importance $\rho(\mathbf{x})$



(a) $\lambda(\mathbf{x}) > 0$, $\rho(\mathbf{x}) > 0$

(b) $\lambda(\mathbf{x}) > 0$, $\rho(\mathbf{x}) > 20$

Figure 7.3: Skeleton extraction using USM with $\lambda(\mathbf{x}) > 0$ on the fandisk model. The skeleton importance which is visualized is the geodesic metric $\rho(\mathbf{x})$. This shows that pruning the skeleton using ρ will cause holes to appear when using the USM skeleton. The cause is non-centered skeleton points.

7.2 Noise model

In this section it is explained how the synthetic noise is created, and an explanation of the parameters is given. The noise is generated directly on the voxel-based model X. This noise model is specifically chosen so that noise protuberances can be interactively added to the model at specific locations, which allows us to directly confirm, replicate, and compare different (experimental) methods. A distinction is made between 0D noise, where the noise protuberances are added independently on the surface, and 1D noise, where the protuberances are added as groups which are connected by a path.

The noise is generated by random protrusions on the surface of the model. The surface M of a binary model is extracted using mathematical morphology:

$$M = (X \oplus b) \backslash X \tag{7.12}$$

Here, b is the 26-connected structuring element. A protrusion is created on a point $\mathbf{x} \in M$ using a noise kernel defined as the mapping $\mathcal{N} : M \mapsto \mathcal{P}(\mathbb{E})$. The kernel can be a sphere with a radius $r \in \mathbb{R}$, which is given as

$$\mathcal{N}_s(\mathbf{x}) = \{\mathbf{y} \mid \mathbf{y} \in \mathbb{E} \wedge |\mathbf{y} - \mathbf{x}| < r\} \tag{7.13}$$

The protrusion also has a constant height parameter $h \in \mathbb{Z}$, where the noise kernel is added in the direction of the normal. Formally it is given by the following operation:

$$\mathcal{H}(\mathbf{x}) = \bigcup_{\mathfrak{h} \in [0,h]} \mathcal{N}_s(\mathbf{x} + \mathfrak{h}\mathbf{n}) \tag{7.14}$$

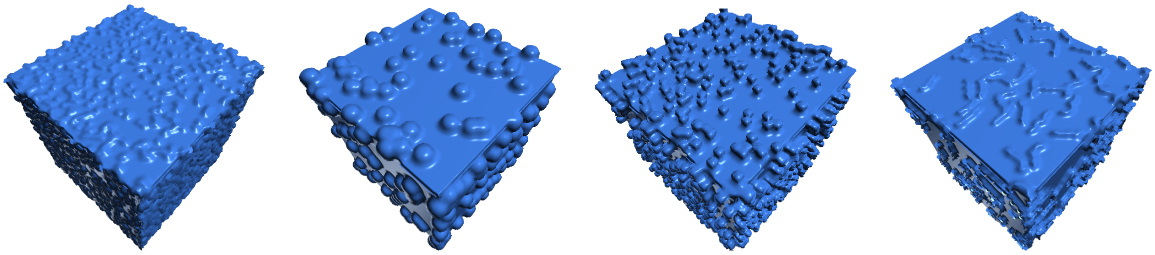
where $\mathbf{n}$ is the normal of $\mathbf{x}$, and is estimated by using least-squares on a local neighborhood of $x \in \mathbf{M}$.

The protrusion is duplicated on the surface in a curvilinear fashion, which consists of a succession of random steps which deviates slightly (by angle) from a single direction. The number of steps in voxels is given by the parameter l. The random direction $\mathbf{d}$ is evaluated by a uniformly distributed unit-length vector field which is tangent to the surface (and perpendicular to the normal $\mathbf{n}$). Denote this random path as $A \in \mathbb{E}^l$. Recall that the l-tuple $A = (\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_l)$ is called a path of length l if $\{\mathbf{a}_k, \mathbf{a}_{k+1}\} \subseteq \mathcal{C}$, for all $k \in [1, l-1]$.

Definition 16. *Let $A \in \mathbb{E}^l$ a random path of length l. Let the neighbors of a point $\mathbf{x} \in M$ be given as as the set-valued function $N(\mathbf{x}) = \{\mathbf{y} \mid \mathbf{y} \in M \wedge \{\mathbf{x}, \mathbf{y}\} \in \mathcal{C}\}$. The path A can be decomposed into a series of successive steps, denoted as $\mathbf{s}_k = \mathbf{a}_{k+1} - \mathbf{a}_k$, where $k \in [1, l-2]$. Given the direction $\mathbf{d}$, the probability of a step is given by:*

$$P(\mathbf{s}_k = \mathbf{e}) = \begin{cases} 1/|N(\mathbf{x})| & \textit{if } \mathbf{e} \in N(\mathbf{x}) \textit{ and} \\ 0 & \textit{otherwise} \end{cases} \tag{7.15}$$

The path A is thus evaluated by a random walk with a constrained direction $\mathbf{d}$ on the discrete surface M. At every site the next step is randomly evaluated using



(a) $p = 0.03$, $r = 2$, $h = 2$, $l = 1$

(b) $p = 0.003$, $r = 7$, $h = 1$, $l = 1$

(c) $p = 0.01$, $r = 3$, $h = 6$, $l = 1$

(d) $p = 0.003$, $r = 2$, $h = 3$, $l = 20$

Figure 7.4: Showcase of different noise parameters on a cube.

the probability distribution given by equation (7.15). Given the path A and that $\mathbf{a}_1 = \mathbf{x}$, the complete curvilinear noise protrusion is given by:

$$\mathcal{L}(\mathbf{x}) = \bigcup_{\mathbf{a}_i \in A} \mathcal{H}(\mathbf{a}_i) \tag{7.16}$$

Finally, the noise streams are added with a certain probability $P(\mathbf{x}) = p$, where $p \in [0, 1]$ is an incidence parameter. Let $T \in \mathcal{P}(M)$ a trial drawn from the sample space M with the probability $P(\mathbf{x}) = p$. Then the noisy model $\hat{X}$ is given by the following equation

$$\hat{X} = \bigcup_{\mathbf{y} \in T} X \cup \mathcal{L}(\mathbf{y}) \tag{7.17}$$

Which is a simple union of the protuberances generated at incident sites $\mathbf{y} \in M$.

The synthetic noise has thus the following parameters:

- p : The incidence value, the probability that a on a point $\mathbf{x} \in M$ a protrusion is generated.
- r : The radius of the kernel of the protrusion. In all cases in this thesis it is the radius of a sphere.
- h : The height of the protrusion in the direction of the normal of $\mathbf{x} \in M$.
- l : The 1D noise length of the protrusion in voxels. Setting the parameter $l > 1$ will cause curvilinear elements in random directions. If set to $l = 1$ the synthetic noise will be 0D.

In figure 7.4 a few examples are shown of the effect of different noise parameters on a cube.

7.3 Estimating geodesic distances

Graph geodesic distances are used as the importance measure of the salience measure. The original graph geodesic is calculated by evaluating the shortest path between the feature points of the discrete surface M. Although this importance measure performs better than simple direct measures such as the Euclidean distance [50], it is prohibitively slow to compute. This would prevent us to use the saliency measure on models with a quality sample size. To allow higher quality samplings, an estimation technique is developed to approximate the geodesic length between two feature points. First, the estimation approach is given in section 7.3.1. Second, the estimation method is evaluated in section 7.3.2, where also a conclusion is given.

7.3.1 Estimation

Let $\mathbf{a} \in M$, and $\mathbf{b} \in M$ two arbitrary points on the surface M. Denote the shortest Euclidean path by $\gamma(\mathbf{a}, \mathbf{b})$ which is defined as:

$$\gamma_{\mathbf{a},\mathbf{b}}(\lambda) = \mathbf{a} + \lambda(\mathbf{b} - \mathbf{a}) \tag{7.18}$$

where $\lambda \in [0, 1]$.

Let $\phi_{\mathbf{a},\mathbf{b}} : \lambda \mapsto M$ be the continuously differentiable curve which minimizes the distance on the surface between $\mathbf{a}$ and $\mathbf{b}$ where $\phi(0) = \mathbf{a}$ and $\phi(1) = \mathbf{b}$. Assume that ϕ is constrained in a plane, so that there exists a vector $\mathbf{d}$ and a constant D where $\mathbf{d} \cdot \phi(\lambda) + D = 0$ for all $\lambda \in [0, 1]$. Under this assumption we can express $\phi_{\mathbf{a},\mathbf{b}}$ with two coordinates in a local basis $(\gamma_{\mathbf{a},\mathbf{b}} - D, \gamma_{\mathbf{a},\mathbf{b}} \times \mathbf{d} - D)$. Note that $\gamma_{\mathbf{a},\mathbf{b}}$ must be contained in this plane, as $\phi_{\mathbf{a},\mathbf{b}}$ contains both $\mathbf{a}$ and $\mathbf{b}$.

It follows that there exists a local parametrization $r(\lambda) = |\gamma(\lambda) - \phi(\lambda)|$ which gives the local height of $\phi(\lambda)$ in respect to $\gamma(\lambda)$. Under these conditions the arc length (which is equivalent to the geodesic length) is given by:

$$\begin{aligned} \text{Arclength}(\phi) &= ||\mathbf{b} - \mathbf{a}|| \int_0^1 \left| \frac{\partial \phi(\lambda)}{\partial \lambda} \right| d\lambda & (7.19) \\ &= ||\mathbf{b} - \mathbf{a}|| \int_0^1 \sqrt{1 + \frac{\partial r(\lambda)^2}{\partial \lambda}} d\lambda & (7.20) \end{aligned}$$

An alternative way to see is that we are projecting γ onto M using the plane with the shortest geodesic distance. The function $r(\lambda)$ gives the distance between the projected point and the point of origin. Here the factor $||\mathbf{b} - \mathbf{a}||$ appears because λ is normalized so that $\lambda \in [0, 1]$.

Instead of computing the plane where ϕ is minimal, one could try to estimate $r(\lambda)$. A good candidate could be the Euclidean distance transform $D(\gamma(\lambda))$, as it gives the distance to the closest point to the surface. Here the assumption lies that many points are on the shortest geodesic path are also the closest to their Euclidean counterpart. Define this estimate as $\hat{r}(\lambda) = D(\gamma(\lambda))$, and define $\mathbf{v} = (\mathbf{b} - \mathbf{a})/||(\mathbf{b} - \mathbf{a})||$ then the derivative of $\hat{r}$ becomes:

$$\frac{\partial \hat{r}(\lambda)}{\partial \lambda} = \nabla D(\gamma(\lambda)) \cdot \mathbf{v} \tag{7.21}$$

	mean (ϵ)	maximum (m)
shark	0.0786	0.8414
bear	0.0636	0.5432
bunny	0.0781	0.8122
cat	0.0729	0.7314
chair	0.2471	0.7747
fandisk	0.1234	0.8109
hammer	0.1188	0.5818
cube	0.1169	0.2884
pot	0.0484	0.3830

Table 7.1: Error of geodesic estimate when using the graph geodesic as ground truth.

which is the directional derivative of the distance transform when tracing the Euclidean path between **a** and **b**. The final estimate is then given by substitution:

$$\phi(\lambda) \approx |\mathbf{b} - \mathbf{a}| \int_0^1 \sqrt{(\nabla D(\gamma(\lambda)) \cdot \mathbf{v})^2 + 1}\, d\lambda \tag{7.22}$$

Here the gradient is estimated using central differences, and the integral using a finite sum.

