SKELETONIZATION OF 2D AND 3D SHAPES VIA INCREMENTAL CARVING OF THE SHAPE DOMAIN

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÜRKAN ÇAĞLAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JANUARY 2023

Approval of the thesis:

SKELETONIZATION OF 2D AND 3D SHAPES VIA INCREMENTAL CARVING OF THE SHAPE DOMAIN

submitted by **GÜRKAN ÇAĞLAR** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering**

Prof. Dr. Sibel Tarı
Supervisor, **Computer Engineering, METU**

Assist. Prof. Dr. Aslı Gençtav
Co-supervisor, **Computer Engineering, TED University**

Examining Committee Members:

Assist. Prof. Dr. Emre Akbaş
Computer Engineering Department, METU

Prof. Dr. Sibel Tarı
Computer Engineering Department, METU

Prof. Dr. Tolga Kurtuluş Çapın
Computer Engineering Department, TED University

Date:24.01.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Gürkan Çağlar

Signature :

ABSTRACT

SKELETONIZATION OF 2D AND 3D SHAPES VIA INCREMENTAL CARVING OF THE SHAPE DOMAIN

Çağlar, Gürkan
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Sibel Tarı
Co-Supervisor: Assist. Prof. Dr. Aslı Gençtav

January 2023, 59 pages

Extracting skeletons of 2D and 3D shapes is important for obtaining a compact representation of shapes, which later can be utilized in various areas such as shape matching, retrieval, deformation, animation, medical imaging, virtual endoscopy and so on. In this thesis, we present a new family of smooth distance transforms what we call as v_k field, which enables exploration of the shape domain by incrementally carving it from inside out. Our field is applicable to both 2D and 3D shapes. After analyzing expansion of v_k field inside the shape domain, we develop a method for constructing curve skeleton of 2D and 3D shapes. Our skeletonization method does not require thinning and the skeletons produced by our method are thin and robust to noise. We present and discuss our skeletonization results for several 2D and 3D shapes with comparison to other skeletonization methods.

Keywords: Shape Representation, Skeletonization, Curve Skeletons

ÖZ

ŞEKİL ALANININ ARTIMLI OYULMASI YOLUYLA 2B VE 3B ŞEKİLLERİN İSKELETLEŞTİRİLMESİ

Çağlar, Gürkan
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Sibel Tarı
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi. Aslı Gençtav

Ocak 2023 , 59 sayfa

2B ve 3B şekillerin iskeletlerini çıkarmak, daha sonra şekil eşleştirme, geri alma, deformasyon, animasyon, tıbbi görüntüleme, sanal endoskopi ve benzeri çeşitli alanlarda kullanılabilen, şekillerin kompakt bir temsilini elde etmek için önemlidir. Bu tezde, v_k alanı olarak adlandırdığımız ve şekil alanının içten dışa doğru adım adım oyularak keşfedilmesini sağlayan yeni bir yumuşak mesafe dönüşümleri ailesi sunuyoruz. Alanımız hem 2B hem de 3B şekillere uygulanabilir. Şekil alanı içindeki v_k alanının genişlemesini analiz ettikten sonra, 2B ve 3B şekillerin eğri iskeletini oluşturmak için bir yöntem geliştirdik. İskeletleştirme yöntemimiz inceltme gerektirmez ve yöntemimizle üretilen iskeletler ince ve gürültüye karşı dayanıklıdır. Çeşitli 2B ve 3B şekiller için iskeletleştirme sonuçlarımızı diğer iskeletleştirme yöntemleriyle karşılaştırmalı olarak sunuyor ve tartışıyoruz.

Anahtar Kelimeler: Şekil iskelet çıkarma, eğri-iskelet, iskeletlendirme

To my wife and family

ACKNOWLEDGMENTS

I would like to send my deep gratitude to everyone who helped me and supported me while conducting this work and writing this thesis.

First of all, I would like to thank Prof. Dr. Sibel Tarı, my advisor, and Assist. Prof. Dr. Aslı Gençtav, my co-advisor, for accepting me as their student, they taught me so much and without their continuous support and guidance this thesis would not have been possible. Talking about guidance, I would also like to thank my thesis committee members Assist. Prof. Dr. Emre Akbaş and Prof. Dr. Tolga Kurtuluş Çapın, for their positive encouragement and constructive comments. Also I am grateful to Assist. Prof. Dr. Venera Adanova for her valuable comments and support.

I am deeply indebted to my friend İbrahim Mert Koç for his continuous support and the time he spent to brainstorm with me. I can not skip this section without mentioning my dear friends, Can Ünaldı and Göknur Ercan throughout this thesis their helps were much obliged.

I would like to offer my special thanks to my family, my entire life they supported me, and help me become a better version of myself.

And finally, I would like to thank my wife Tuçe for her everlasting support, without her presence I would not be able to accomplish this study.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
PDE	Partial Differential Equation
MAT	Medial Axis Transform
GPU	Graphics Processing Unit

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Skeletons of shapes are one of the most prominent topics for image processing and computer vision. The importance of the topic stems from the demand to extract compact and useful features of the 2D and 3D shapes, namely, their skeletons. Many fields from various disciplines (e.g., animation, medical imaging, virtual endoscopy, shape matching, shape retrieval, shape tracking, shape manipulation, shape recognition, etc.) implement a solution for skeleton extraction. Extracting the skeleton of a shape is also called "skeletonization".