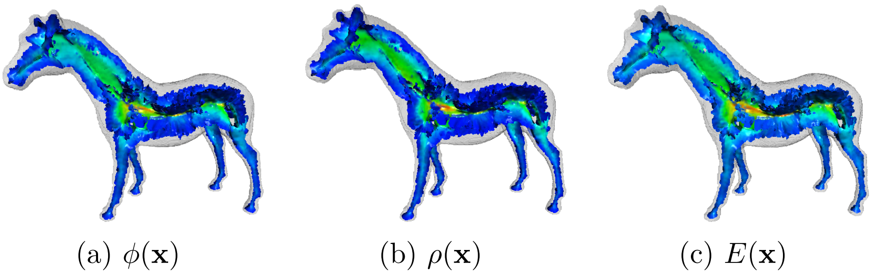
Evaluation

Here a qualitative and quantitative evaluation is given of the estimation approach compared to the graph geodesic metric (which is taken as ground-truth). For the evaluation of the geodesic distances the following error measures (from [14]) are used:

$$\epsilon = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \frac{|\rho(\mathbf{x}) - \rho_{est}(\mathbf{x})|}{\rho(\mathbf{x})} \tag{7.23}$$

$$m = \max_{\mathbf{x} \in \mathcal{S}} \frac{|\rho(\mathbf{x}) - \rho_{est}(\mathbf{x})|}{\rho(\mathbf{x})} \tag{7.24}$$

From a selection of models the skeletons were extracted using the IMA method (IE). Then the geodesic estimate is applied to the feature pairs of the skeleton points, and compared to the graph geodesic. The results are shown in table 7.1. We can see that at most models there the mean error (ϵ) is 10% on average, at the worst case around 24% (chair model), and at the best case around 5% (pot model). The maximum error ($m = 1$) is on average around 64%, but at the worst case around 84% (bunny model), and at the best case around 29% (cube model). The maximum errors are quite significant. The largest errors occur with the models with thin surfaces, such as the seat of the chair or the fins of the shark. This is expected, as the geodesic arcs $\gamma_{\mathbf{a},\mathbf{b}}(\lambda)$ are then very elongated, so that the ratio between the euclidean distance and the geodesic distance is very large. The estimation suffers in these cases, as our assumption, that many points on the shortest geodesic path are also the closest to their Euclidean counterpart, is then violated.



(a) $\phi(\mathbf{x})$ (b) $\rho(\mathbf{x})$ (c) $E(\mathbf{x})$

Figure 7.5: Visual comparison of the geodesic estimate $\phi(\mathbf{x})$, the graph geodesic $\rho(\mathbf{x})$ and the euclidean distance $E(\mathbf{x})$ on the horse model. The importance metrics are colored using the rainbow colormap, where the minimum and maximum are chosen independently for maximum contrast.

Although ϕ can be seen as a rough estimate, it is still more similar to the geodesic measure than the Euclidean distance. This can be seen in figure 7.5, where the estimate is almost identical in appearance to the geodesic metric, while the Euclidean distance has a more uniform distribution around the neck and the legs. Thresholding $E(\mathbf{x})$ would disconnect importance sheets from the legs and the neck at lower thresholds than it does for $\rho(\mathbf{x})$ or $\phi(\mathbf{x})$.

7.3.2 Conclusion

Although the geodesic estimate has its uses, it was found that it has problems with estimating thin parts of the models such as the seat of a chair, or the fins of a shark. For this reason the geodesic estimate is not used for the final results, but instead the graph geodesic measure is used. However, it has been a practical instrument for quickly testing new results.

7.4 Reconstruction

Recall that reconstruction is given by the union of all spheres centered on the points comprising the skeleton, each with a radius given by the distance transform $D(\mathbf{x}, B)$. Let $\mathcal{S}$ be the skeleton of X then we have that:

$$X = R(\mathcal{S}) = \bigcup_{\mathbf{x} \in \mathcal{S}} \{\mathbf{y} \mid y \in D(\mathbf{x}, B)\} \tag{7.25}$$

The direct evaluation of this function leads to a slow algorithm that is superlinearly dependent on the number of skeleton points. Here, a reconstruction algorithm is presented that is in $O(n)$, where $n = |\mathbb{E}|$ the number of voxels of the input volume. This is by recovering the EDT from skeleton points together with their sphere radii, and then thresholding the result.

7.4.1 Reverse Euclidean distance transform

In the same way that the Euclidean distance transform gives the distance to the background by using the original input shape, the Reverse Euclidean Distance Transform (REDT) can recover $D(\mathbf{x})$ by only using the Medial Axis Transform as input.

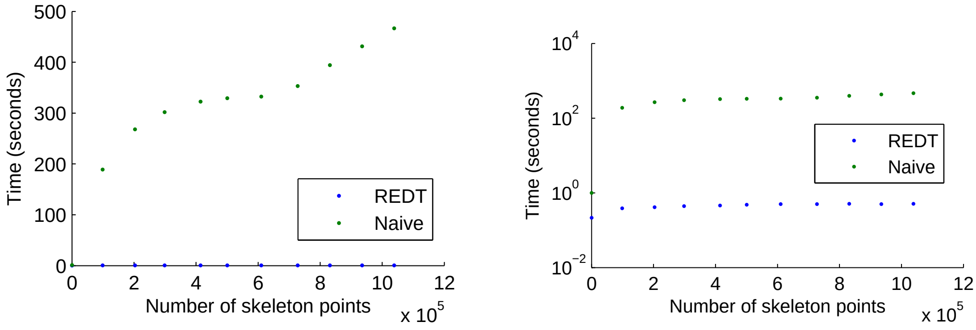
A point $\mathbf{x} \in \mathbb{E}$ belongs to $\mathcal{S}$ if it contained in at least one sphere whose center is a point $\mathbf{y}$ of $\mathcal{S}$, with radius $D(\mathbf{y})$. The distance can be recovered by taking the maximum distance to each point $\mathbf{y}$ subtracted by the radius. Hence, the REDT consists of the following equation

$$\bar{D}(\mathbf{x}, \mathcal{S}) = \max\{D(\mathbf{y}) - ||\mathbf{y} - \mathbf{x}|| \mid \mathbf{y} \in \mathcal{S}\} \tag{7.26}$$

Note that the following equality holds $\bar{D}(\mathbf{x}, \mathcal{S}) = D(\mathbf{x}, R(\mathcal{S}))$, as is proven in [13]. Therefore, the model X can be recovered by

$$X = \{\mathbf{x} \mid \mathbf{x} \in \mathbb{E} \wedge \bar{D}(\mathbf{x}, \mathcal{S}) > 0\} \tag{7.27}$$

Computing the reverse euclidean transform is done using a variation of the algorithm presented by Coeurjolly and Montanvert [13], which is in $O(n)$. Thus, by recovering



(a) Plot (b) Log-plot

Figure 7.6: Difference of reconstruction speed using the naive evaluation and the REDT approach.

the euclidean distance transform by using the skeleton points and their radii, we can efficiently reconstruct the model. The steps are not given in detail as it is not important for the outcome of the results; any algorithm that computes the reverse euclidean distance transform gives the same output.

7.4.2 Comparison

In figure 7.6 a speed-comparison is shown of the implemented algorithm and the naive method using equation (7.25). The cube model was used for this result; the plots were created by successively thresholding the importance measure $\rho(\mathbf{x})$ using equally-spaced thresholds, and then plotting the number of the skeleton points versus the time it takes to reconstruct the model. The time of the REDT algorithm takes around 0.5 seconds, while the naive approach takes from 1 second to 500 seconds depending on the number of skeleton points. It shows that the approach using the reverse euclidean distance transform is roughly 600 times faster on average (on this model), independent of the number of skeleton points.

7.5 Conclusion

Skeleton Extraction Two skeleton extraction methods were implemented and integrated in the pipeline: the IMA-based extraction method [27] and the mass-transport method (USM) from [29]. The IMA method is explained, and a quick and simple estimation method (IE) is given by implicitly thresholding the euclidean distance without extracting the feature points. The USM method is also introduced, and an evaluation is given, where it was compared to IMA (IE). It was concluded that for the use-cases of this thesis the use of IMA (IE) is more practical, as the skeleton points are centered, which allows an easier implementation of the geodesic distances. Additionally, the importance measure $\lambda(\mathbf{x})$, when applied to a non-smoothed model, has trouble differentiating ligature sheets from core skeleton parts.

Noise Model The noise model was introduced. It is generated by directly adding kernel-based protuberances on the voxel-based model, where the kernel is a sphere in all cases of this thesis. The noise-generation can be influenced with the following parameters: the incidence value (p), the radius of the kernel (r), the height of the kernel (h) and the length of the curvilinear paths of the protuberances. The effect of different parameters has been shown by generating noise on the cube model.

Geodesic Estimate To allow higher quality samplings, an estimation technique was developed to approximate the geodesic length between two feature points. The error of this geodesic estimation approach has been investigated. Although it works well on average, its use is limited because the error on models with thin parts is significantly higher. However, it has been a practical instrument for quickly testing new results.

Reconstruction In the final step of the pipeline the model is reconstructed. An approach based on the Reverse Euclidean Distance Transform (REDT) is presented that performs roughly 600 times faster than naively splatting spheres.

8 Results

In the previous chapters multiple adaptations and their implementations were shown and compared. Here we show the final results where the best performing adaptations are integrated into the pipeline. In particular, the derivative-based measure introduced in chapter 4, the global streamline method shown in chapter 5 and the minification filter given in chapter 6.

The adapted pipeline is tested on a selection of models with convex noise, where the intermediate steps are shown as well as the final reconstruction. These results are shown in section 8.1. The end-results are also compared to various other smoothing methods. To make the comparison, isosurfaces are extracted from the noisy models as well as from the final reconstructions. This comparison is given in section 8.2, and a conclusion of the final results can be found in section 8.3.

8.1 Complete pipeline

The saliency pipeline that was introduced in chapter 3. Throughout this thesis, several adaptations were proposed that improves the result for 3D models. An overview of this adapted pipeline is shown in section 8.1.1. The intermediate steps and the final results on a selection of noisy models are shown in section 8.1.2.

8.1.1 Adapted pipeline

Just as the original pipeline the adapted pipeline is divided into two stages, where in the first stage convex noise is removed using the foreground skeleton, and optionally, in the second stage concave noise is removed using the background skeleton. The first stage and second stage consists of the following steps:

1. Compute the foreground skeleton of the noisy binary image.
2. Compute the derivative measure $\overline{\sigma}(\mathbf{x})$, and create saliency measure ρ^+ using the global streamline method with the threshold k.
3. Prune skeleton by thresholding the saliency measure $\rho^+(\mathbf{x})$ using the threshold t.
4. Optionally, perform minification to remove EDT perturbations on models with flat surfaces.
5. Reconstruct the simplified skeleton and the (regularized) EDT; which will result in the smooth model where cusps under the threshold are removed.

There are three large differences with the original pipeline. First, instead of the saliency measure $\sigma(\mathbf{x})$, the derivative measure $\bar{\sigma}(\mathbf{x})$ is used. Second, the global streamline method is computed to create the saliency measure ρ^+, which can be used to regularize the skeleton by directly thresholding the measure. It is no longer needed to select the connected component which contains the root. Last, before the reconstruction phase, the EDT is optionally regularized using minification to remove the 'EDT ripples'.

8.1.2 Results

On a selection of models 1D noise is added using the parameters ($p = 0.02, r = 2.1, h = 3, l = 6$). The resolutions and the total processing time for each model is shown in table 8.1. The skeletons were extracted using the adapted IMA method (see section 7.1.1). For the importance measure ρ the (estimated) geodesic measure is used. For all the models the global streamline method was used with the threshold $k = 0.9$ for the derivative measure, and the threshold $t = 18$ is used for ρ^+ to regularize the skeleton. For the EDT minification the implicit mode is used with the radius parameter set to $r = 3$. The final reconstruction step is done using the vector-based method introduced in section 7.4. All the models and skeletons are visualized using point-based rendering with splat kernels.

The results of the pipeline on a selection of models are shown in figures 8.1 to 8.9. On the top row the skeletons in the various stages of the pipeline are visualized, where on the bottom row the original noisy model and its reconstructions are shown. Although the raw skeletons contain a large amount of ligature branches, we can see that using the ρ^+ metric the core skeleton can almost fully be extracted. The regularized skeletons are both visualized using the EDT and the ρ^+ metric using the rainbow color map. Just as in chapter 4, low importance areas are colored blue, while high-importance areas are colored red. Although the importance measure ρ^+ is not without flaws, as it assigns lower importance values in a few cases to important edges (as can be seen in the fandisk model), the assigned importance values of these edges are still considerably higher than the ligature sheets.

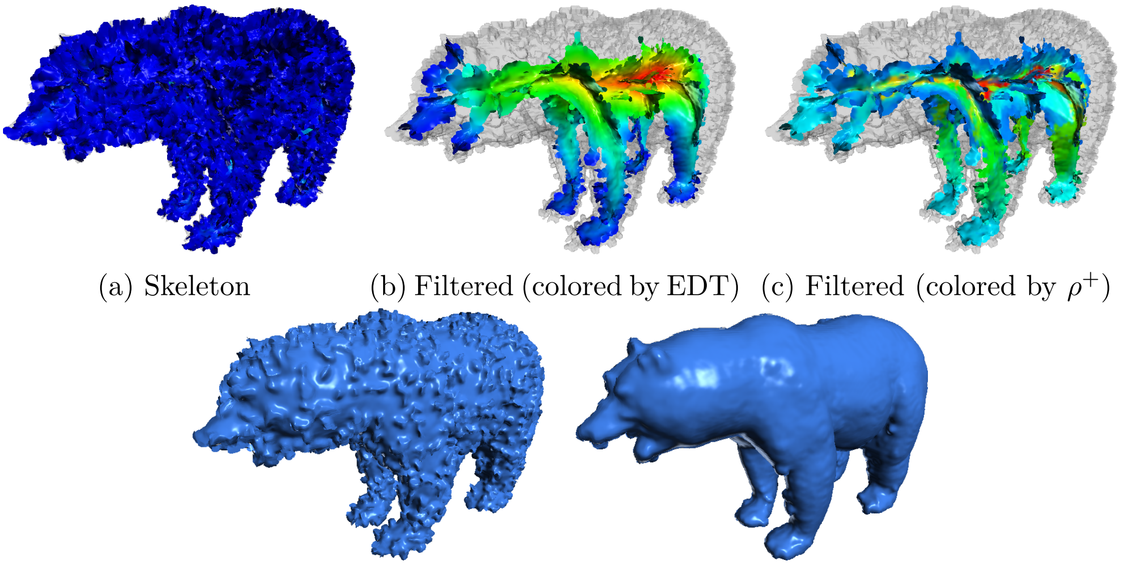
In the reconstructions of the models most of the important features are kept, with the exception of the shark model. Here, the threshold $t = 18$ does remove important core skeleton parts and as a result truncates the tips of the fins. This is because the fins are relatively flat and as such do not increase in importance along the skeleton axis, which in turn causes the derivative measure to go below the threshold k. Note that by choosing a lower threshold, i.e., $t = 10$ will keep the fins mostly in tact while still removing most of the noise. For the other models the threshold $t = 18$ is sufficient in removing almost all of the ligature branches without affecting the core skeleton. For example, the sides of the cubes, the ears of the bunny and the edges of the fandisk are all kept intact.

The models of the figures 8.2, 8.3, 8.5, 8.8 and 8.9 contain relatively flat surfaces where 'EDT ripples' are visible in the reconstruction. Therefore, the minification filter improves the final reconstructions for these models. For the pot model shown in figure 6.6 this is not the case, as was already discussed in chapter 6. The minification filter reduces the small-scale edges of the lid.

All in all, most of the convex noise can be removed of the tested models, while still keeping the most important features and edges. In all cases the final reconstructions are a visible improvements compared to the noisy models.

	Resolution	Processing time (seconds)
shark	420^3	70.54
bear	276^3	86.30
bunny	216^3	132.13
cat	420^3	145.23
chair	420^3	231.64
fandisk	276^3	681.30
hammer	520^3	52.34
cube	148^3	42.34
pot	276^3	271.83

Table 8.1: For each model, the resolution and the total processing time of the adapted pipeline (including IO) is shown.



(a) Skeleton

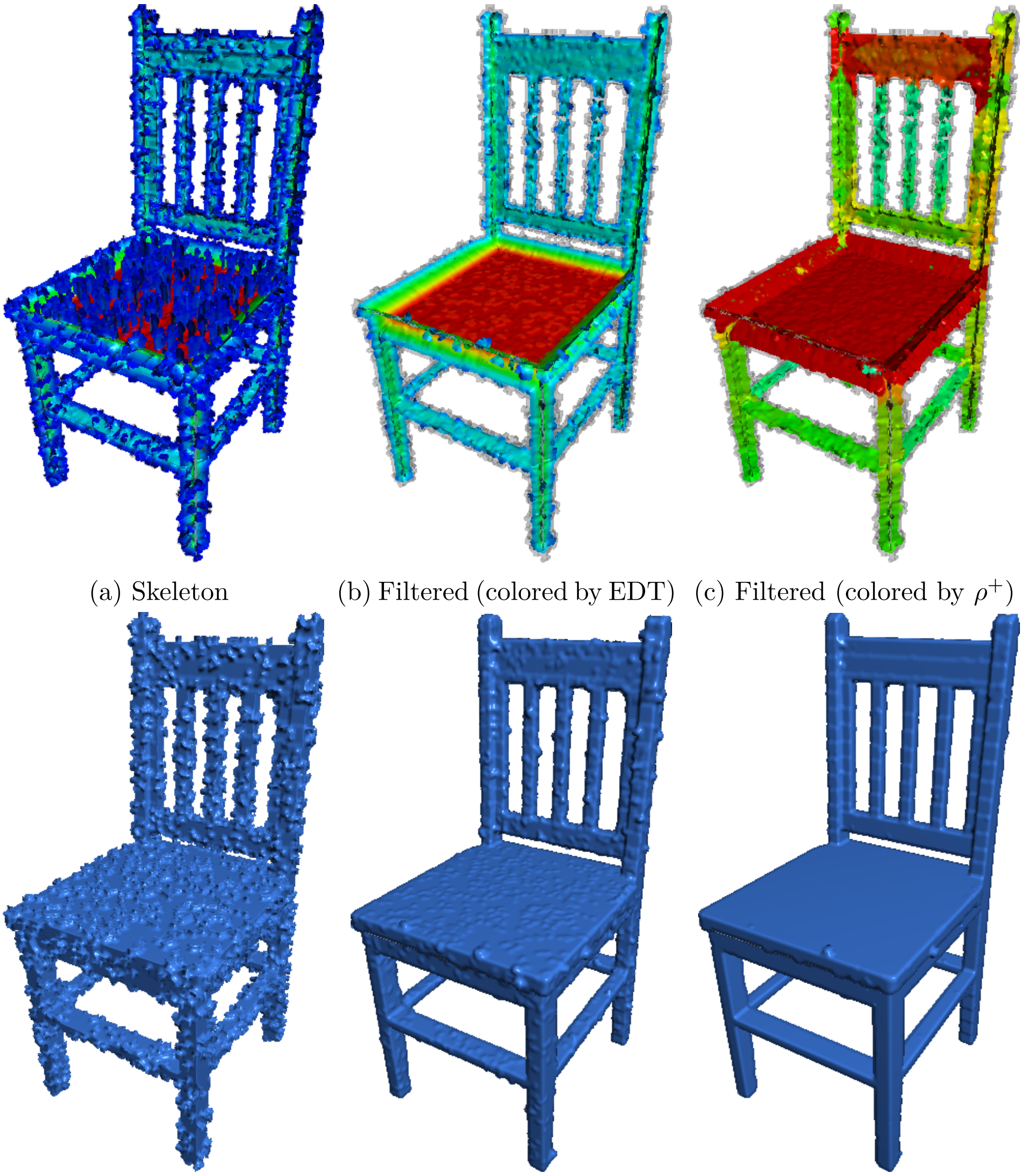
(b) Filtered (colored by EDT)

(c) Filtered (colored by ρ^+)