Skeletonization is the task of constructing the distilled features of a shape, namely, its descriptor. An analogy, which is referred to ubiquitously in this field, comes from [8]; the grass fire analogy, in which a dry grass field, burning from its boundary, forms the skeleton when the fire fronts meet and quench at certain places. Although the field of skeletonization of 2D shapes is quite mature and well-researched, that of skeletonization of 3D shapes has a lot of unexplored research topics; therefore, the field of 3D shapes still needs further research. Specifically, obtaining the most compact representation of an arbitrary 3D object has proved itself to be a challenging problem; this representation is called a "curve-skeleton" of a shape.

As explained in [9], "curve-skeletons are a subset of the medial surface of a 3D object", which could be beneficial for shape demonstration and representation; for example, reduced-model formulation, visualization enhancement, mesh repair, and virtual navigation. The difficulty and so-called “ill-defined" objective of the curve-skeleton extraction has forced the literature to focus on very specific shapes with limited robustness, narrow-scope heuristics, and unclear generalization performances. In other words, some methods demonstrate remarkable results on tubular shapes, while others outperform for shapes with vast areas. Some methods show counter-intuitive under-performance for trivial shapes (e.g. a rectangular slab) whose curve-skeleton can be very well-defined by their geometric structure. Some methods are unable to generalize for both 2D and 3D objects or when noise is introduced to the shape.

Our method, on the other hand, is applicable to both 2D and 3D shapes, and it is robust to boundary texture/noise. Furthermore, it has the capability of representing the boundary texture/noise with small skeletal segments. The main tool in our method is a smooth distance field defined on the shape domain where iso-curves/iso-surfaces of the field provide useful information about the shape. We start with computing the smooth field on the whole shape domain. Then, we carve the shape domain from the point at which the smooth field has its maximum value, which roughly marks the innermost point of the shape. After computing the smooth field on the carved shape domain, we expand the carving according to the hills and valleys of the field between the carved region inside the shape and the background region outside the shape. The iso-curves/iso-surfaces of the smooth field determine not only the expansion of the carving but also the constituent parts of the shape remaining after carving. In the first row of Figure 1.1, we illustrate the expansion of the carving at sample levels for an example 3D shape. In the second row of Figure 1.1, we see parts of the shape outside the carved region at these sample levels where the parts are obtained via iso-surfaces of the smooth field computed on the corresponding shape domain remaining after carving. Notice that, at level 3 shown in Figure 1.1 (a), there are two parts corresponding to the front and back of the shape. As the level increases, these parts are partitioned further (see level 5 in Figure 1.1 (c)). We construct a skeletal representation of a 2D/3D shape by considering the relations among the parts of the shape outside the carved region at consecutive levels. Our skeletonization method involves four steps, namely, determining skeleton points, linking the skeleton points at consecutive levels, pruning the links from the skeletal branches modeling the boundary texture/noise to the main skeletal graph, and re-connecting the skeletal branches modeling the shape regions with a significant size determined by a parameter κ. In Figure 1.2, we present our skeletonization result for the 3D shape in Figure 1.1 af-

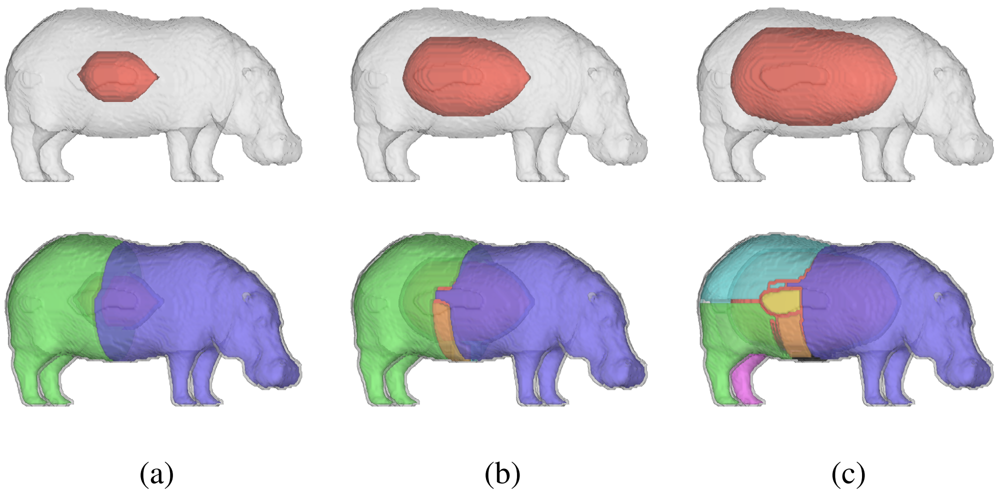


Figure 1.1: For a 3D hippopotamus shape, the carved region is visualized in the first row and parts of the shape outside the carved region are visualized in the second row at levels (a) 3, (b) 4, and (c) 5.

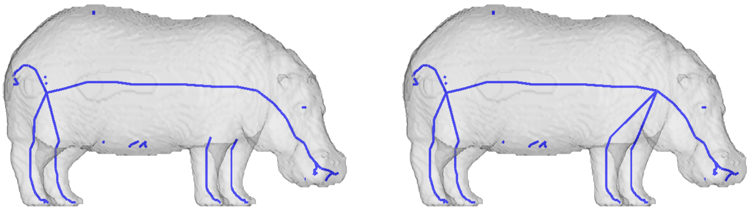


Figure 1.2: Our skeletonization result for the 3D hippopotamus shape (a) after pruning, and (b) after re-connection ($\kappa = 1/50$).

ter pruning and re-connection steps. We see that all components of the shape are successfully captured by our curve skeleton result.