(d) Model

(e) Reconstruction

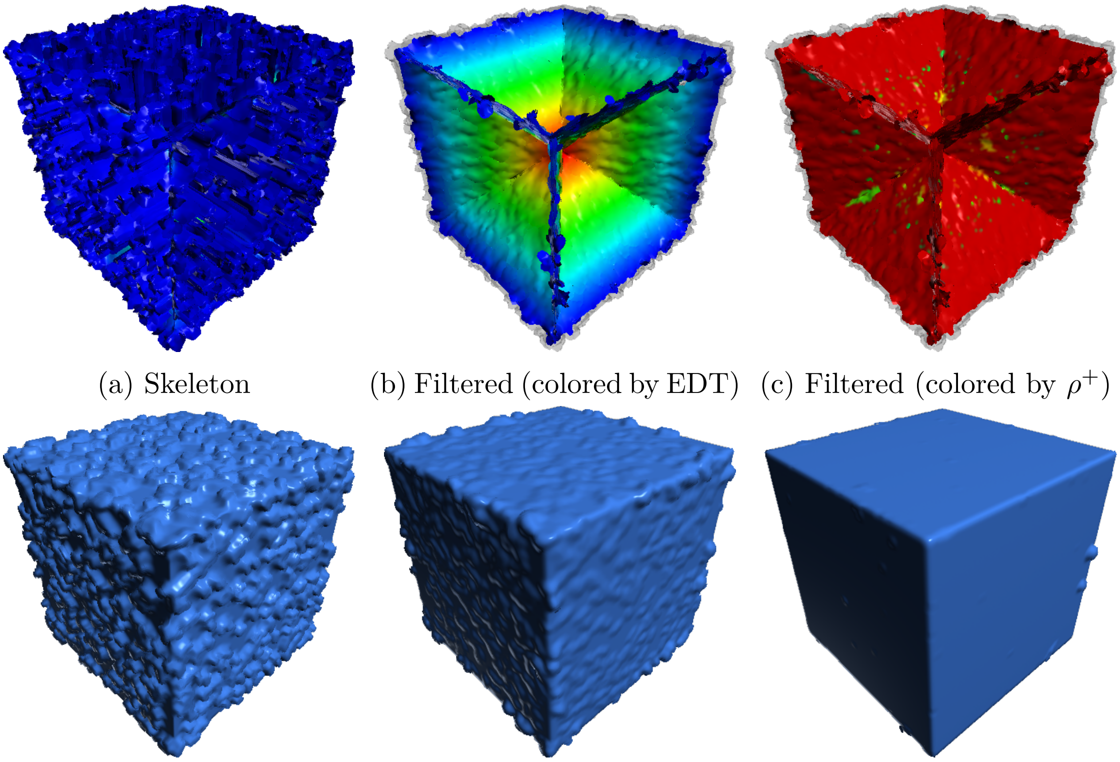
Figure 8.1: Bear model with 1-d spherical creases.



(a) Skeleton (b) Filtered (colored by EDT) (c) Filtered (colored by ρ^+)

(d) Model (e) Reconstruction (f) Min filtered

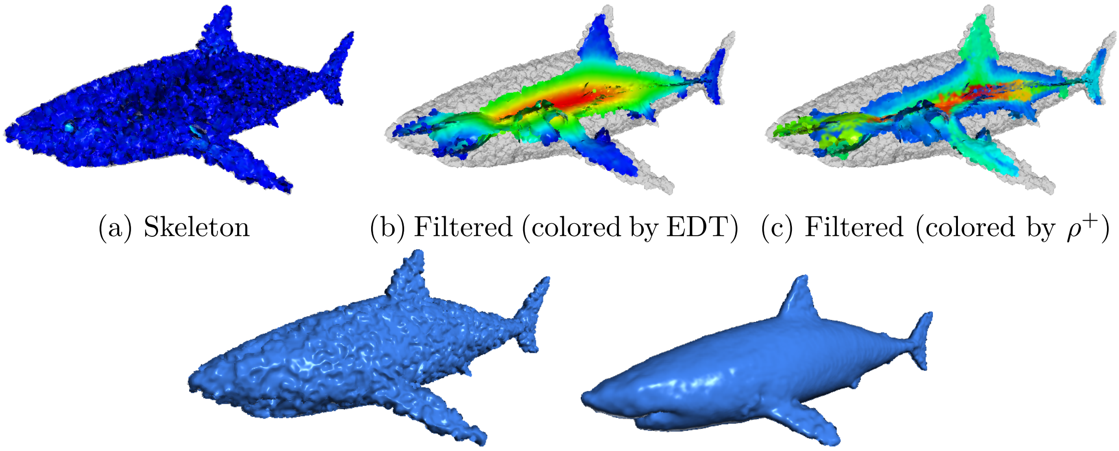
Figure 8.2: Chair model with 1-d spherical creases.



(a) Skeleton (b) Filtered (colored by EDT) (c) Filtered (colored by ρ^+)

(d) Model (e) Reconstruction (f) Min filtered

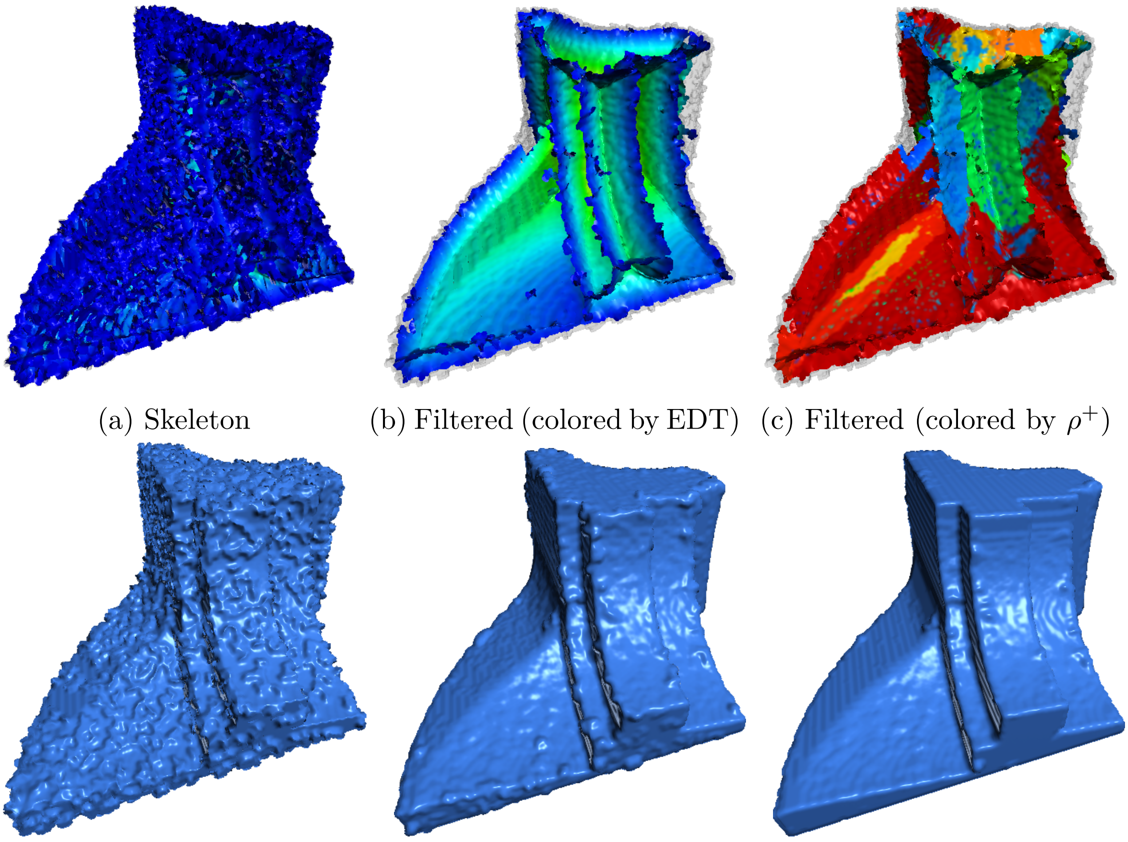
Figure 8.3: Cube model with 1-d spherical creases.



(a) Skeleton (b) Filtered (colored by EDT) (c) Filtered (colored by ρ^+)

(d) Model (e) Reconstruction

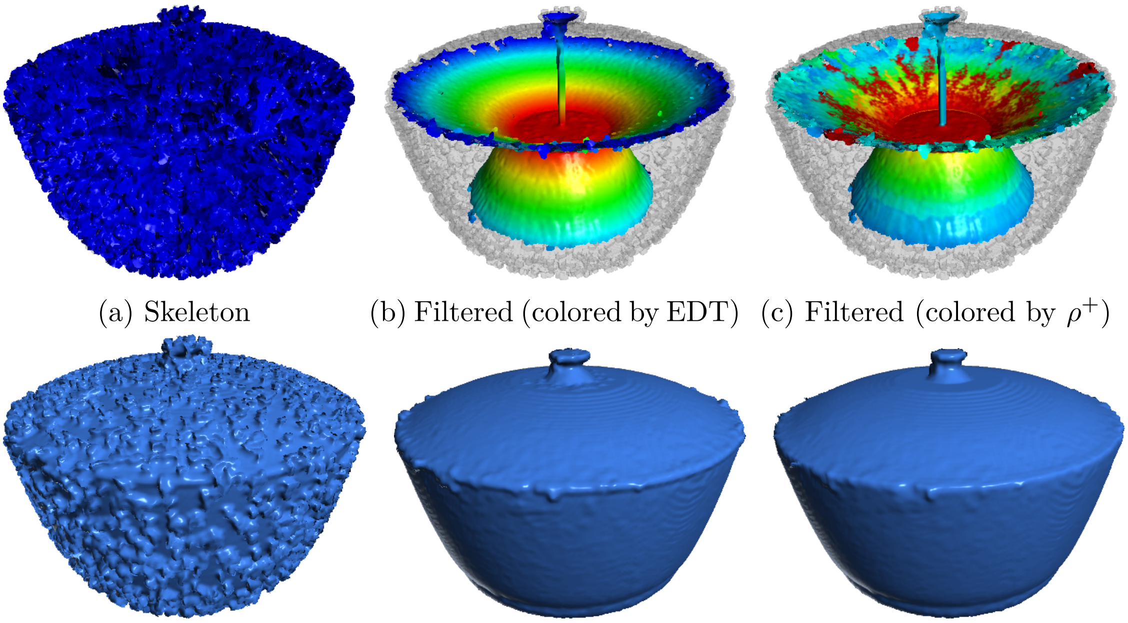
Figure 8.4: Shark model with 1-d spherical creases.