Curve skeletons of 3D shapes can have several properties that make them "good" representations. However, whether a curve skeleton is good or not depends on the needs of the application in which it is to be used [10]. For example, curve skeletons could be invariant under isometric transformations, which means that the skeleton of the transformed shape should be compatible with the skeleton of the original shape. This property is supported by our method and it is especially important for shape matching applications in which the curve skeleton is used as a shape descriptor. Thinness is another property meaning that the curve skeleton should be one voxel wide in all of its length. This property is satisfied by our method and it is important for virtual navigation applications such as virtual endoscopy. Another important property for curve skeletons and also for virtual endoscopy is the smoothness of the skeletal curves that determines the movement of the camera as it could cause abrupt changes in the camera view. Considering our method, the smoothness of the curve skeletons varies based on the shape. For some tubular shapes, our results are smooth whereas, for others, it is not. Our method does not guarantee the smoothness of the resulting skeleton. A property that our method complies well with is robustness, which means that the curve skeleton should not be affected by the boundary noise and the resulting skeleton should be similar for the original shape and its noisy version. Homotopy is another property meaning that both the shape and its skeleton should be topologically equivalent i.e. they should have the same number of tunnels, connected components, and cavities [11]. This property is not yet supported by our method but making it homotopic is future work. Reliability is another property referring that every boundary point is visible from at least one curve skeleton location. The next property is centeredness where, for most applications like virtual navigation and animation, relaxed centeredness is enough. The last property is connectedness. For our method, any skeletal segments not connected to the main skeletal graph are regarded as boundary texture/noise and we could easily eliminate these segments.

1.2 Contributions

Our contributions are as follows:

- We present a new family of smooth distance fields that we call as v_k field for exploring the shape domain by incrementally carving it from inside out.
- v_k field is applicable to both 2D and 3D shapes.
- v_k field at a particular level shows the part-coding behavior of the smooth distance transform [1, 2] proposed by Tarı in 2009.
- By using v_k field, we develop a new method for constructing curve skeleton of 2D and 3D shapes.

- Our skeleton model is robust to boundary texture/noise.
- Our skeletonization method produces thin curve skeletons without requiring a thinning procedure.

1.3 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we review the literature on extracting curve skeletons of 3D shapes. In Chapter 3, we present v_k field as a new family of smooth distance transforms providing exploration of the shape domain by incrementally carving it from inside out. In Chapter 4, we present how we construct curve skeleton of 2D and 3D shapes using v_k field. In Chapter 5, we present and discuss results of our curve skeleton method for 2D and 3D shapes. Finally, conclusion and future work for our method are given in Chapter 6.

CHAPTER 2

LITERATURE REVIEW ON CURVE SKELETON OF 3D SHAPES

In this section, we discuss some of the previous works on the extraction of curve skeletons of 3D shapes. We categorize the methods under the sections defined by [11], namely, thinning methods, geometric methods, and distance or general field methods. However, the defined categories are not always mutually exclusive, nor they are isolated; it is possible that methods from different categories can be orchestrated into one method for curve-skeleton extraction.

2.1 Thinning Methods

Thinning methods work by removing excess voxels (only work on volumes but not meshes) while preserving the topology of the shape with the given set of rules. These methods rely on the concept of a simple point introduced in [12], which means the expandable point/voxel that does not damage the topology of the shape if it is removed. The fact that these points are locally calculated makes the thinning methods fast. Thinning methods start from the boundary and stop when there are no more simple points to be removed. The removal of the simple points could result in a short skeleton because the endpoints of the skeleton are also simple points [11]. One shortcoming of these methods is that they do not guarantee centeredness since they are voxel-based skeleton extraction methods and the center is sometimes decided between two voxels.

Thinning methods can utilize directional thinning, which implies that there is a certain order to remove voxels while thinning and the order could not be skipped. To elaborate on the directional thinning, an example can be given from the paper [13] which has the order: "US; NE; WD; ES; UW; ND; SW; UN; ED; NW; UE; SD" where the

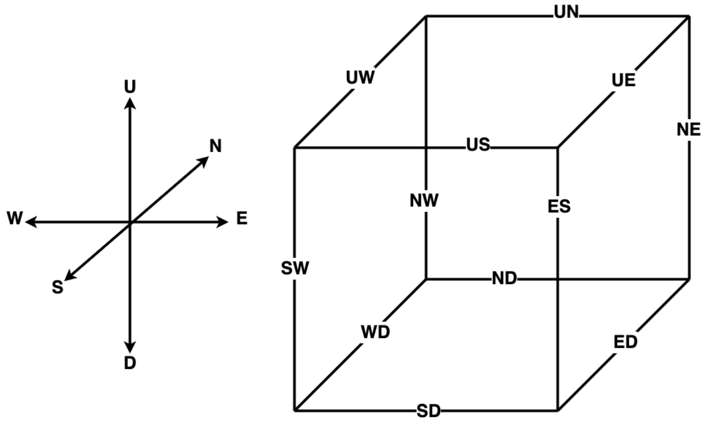


Figure 2.1: Direction Axis on the left, voxel edge naming on the right.

naming of the directions is shown in Figure 2.1. With this extended order (previous methods work with six directions) and with the parallel nature of the method, they manage to preserve the topology of the shapes.