(a) Skeleton (b) Filtered (colored by EDT) (c) Filtered (colored by ρ^+)

(d) Model (e) Reconstruction (f) Min filtered

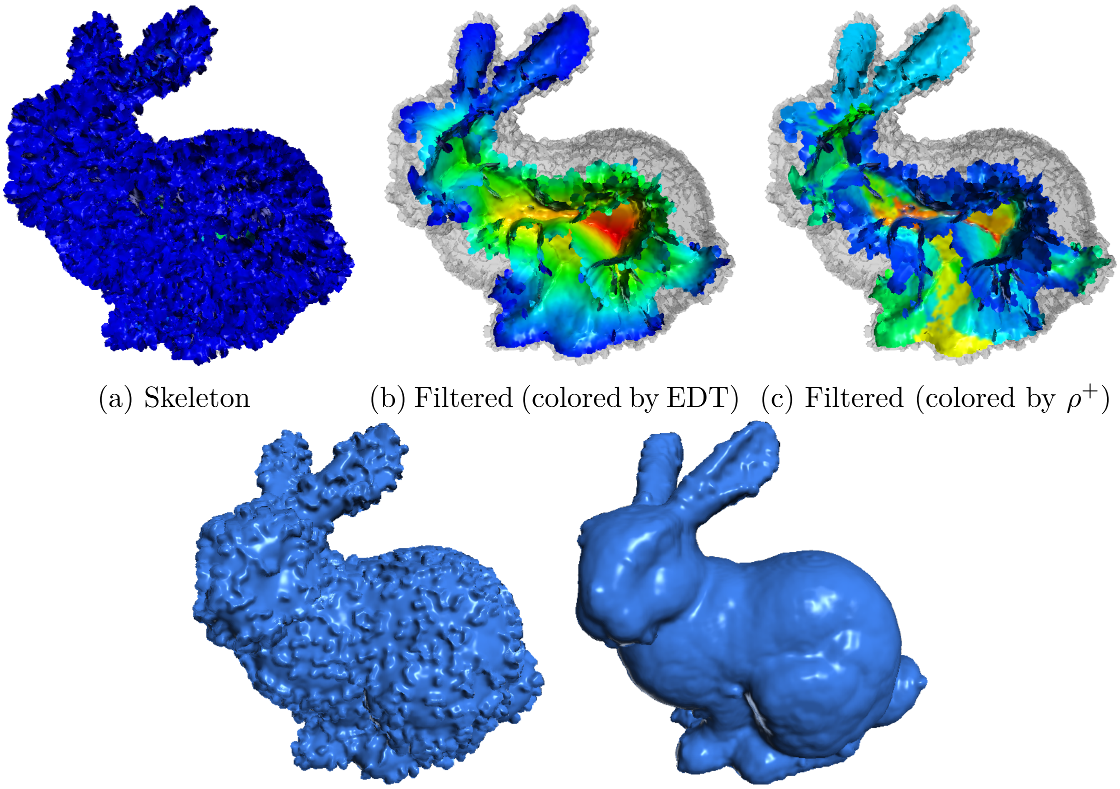
Figure 8.5: Fandisk model with 1-d spherical creases.



(a) Skeleton (b) Filtered (colored by EDT) (c) Filtered (colored by ρ^+)

(d) Model (e) Reconstruction (f) Min filtered

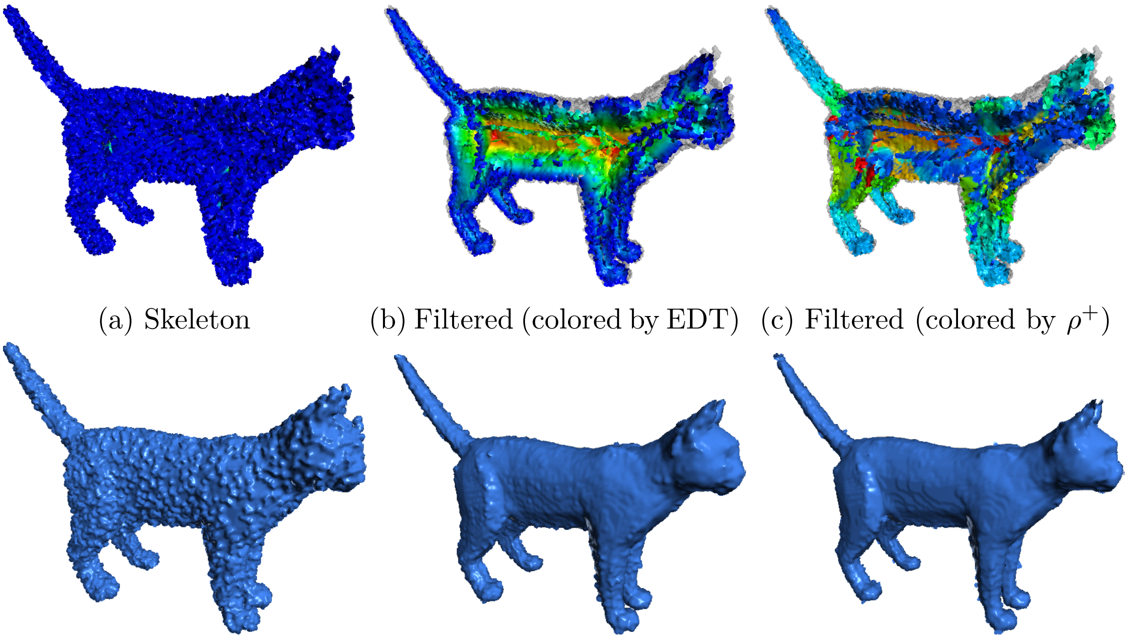
Figure 8.6: Pot model with 1-d spherical creases.



(a) Skeleton (b) Filtered (colored by EDT) (c) Filtered (colored by ρ^+)

(d) Model (e) Reconstruction

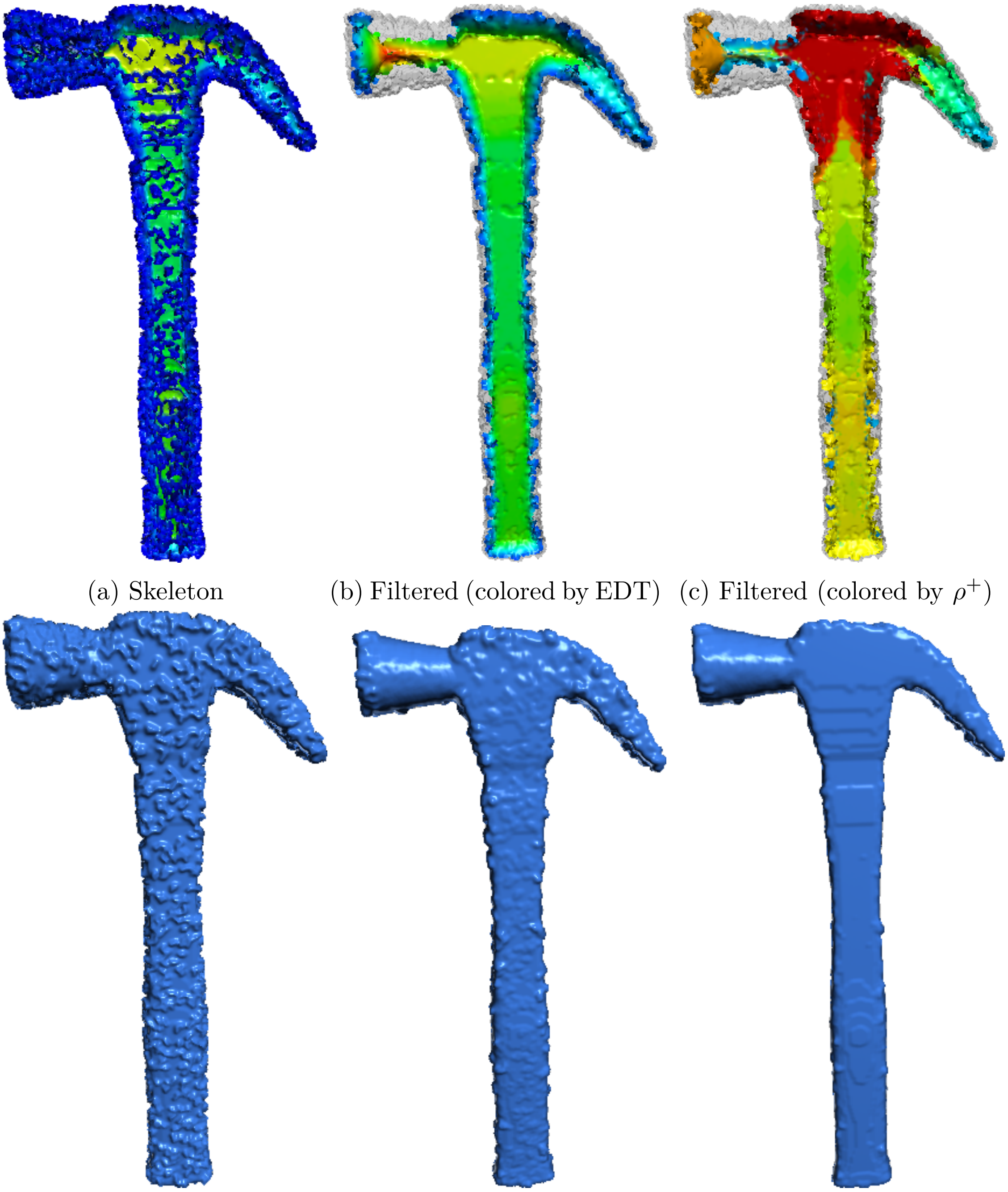
Figure 8.7: Bunny model with 1-d spherical creases.



(a) Skeleton (b) Filtered (colored by EDT) (c) Filtered (colored by ρ^+)

(d) Model (e) Reconstruction (f) Min filtered

Figure 8.8: Cat model with 1-d spherical creases.



(a) Skeleton (b) Filtered (colored by EDT) (c) Filtered (colored by ρ^{+})

(d) Model (e) Reconstruction (f) Min filtered

Figure 8.9: Hammer model with 1-d spherical creases.

8.2 Comparison

The end-results are also compared to various other triangle-based and point-based smoothing methods. Here is a list of the smoothing methods that were compared to our method:

- Laplacian smoothing (surface preserve): Simple mean filter of the local neighborhood of vertices applied iteratively.
- Two-step smoothing method [6]: It is based on a normal smoothing step where similar normals are averaged together, and a step where the vertices are fitted on the new normals. These two steps are applied iteratively in succession.
- Mean Least Squares (MLS) projection [25]: a feature-preserving point-based method based on moving least squares fitting of algebraic spheres.
- Robust Improved Mean Least-Squares (RIMLS) projection [48]: a feature-preserving point-based method that uses robust statistics (M-estimators) to create a implicit least squares procedure that has good edge-preserving qualities.

Note that this is not an exhaustive list of methods, nor are these the best feature-preserving approaches. The methods listed here are chosen because they are implemented in Meshlab, an open source solution that allows processing and editing of unstructured 3D triangular meshes. Thus, the implementations of these methods are directly available and can be quickly compared.

The last three methods listed here are feature-preserving methods. The two-step method controls te feature-preserving part by limiting the angle of the normals of the sampled neighboring vertices. The projection methods controls the feature preserving part by a scale and sharpness parameter, which respectively influences the radius of the smoothing kernel and the sharpness of the projection.