Another shortcoming of the topological thinning methods, as are the others, are that resulting skeletons are affected by noise but, in [14], they develop a method that alternates between thinning and skeleton pruning. The candidates of the skeleton pruning in this method are short curve branches, small surface branches, jagged surface borders, and narrow surface bands. The handicap of this method is that it expects two specific user inputs related to the size of the curve and surface features but the method performs well on cylindrical and plate-like shapes.

In another work [15] of this section, the authors propose a method that formulates a measure for skeleton significance called medial persistence. They form medial persistence based on the observation that local descriptors are easy to cave into boundary noise while global descriptors are hard to compute. They measure the medial persistence by looking at the elapsed time during the removal process and make the observation that a discrete voxel persists in an isolated form. For example, an edge could survive if it is not adjacent to any face meaning that it is isolated. Also, the method can determine the medial persistence of the voxels in a single thinning pass. In 2D, they observe that, over the iterations, thin and long edges and faces centered at wide parts take longer to disrupt. They apply the same reasoning to 3D shapes and see that it works on 3D shapes, too.

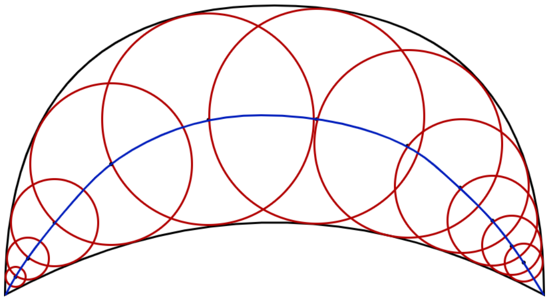


Figure 2.2: Example of a ball inscription method.

When we compare our method with the thinning methods, we see that our method does not involve the removal of the simple points, namely, thinning. Moreover, in our method, the skeleton points and their connections are found at the same time whereas the thinning methods require post-processing for grouping skeletal points and forming a skeletal graph by connecting them. Also, as a feature common to our method and the thinning methods, we see that both involve pruning and re-connection steps.

2.2 Geometric Methods

Geometric methods, contrary to other methods that we mention, work on meshes and scatter points to extract the skeleton. We can divide the geometric methods under two sub-categories: the popular medial-axis/surface-based methods, and Reeb graph-based methods.

Considering medial-axis/surface-based methods, we see that some methods use the Voronoi diagram while others use the ball inscription method that works by inserting a ball into the shape and shifting it along the skeletal curve, while also changing the size of the ball as shown in Figure 2.2. The earlier versions of this method produce good results if boundary noise is not a problem. Otherwise, the ball inscription methods contact every surface point, which results in including every small noise in the skeleton. Intuitively, shapes with more details create a performance drop. Since this idea is computation heavy, the authors in [16] utilize the GPU, make ball calculations in parallel, and claim that they speed up over two times faster than earlier works with lower memory usage. These methods require curve thinning algorithms since their usual output is medial surface.

The authors in [7] leverage Voronoi diagrams where they first re-mesh the shape via edge-split and edge-collapse to obtain simplified mesh connections and then create Voronoi poles and skeletonize the shape with mean curvature flow. After initial medial skeleton is calculated, iterative mesh contraction begins and with that smaller intermediate medial skeletons are created. Mesh contraction, if continued until the end, would result in a point but, in the method, they counterbalance contraction force with attraction forces that are directed towards the vertices from previous time steps. The balance is achieved with several parameters, which are mentioned briefly in § 5.3. With the last iteration of mesh contraction, a final edge collapse is applied to extract curve skeleton.

A strategy we do not mention until now is the divide-and-conquer approach. In [17], the authors first segment the shape based on the k-means fuzzy clustering algorithm where the geodesic distance between vertex pairs is used as the dissimilarity measure. After segmentation, every vertex is assigned to a group. Then, the cut planes and the center points are calculated with a rotational symmetry axis based method in order to form the curve skeleton branch inside the group. Since the groups are disconnected, a Laplacian smoothing step is implemented to connect the endpoint of the branches where the connections may form thick joints. Thinning is performed for the joints and the centers of the joints are shifted so that a re-centering is made. The method ends with sub-sampling from the skeleton and connecting sample points with short curved segments.

A Reeb graph uses Morse theory to encode the topology of the original shape as a 1D structure. The topology is acquired by “following the evolution of the level sets of a real-valued function", which is defined on the respective shape and made explicit such that the curve skeletons are formalized without a basis on symmetry [11, 10]. An example method [18] for the discrete Reeb graph, which means that the graph can be extracted from a discrete surface with gaps and noise, slices shape from bottom to top horizontally and works on the 2D plane, created by the slicing step, considers each curve as a node in the graph and connects nodes if the curves in the adjacent slices are connected. Formed node set, is the method’s skeleton.

Comparing our method with the geometric methods, input is the most apparent difference since geometric methods works on meshes and scatter points while our method works on voxels. Geometric methods may have re-connection step, as is ours, which could be in different forms but the one mentioned in this chapter is done via Laplacian smoothing also geometric methods could have pruning steps.

2.3 Distance or General Field Methods

We categorize the related works based on the first step they apply to skeletonize the given shape. Under this category, we have methods that start with the field calculation and continue with thinning, pruning, or re-connection. Our method is in this category because it starts with the calculation of v_k field and then continues with pruning and re-connection steps as we explain in Chapter 3 and 4.

Some of the methods in this category skeletonize the shapes using their distance transform that assigns each shape point its distance to the nearest boundary [10]. Generally, the methods that use the distance transform create a large pool of skeleton points that need to be pruned. However, pruning may eliminate valuable connections so that an extra re-connection step is needed [11].

One method [19] that involves some of the mentioned steps takes voxel images or volumes as the input and outputs surface and curve skeletons. The method leverages <3,4,5> weighted distance transform called chamfer technique [20] and has the adjacent element order <f,e,v> as weights, each element representing the number of faces, edges and vertices along the way to the boundary. After the distance transform is computed, voxels are grouped based on the values they have and whether they are simple points or not, are found out. The method continues with eliminating voxels and creates a subset that potentially has curve skeleton points in it. While the method preserves complete topology over surface skeletons, it can only preserve partial topology over curve skeletons. It then applies to thinning in order to calculate the curve skeleton.

Another method [4] extracts curve skeletons as a subset of surface skeletons which almost guarantees that the curve skeleton will be centered because points in the curve

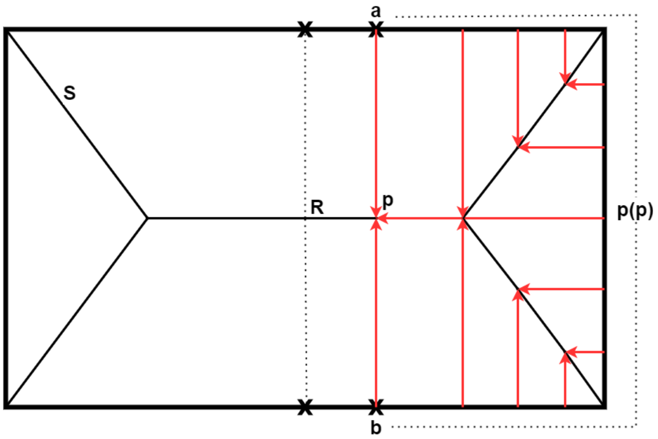


Figure 2.3: Origins of the trajectories form the collapse of p [4]

skeleton are matched with surface skeletons based on the distance. They introduce the collapse measure which is robust to noise, produces connected skeletons, and describes shapes at different levels. Collapse measure means that mass in the object boundary is advacted onto and then along the skeleton, and the total advacted mass through the point is it's collapse measure. In Figure 2.3 we show the point p's collapse measure calculation concept on a 2D rectangle where R is root, S is skeleton, $\delta\Omega$ is the boundary and a and b is our equi-distant points. They approximate this measure without simulation but with direct computation. Method continues with thresholding where some points are eliminated and the remaining points are the simplified skeleton result of the method.

In [21], the method [22, 23] extracting local symmetry axis of 2D shapes is extended to shapes in arbitrary dimensions including 3D shapes. The method involves first computing a smooth field inside the shape domain and then determining skeleton points based on the properties of the level curves/surfaces of the field passing through the points. The method is able to extract both surface and curve skeleton of 3D shapes.

When we compare our method with the last method we mention, we see that both of the methods determine skeleton points based on the level curves/surfaces of a smooth field. For the other methods, we see that after the fields are calculated, the methods go through pruning and re-connection steps but, to the best of our knowledge, our method is the only one exploring the shape domain by carving it from the inside out.

CHAPTER 3

A FAMILY OF SMOOTH DISTANCE TRANSFORMS

In this chapter, first, we present what we call as v_k field, which denotes a family of smooth distance transforms defined on incrementally carved forms of the shape domain. Then, we discuss its connection to the part-coding distance transform of Tarı [1, 2].

3.1 v_k Field

Let shape be an open set Ω with boundary $\partial\Omega$. v_k for $k = 0, 1, 2, 3, \ldots, k_{max}$ is an ordered family of fields defined on incrementally carved shape domains Ω_k where $\Omega_0 = \Omega$ and $\Omega_k \subset \Omega_{k-1}$ for $k = 1, 2, 3, \ldots, k_{max}$.

Each $v_k : \Omega_k \to \mathbb{R}$ is a normalized version of a smooth distance field obtained by solving the following Partial Differential Equation (PDE) on the respective shape domain Ω_k

$$\Delta v_k - \frac{1}{|\Omega_k|^2}\, v_k = -1 \text{ subject to } v_k\,|_{\partial\Omega_k} = 0 \tag{3.1}$$

where Δ is the Laplace operator, $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ in 2D and $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ in 3D, and $|\Omega_k|$ denotes the number of shape points in Ω_k. We normalize each v_k field by dividing its values to its maximum value.

Each v_k is minimal on the boundary $\partial\Omega_k$ and increases towards the center of Ω_k. This is due to that the solution of (3.1) is a scaled version of the field minimizing the following energy:

$$\arg\min_{v_k} \int\limits_{\Omega_k} \left(|\nabla v_k|^2 + a^2\,(1 - v_k)^2 \right) \mathrm{d}\omega_k \text{ subject to } v_k\Big|_{\partial\Omega_k} = 0. \tag{3.2}$$

We see that each v_k field is smooth due to the first term in the energy, close to 1 inside Ω_k due to the second term in the energy, and 0 on $\partial\Omega_k$ due to the boundary condition. Thus, each v_k is a smoothed form of the characteristic function which takes 0 on $\partial\Omega_k$ and 1 inside Ω_k where the smoothing degree is inversely proportional to the parameter a^2. We keep the smoothing degree of each v_k high by setting a^2 to a very small value $(1/|\Omega_k|^2)$ which depends on the size of the corresponding shape domain Ω_k.