8.2.1 Results

For a selection of models (cube, bear, fandisk), synthetic 1D stripes are added using the parameters ($p = 0.014, r = 2.1, h = 3, l = 6$). On the cat model noise of higher amplitude is added using the parameters ($p = 0.01, r = 2.1, h = 7, l = 4$). For our method the end-result was extracted using the same steps and parameters as explained in section 8.1.2. The noisy models and the reconstructions results were converted to meshes using marching cubes so that it can be compared to the other methods listed above.

The four methods listed above were applied on the noisy meshes and compared to our converted reconstructed results. The parameters of the methods mostly left to the defaults of Meshlab, except for the scale and number of iterations, as leaving these to their default values will mostly keep the noise intact. This is likely because these set of parameters are hand-tuned to remove Gaussian noise in most cases, which is easier to remove, and thus requires lower scales and a smaller number of iterations. The following parameters for each methods were used to create the smoothing result:

Laplacian smoothing For the tested models the parameters which are used are (max normal deviation (deg): 60, iterations: 50). Both values are different from the defaults (resp. 0.5, and 3), as these parameters are set too low to remove the noise.

Two-step method The parameters are (smoothing steps: 40, deg: 60, normal smoothing steps: 20, vertex fitting steps: 20). The only difference with the defaults is the number of smoothing steps, which was changed from 3 to 40.

MLS The defaults were used for most of the parameters, except for the scale (where $r = 2$ is the default). This was set to $r = 15$, which is sufficiently larger than the scale of the noise, so that the method should be able to discern perturbations from flat surfaces.

RIMLS This method has proven to be unstable on some noisy models, even with the default parameters. On the fandisk- and cube- model the scale parameter was set to $r = 15$, which is the highest value which could be used before the result became unstable. On the bear- and cat- model the even default parameters were unstable, and changing these did not seem to improve the result, and therefore the method is omitted for these models.

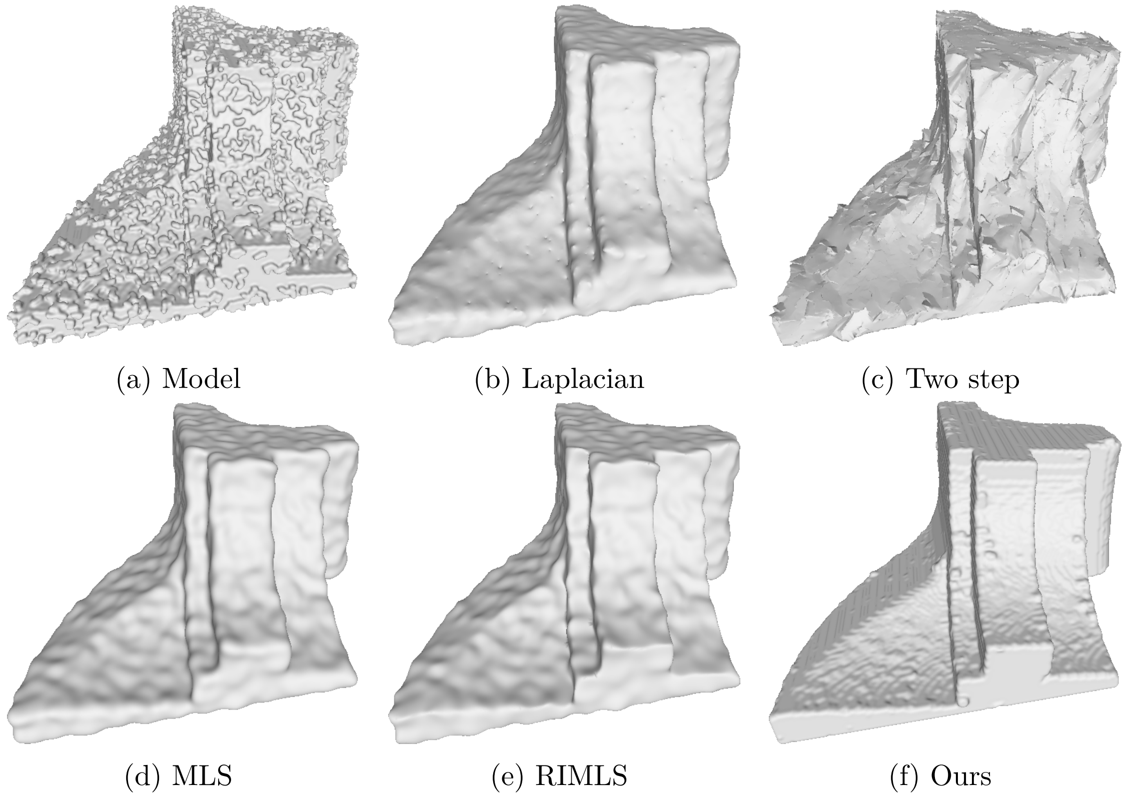
The methods are then compared by creating screencaps of the meshes using the default smooth rendering of Meshlab. The results are shown in figures 8.10 to 8.12. Here we can see that the Laplacian method is able to sufficiently smooth the model but also blends features. The results of the two-step method seem to be sharper, but it also introduces non-manifold edges and generally reduces the quality of the meshes. MLS projection preserves features better than the Laplacian method, but it also exaggerates the noise, as it is unable to make the distinction of local features to globally relevant edges. The RIMLS projection method, when stable, is sharper than MLS, but also enhances noise in a similar manner. For the cat model the we can see that higher amplitude noise affects the tested smoothing methods even more, while in our method the cat can still be cleanly extracted. This is expected, as our method is largely invariant to the amplitude of the noise, as the geodesic importance measure does not increase as h is increased.

Of the tested models the skeleton-based method removes most of the noise, while keeping the important features intact. For example, the ears of the cat, the edges of the fandisk and the edges of the cube. These feature are blurred in the other smoothing methods, but are not affected by the adapted saliency pipeline.

8.3 Conclusion

The adapted saliency pipeline has been tested on a selection of voxel models with synthetic 1D noise. With the global streamline method almost all of the ligature branches can separated from the core skeleton using the threshold $t = 18$. In most test models, with the exception of the shark model, the core skeleton is kept largely intact. The EDT minification filter was tested on a selection of models with flat surfaces. It seems to significantly improve the results compared to the raw reconstructions of these models. It was found that almost full removal of the synthetic noise is possible in most cases.

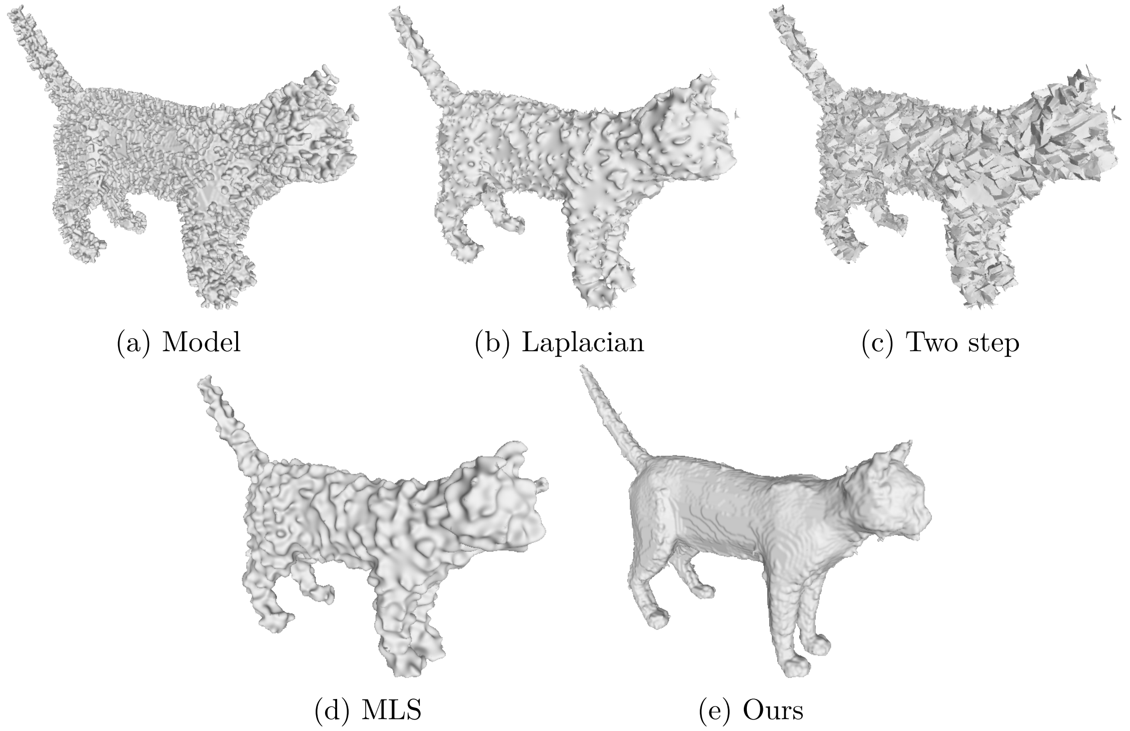
The end-results are also compared to various other triangle-based and point-based smoothing methods. In particular, Laplacian smoothing, the two-step method, MLS- and RIMLS- projection. Although these are not the state-of-the-art feature-preserving smoothing methods, the comparison gives a good indication of the feasibility of the pipeline. Compared to these methods, the adapted saliency pipeline removes by far the most noise without distorting important features, and shows that skeleton-based smoothing is feasible for 3D models.



(a) Model (b) Laplacian (c) Two step

(d) MLS (e) RIMLS (f) Ours

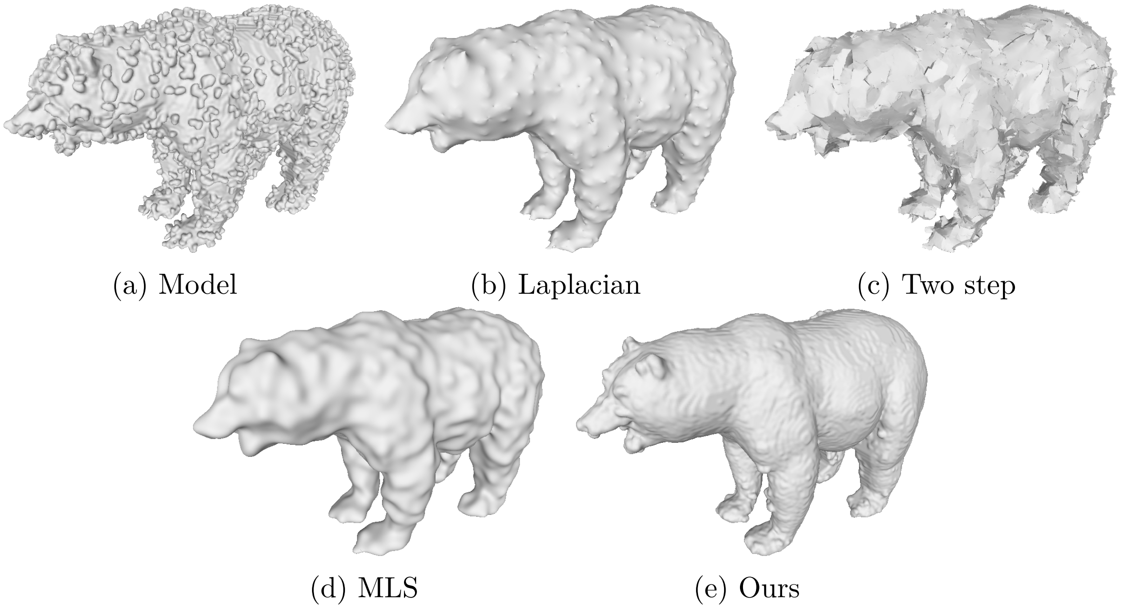
Figure 8.10: Comparison of skeleton-based smoothing to various other methods on the fandisk model.