Each v_k suggests a binary partitioning of the respective shape domain Ω_k via watershed boundary W_k formed between enclosing boundaries of Ω_k, namely, W_{k-1} and $\partial\Omega$, where each Ω_k is constructed by subtracting enclosure of W_{k-1} from Ω_{k-1} that is $\Omega_k = \Omega_{k-1} \setminus \text{enclosure}(W_{k-1})$.

In Figure 3.1, we illustrate v_k field construction via an example shape. Ω_0 is equal to Ω and, hence, v_0 is defined on the whole shape domain. W_0 is determined as a subset of points where local maxima of v_0 are attained. Regarding the example in Figure 3.1, W_0 is defined as the global maximum point of v_0 (see red point in Figure 3.1 (a)).

Ω_1 is constructed by subtracting W_0 from Ω_0 which implies that Ω_1 corresponds to the region bounded by W_0 and $\partial\Omega$ that is $\partial\Omega_1 = W_0 \cup \partial\Omega$. v_1 is minimal on $\partial\Omega_1$ and increases towards center of Ω_1. Accordingly, W_1 is extracted from v_1 as the watershed boundary between W_0 and $\partial\Omega$ (see red contour in Figure 3.1 (b)).

The same procedure is followed for each of the remaining levels $k = 2, 3, \ldots, k_{max}$. Ω_k is constructed by subtracting enclosure of W_{k-1} from Ω_{k-1}. $\partial\Omega_k = W_{k-1} \cup \partial\Omega$ since $\partial\Omega_{k-1} = W_{k-2} \cup \partial\Omega$ and W_{k-1} lies between W_{k-2} and $\partial\Omega$. v_k is minimal on $\partial\Omega_k$ and increases towards center of Ω_k. Accordingly, W_k is extracted from v_k as the watershed boundary between W_{k-1} and $\partial\Omega$. In Figure 3.1 (a)-(f), we visualize v_k field at sample levels $k = 0, 1, 2, 3, 4, 5$ where v_k field of this shape has a total of 27 levels ($k_{max} = 27$).

The watershed boundary, and hence carving, expands from level to level. The maximum level is reached when the watershed boundary no longer expands that is $W_{k_{max}+1} = W_{k_{max}}$. In Figure 3.1 (g), we present carved regions overlaid onto each other showing expansion of the watershed boundary throughout all of 27 levels for the example shape.

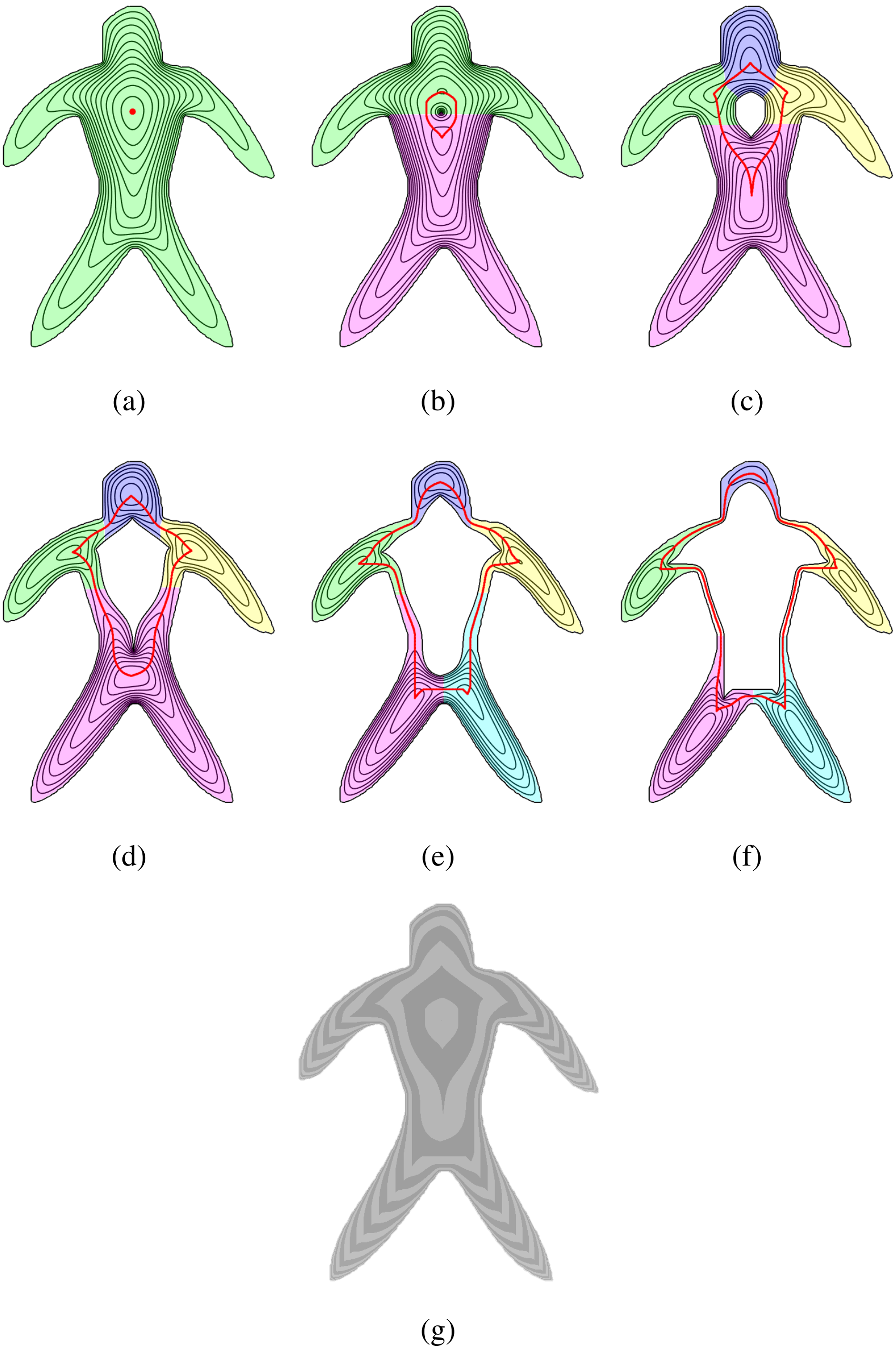
Notice that level curves of each v_k divide the corresponding shape domain Ω_k into meaningful regions each shown using a different color in Figure 3.1 (a)-(f).

3.1.1 Extraction of Watershed Boundary

Watershed boundary can be described as follows. Consider a topographic relief which is flooded by placing the sources at the local minima. The level of the flood is uniform over the relief and increases with uniform speed. The moment that the floods filling two distinct catchment basins start to merge, a dam is built in order to prevent mixing of the floods. The union of all dams constitutes the watershed boundary.

In Figure 3.2, we illustrate v_2 shown in Figure 3.1 (c) as a topographic relief. Notice that there are two local minima, one in the background outside the shape domain and one in the central carved region. Red barrier in Figure 3.2 (bottom) illustrates the watershed boundary between these two local minima.

In order to obtain the watershed boundary, we utilize MATLAB's watershed function in which the method presented in [24] is implemented.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 3.1: (a)-(f) v_k field at sample levels $k = 0, 1, 2, 3, 4, 5$ where $k_{max} = 27$. W_k is denoted via red contour. (g) Carved regions overlaid onto each other showing expansion of the watershed boundary throughout all of 27 levels.

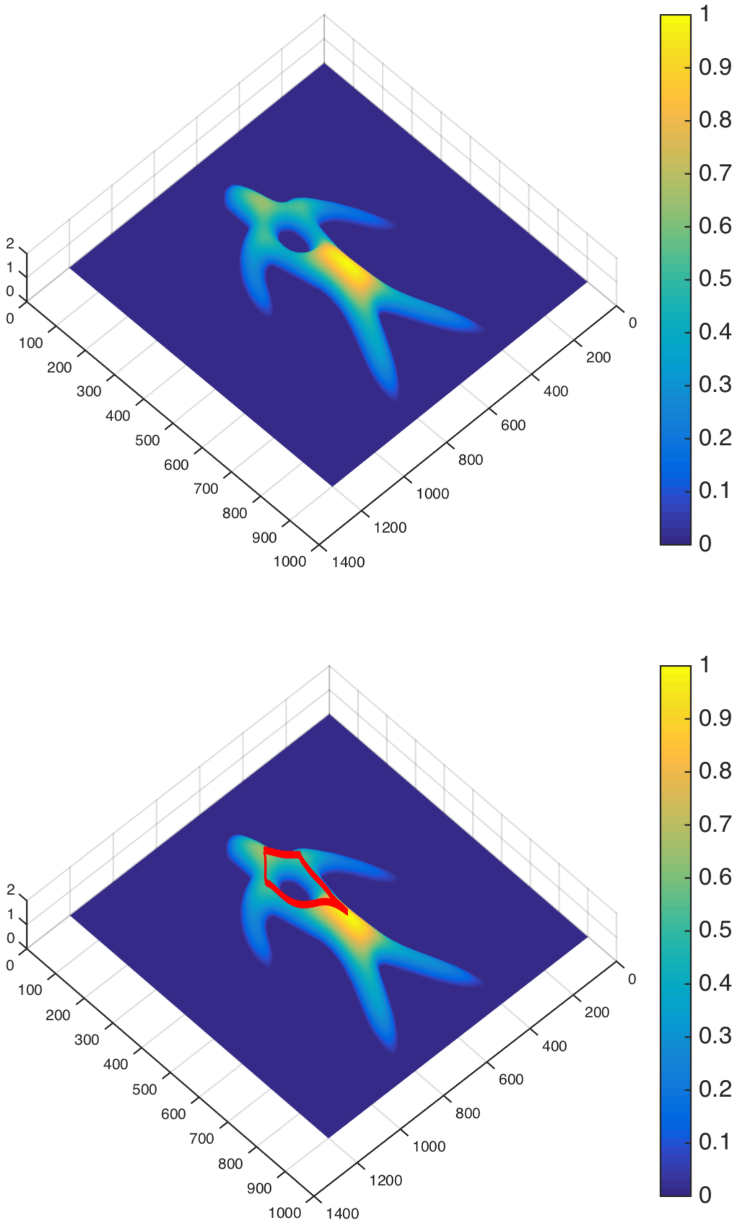


Figure 3.2: v_2 in Figure 3.1 (c) shown as a topographic relief where W_2 is illustrated via red barrier at the bottom.

3.1.2 Limitation: Shape Resolution

As illustrated in Figure 3.3, low shape resolution prevents expansion of the watershed boundary through the entire shape domain. This is due to that, at low resolutions, discretization affects both computation of v_k field and approximation of the watershed boundary significantly. We see that the entire shape domain is covered when we upscale the shape as in Figure 3.3 (right).