(a) Model (b) Laplacian (c) Two step

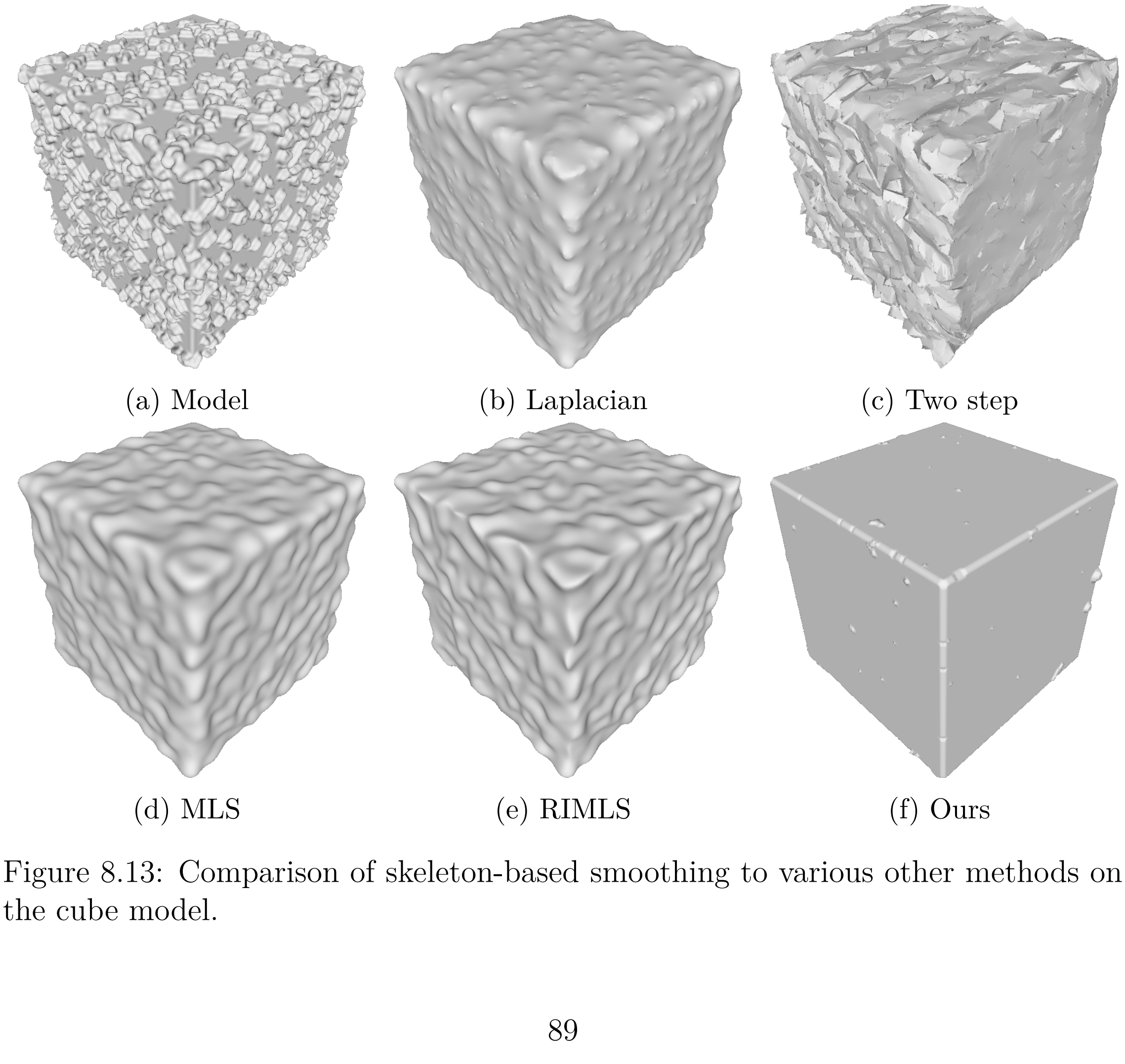
(d) MLS (e) Ours

Figure 8.11: Comparison of skeleton-based smoothing to various other methods on the cat model.


(a) Model (b) Laplacian (c) Two step

(d) MLS (e) Ours

Figure 8.12: Comparison of skeleton-based smoothing to various other methods on the bear model.


(a) Model (b) Laplacian (c) Two step

(d) MLS (e) RIMLS (f) Ours

Figure 8.13: Comparison of skeleton-based smoothing to various other methods on the cube model.

9 Concluding remarks

The feature-preserving skeleton-based saliency pipeline from Telea [58], which has shown to be successful in denoising 2D shapes, has been adapted in multiple ways to remove small-scale noise details from 3D models. The resulting pipeline allows robust removal of noise, and, compared to a selection of tested smoothing methods, it removes more noise without distorting important features. Referring back to the research question in the introduction, and the problems in the original pipeline in section 3.2, we can draw the following concluding remarks.

Low detection range problems Noise near edges are more troublesome due to lower sampling in 3D, requiring an improved detection range. This problem has largely been solved by using the derivative-based measure. Different velocity fields were compared, and the feature-based velocity field has a slight advantage on the tested models, and where finite differences of the Euclidean distance metric comes a closed second in terms of accuracy and speed. The proposed measure gives a significant improvement for 3D models compared to the original saliency measure.

Multi-axial features Important features in one axis can overlap with noise in a different axis, so that the ligature sheet is orthogonal to the core skeleton. Several approaches were tested, where the best result is achieved by using the global streamline method. The direct streamline approach seems to work well with removing most of the noise, but is too aggressive and removes important features as well as noise. The global method does not remove the ligature sheets in all cases, but does keep most of the core skeleton intact. In the final results we can separate the core skeleton from the intersecting ligature sheets in most cases.

EDT ripples After removing all ligature sheets, small noise structures on the surface still persist due to perturbations on the core skeleton and the EDT. Several filters were proposed to remove the EDT ripples: opening-based filtering, EDT minification, least squares projection, simple mean and median filters and flat-projection. From the tested methods the EDT minification method gave the best visual results. Although these filters remove noise, they all share the side-effect of reducing sharp corners. Creating a filter that preserves these corners has yet to be developed.

Skeleton method A selection of skeleton methods were evaluated and some were integrated in our pipeline, as well as different importance metrics. These methods include: A variation of the IMA skeleton [27] and the unified method [29]. It was found that an adapted IMA skeleton method with the graph geodesic measure gave the best overall result. Although the unified method is very fast, it has difficulty differentiating ligature sheets from core skeleton parts in its importance measure, and was therefore not used in the final results.

Results The complete pipeline has been tested on a selection of voxel models with synthetic noise. It was found that almost full removal of the synthetic noise is possible with the used noise parameters. The end-results are also compared to various other triangle-based and point-based smoothing methods. Compared to the tested methods, the adapted saliency pipeline removes by far the most noise without distorting important features, and shows that skeleton-based smoothing is feasible for 3D models.

9.1 Future work

Although the existing pipeline has proven to work well on synthetic noise, there is still room of improvement before it is useful for practical applications.

Input The potential of the noise removal of this pipeline is severely reduced by the input format. As the pipeline is only applicable to voxel models; it does not have a direct way to deal with meshes or point clouds. Yet most models that are not synthetic come from 3D scanners which only output point clouds. To remove noise in these cases the models have to be voxelized, which due to its dense format takes more memory, and ultimately puts a practical upper limit of the resolution.

To overcome this there are multiple solutions: one could use the skeleton methods that are directly applicable to meshes or point clouds [10]. This requires to embed a connectivity graph using local neighborhood structures which can introduce new problems on its own [45]. An alternative approach is to use a hybrid data structure as introduced by Lindblad and Sladoje [40], where grid-line sampling is used to create subvoxel precise sampling. Here each coordinate of a point can be assigned any real value, whereas other coordinates are restricted to discrete sets of values. Its advantage is a much more accurate boundary representation, and subsequently accurate EDT approximations and better reconstructions. This can be integrated with our pipeline without any structural changes, and is therefore a good candidate to improve our result.

Performance All of the used algorithms in the proposed pipeline can be theoretically implemented in $O(n)$, with the exception of computing the graph geodesics, where n is the number of input voxels. The current implementation of the extended feature transform, however, uses a k-d tree to find the closest feature points, which causes the implementation to scale superlinearly with respect to n. This is something that can be improved, by using the extended feature transform proposed by Hesselink and Roerdink [27], which is consequently also a step which is used to compute the IMA skeleton.

EDT processing From the tested methods the EDT minification method gave the best results, but it is not feature-preserving. To improve on this result some method that smooths the surface skeleton as well as the EDT must be developed. A key observation is that EDT ripples are most apparent on flat surfaces, and therefore methods that try to explicitly detect the perturbations using the regularized skeleton should give the best outcome. One possible solution might be to measure the area affected by the removal of skeleton sheets, and reduce the EDT of core skeleton parts neighboring the removed sheets by a proportional amount. This can be done in addition to bending the core skeleton parts in the opposite direction of the ligature sheets.

Skeleton methods The ideal surface skeleton method for our use-case should have a monotonically increasing metric, should be fast to compute, should be able to differentiate ligature sheet from the core skeleton, should handle models with holes and allow accurate reconstructions. Despite of a large variety of published methods, a voxel-based method with these requirements does not seem to exist yet, and should be developed.

Medial scaffold For 2D skeletons branch-based pruning methods are highly effective at regularizing skeletons [42]. This is because branches can be removed as a whole instead of partially, and therefore this technique does not suffer from shortening important branches. Although surface skeletons do not share the same simple topology as 2D skeletons, they can be organized in a similar manner, namely the medial axis hypergraph introduced by Giblin and Kimia [21]. This classification has been used by Leymarie and Kimia [38] to propose the notion of a medial scaffold(MS), a hierarchical representation of the surface skeleton points, which can be seen as the 3D version of the splice transform. The theoretical classification of the instabilities of this structure, or transitions (sudden topological changes due to a small perturbation) can be used to mark boundaries of skeleton sheets and curves. This was used effectively for regularizing the skeleton of meshes [9]. It might be interesting to integrate the detection part of their method and use it to remove ligature sheets as a whole instead of removing individual voxels. This would allow us to simplify our method, as we would no longer need a selection operator, or apply saliency post-processing in our pipeline.

Bibliography

[1] C. Arcelli and G. S. D. Baja. A width-independent fast thinning algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (4):463–474, 1985.

[2] C. Arcelli, D. Baja, and G. Sanniti. A one-pass two-operation process to detect the skeletal pixels on the 4-distance transform. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(4):411–414, 1989.

[3] C. Aslan, A. Erdem, E. Erdem, and S. Tari. Disconnected skeleton: Shape at its absolute scale. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(12):2188–2203, 2008.

[4] D. Attali, J.-D. Boissonnat, and H. Edelsbrunner. Stability and computation of medial axes-a state-of-the-art report. *Mathematical foundations of scientific visualization, computer graphics, and massive data exploration*, 6:109–125, 2009.

[5] X. Bai, L. J. Latecki, and W.-Y. Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(3):449–462, 2007.

[6] A. Belyaev and Y. Ohtake. A comparison of mesh smoothing methods. In *Israel-Korea Bi-national conference on geometric modeling and computer graphics*, volume 2. Citeseer, 2003.

[7] J. W. Brandt and V. R. Algazi. Continuous skeleton computation by voronoi diagram. *CVGIP: Image understanding*, 55(3):329–338, 1992.

[8] E. J. Breen and R. Jones. Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding*, 64(3):377–389, 1996.

[9] M.-C. Chang and B. B. Kimia. Regularizing 3d medial axis using medial scaffold transforms. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[10] M.-C. Chang, F. F. Leymarie, and B. B. Kimia. Surface reconstruction from point clouds by transforming the medial scaffold. *Computer Vision and Image Understanding*, 113(11):1130–1146, 2009.

[11] W.-P. Choi, K.-M. Lam, and W.-C. Siu. Extraction of the euclidean skeleton based on a connectivity criterion. *Pattern Recognition*, 36(3):721–729, 2003.

[12] U. Clarenz, M. Rumpf, and A. Telea. Robust feature detection and local classification for surfaces based on moment analysis. *IEEE Transactions on Visualization and Computer Graphics*, 10(5):516–524, 2004.

[13] D. Coeurjolly and A. Montanvert. Optimal separable algorithms to compute the reverse euclidean distance transformation and discrete medial axis in arbitrary dimension. *arXiv preprint arXiv:0705.3343*, 2007.

[14] S. Deutsch and G. Medioni. Unsupervised learning using the tensor voting graph. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 282–293. Springer, 2015.

[15] G. S. di Baja. Well-shaped, stable, and reversible skeletons from the (3, 4)-distance transform. *Journal of visual communication and image representation*, 5(1):107–115, 1994.

[16] G. S. Di Baja and E. Thiel. Skeletonization algorithm running on path-based distance maps. *Image and vision Computing*, 14(1):47–57, 1996.

[17] J. Digne, D. Cohen-Steiner, P. Alliez, F. De Goes, and M. Desbrun. Feature-preserving surface reconstruction and simplification from defect-laden point sets. *Journal of mathematical imaging and vision*, 48(2):369–382, 2014.

[18] P. Dimitrov, J. N. Damon, and K. Siddiqi. Flux invariants for shape. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–835. IEEE, 2003.

[19] M. Etzion and A. Rappoport. Computing voronoi skeletons of a 3-d polyhedron by space subdivision. *Computational Geometry*, 21(3):87–120, 2002.

[20] Y. Ge and J. M. Fitzpatrick. On the generation of skeletons from discrete euclidean distance maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(11):1055–1066, 1996.

[21] P. Giblin and B. B. Kimia. A formal classification of 3d medial axis points and their local geometry. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):238–251, 2004.

[22] P. J. Giblin and B. B. Kimia. On the local form of symmetry sets, and medial axes, and shocks in 3d. In *Proc. of CVPR*, pages 566–573, 2000.

[23] P. J. Giblin and B. B. Kimia. On the local form and transitions of symmetry sets, medial axes, and shocks. *International Journal of Computer Vision*, 54 (1-3):143–157, 2003.

[24] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt. Shape representation and classification using the poisson equation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):1991–2005, 2006.

[25] G. Guennebaud and M. Gross. Algebraic point set surfaces. In *ACM Transactions on Graphics (TOG)*, volume 26, page 23. ACM, 2007.

[26] L. He and S. Schaefer. Mesh denoising via l 0 minimization. *ACM Transactions on Graphics (TOG)*, 32(4):64, 2013.

[27] W. H. Hesselink and J. B. Roerdink. Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(12):2204–2217, 2008.

[28] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. In *ACM transactions on graphics (TOG)*, volume 28, page 176. ACM, 2009.

[29] A. Jalba, A. Sobiecki, and A. Telea. An unified multiscale framework for planar, surface, and curve skeletonization.

[30] L. Ji and J. Piper. Fast homotopy-preserving skeletons using mathematical morphology. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):653–664, 1992.

[31] T. R. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 943–949. ACM, 2003.

[32] S. Krinidis and V. Chatzis. A skeleton family generator via physics-based deformable models. *Image Processing, IEEE Transactions on*, 18(1):1–11, 2009.

[33] L. Lam, S.-W. Lee, and C. Y. Suen. Thinning methodologies-a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, 14(9): 869–885, 1992.

[34] C. Lange and K. Polthier. Anisotropic smoothing of point sets. *Computer Aided Geometric Design*, 22(7):680–692, 2005.

[35] A. Leborgne, J. Mille, and L. Tougne. Extracting noise-resistant skeleton on digital shapes for graph matching. In *Advances in Visual Computing*, pages 293–302. Springer, 2014.

[36] T.-C. Lee, R. L. Kashyap, and C.-N. Chu. Building skeleton models via 3-d medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, 1994.

[37] F. Leymarie and M. D. Levine. Simulating the grassfire transform using an active contour model. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (1):56–75, 1992.

[38] F. F. Leymarie and B. B. Kimia. The medial scaffold of 3d unorganized point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 (2):313–330, 2007.

[39] B. Li, R. Schnabel, R. Klein, Z. Cheng, G. Dang, and S. Jin. Robust normal estimation for point clouds with sharp features. *Computers & Graphics*, 34(2): 94–106, 2010.

[40] J. Lindblad and N. Sladoje. Exact linear time euclidean distance transforms of grid line sampled shapes. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 645–656. Springer, 2015.

[41] H. Liu, Z. Wu, D. F. Hsu, B. S. Peterson, and D. Xu. On the generation and pruning of skeletons using generalized voronoi diagrams. *Pattern Recognition Letters*, 33(16):2113–2119, 2012.

[42] H. Liu, Z.-H. Wu, X. Zhang, and D. F. Hsu. A skeleton pruning algorithm based on information fusion. *Pattern Recognition Letters*, 34(10):1138–1145, 2013.

[43] P.-C. Liu, F.-C. Wu, W.-C. Ma, R.-H. Liang, and M. Ouhyoung. Automatic animation skeleton using repulsive force field. In *Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on*, pages 409–413. IEEE, 2003.

[44] P. Maragos and R. D. Ziff. Threshold superposition in morphological image analysis systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(5):498–504, 1990.

[45] R. R. Moelker. Edge preserving salience based smoothing of 3d shapes skeletons, 2014.

[46] R. Ogniewicz and M. Ilg. Voronoi skeletons: Theory and applications. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 63–69. IEEE, 1992.

[47] R. L. Ogniewicz and O. Kübler. Hierarchic voronoi skeletons. *Pattern recognition*, 28(3):343–359, 1995.

[48] A. C. Öztireli, G. Guennebaud, and M. Gross. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer Graphics Forum*, volume 28, pages 493–501. Wiley Online Library, 2009.

[49] D. Reniers. *Skeletonization and segmentation of binary voxel shapes.* PhD thesis, Technische Universiteit Eindhoven, 2009.

[50] D. Reniers, A. Jalba, and A. Telea. Robust classification and analysis of anatomical surfaces using 3d skeletons. In *Proceedings of the First Eurographics conference on Visual Computing for Biomedicine*, pages 61–68. Eurographics Association, 2008.

[51] D. Reniers, J. J. Van Wijk, and A. Telea. Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. *Visualization and Computer Graphics, IEEE Transactions on*, 14(2):355–368, 2008.

[52] C. Ronse. Set-theoretical algebraic approaches to connectivity in continuous or digital spaces. *Journal of Mathematical Imaging and Vision*, 8(1):41–58, 1998.

[53] W. Shen, X. Bai, R. Hu, H. Wang, and L. J. Latecki. Skeleton growing and pruning with bending potential ratio. *Pattern Recognition*, 44(2):196–209, 2011.

[54] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein. Noise in 3d laser range scanner data. In *Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on*, pages 37–45. IEEE, 2008.

[55] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, and A. Telea. 3d skeletons: A state-of-the-art report. In *Computer Graphics Forum*, volume 35, pages 573–597. Wiley Online Library, 2016.

[56] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM, 1995.

[57] H. Tek and B. B. Kimia. Boundary smoothing via symmetry transforms. *Journal of Mathematical Imaging and Vision*, 14(3):211–223, 2001.

[58] A. Telea. Feature preserving smoothing of shapes using saliency skeletons. In *Visualization in Medicine and Life Sciences II*, pages 153–170. Springer, 2012.

[59] A. Telea and J. J. Van Wijk. An augmented fast marching method for computing skeletons and centerlines. In *Proceedings of the symposium on Data Visualisation 2002*, pages 251–ff. Eurographics Association, 2002.

[60] E. R. Urbach, J. B. T. M. Roerdink, and M. H. F. Wilkinson. Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29 (2):272–285, 2007.

[61] G. Vialaneix and T. Boubekeur. Sbl mesh filter: fast separable approximation of bilateral mesh filtering. In *ACM SIGGRAPH 2011 Talks*, page 24. ACM, 2011.

[62] M. Wei, W. Shen, J. Qin, J. Wu, T.-T. Wong, and P.-A. Heng. Feature-preserving optimization for noisy mesh using joint bilateral filter and constrained laplacian smoothing. *Optics and Lasers in Engineering*, 51(11):1223–1234, 2013.

[63] Z. Wood, H. Hoppe, M. Desbrun, and P. Schröder. Removing excess topology from isosurfaces. *ACM Transactions on Graphics (TOG)*, 23(2):190–208, 2004.

[64] Z. Yu and C. Bajaj. A segmentation-free approach for skeletonization of gray-scale images via anisotropic vector diffusion. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–415. IEEE, 2004.

[65] W. Zhang, B. Deng, J. Zhang, S. Bouaziz, and L. Liu. Guided mesh normal filtering. In *Computer Graphics Forum*, volume 34, pages 23–34. Wiley Online Library, 2015.