In order to overcome this limitation, we could upscale the shape until the corresponding v_k field covers the whole shape domain. This is one of the future works.

3.1.3 Applicability to 2D and 3D Shapes

v_k field construction involves two main steps that are computation of the smooth distance field and extraction of the watershed boundary, both of which are applicable to 2D and 3D shapes.

3.1.4 Different W_0 Configurations

It is possible to start carving from more than one central point by determining W_0 to consist of more than one local maximum point of v_0. For example, considering dumbbell-like shape in Figure 3.4 (b), we start carving from two central points since its body contains two centers. Also, considering a composite shape such as frog shape in Figure 3.4 (c)-(d), points in the center of ligature zones where multiple shape parts are connected to each other can be added to W_0 (see Figure 3.4 (d)).

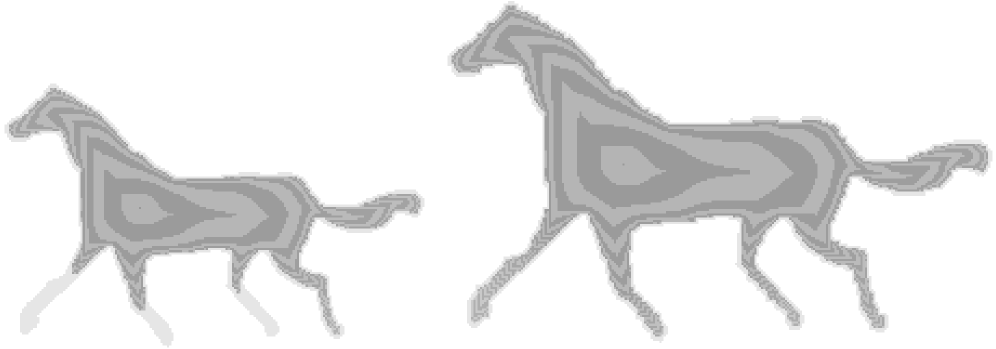


Figure 3.3: Low shape resolution prevents expansion of the watershed boundary through the entire shape domain. The right shape is obtained by upscaling the left shape 1.5 times.

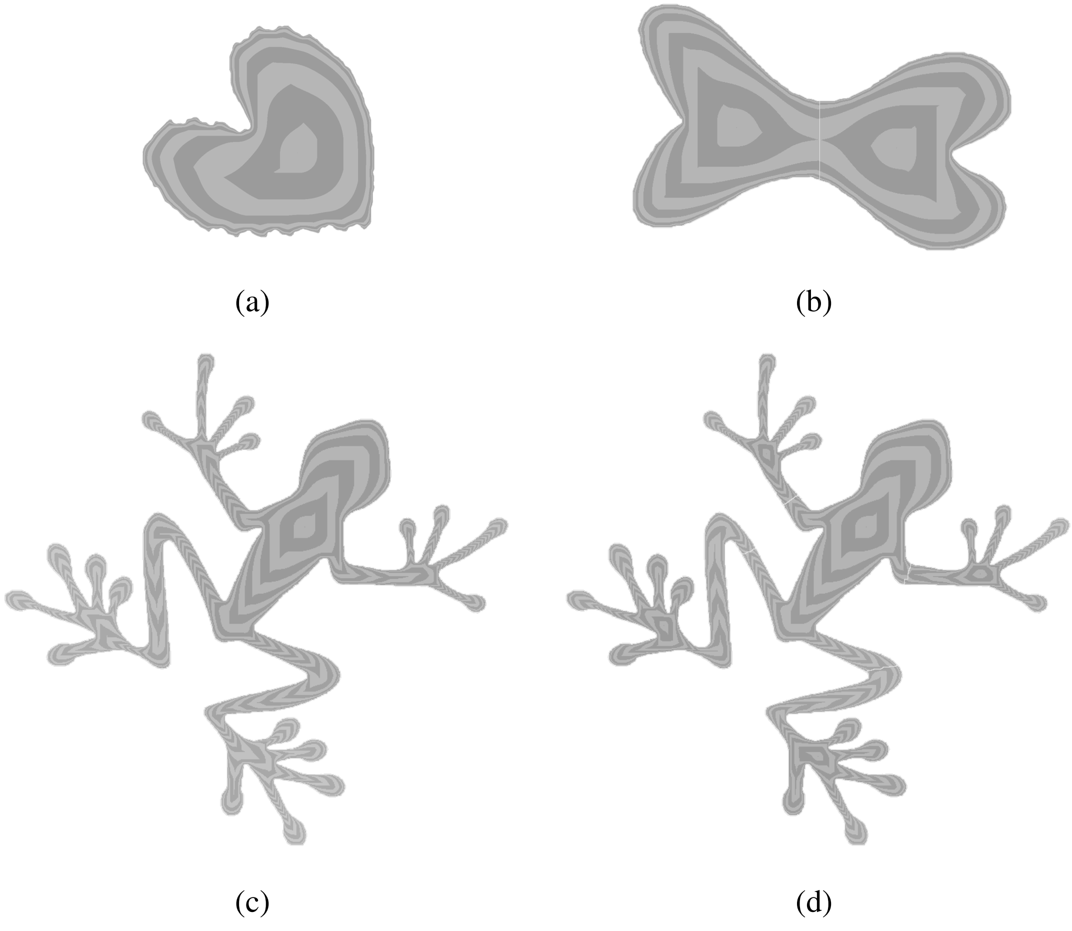


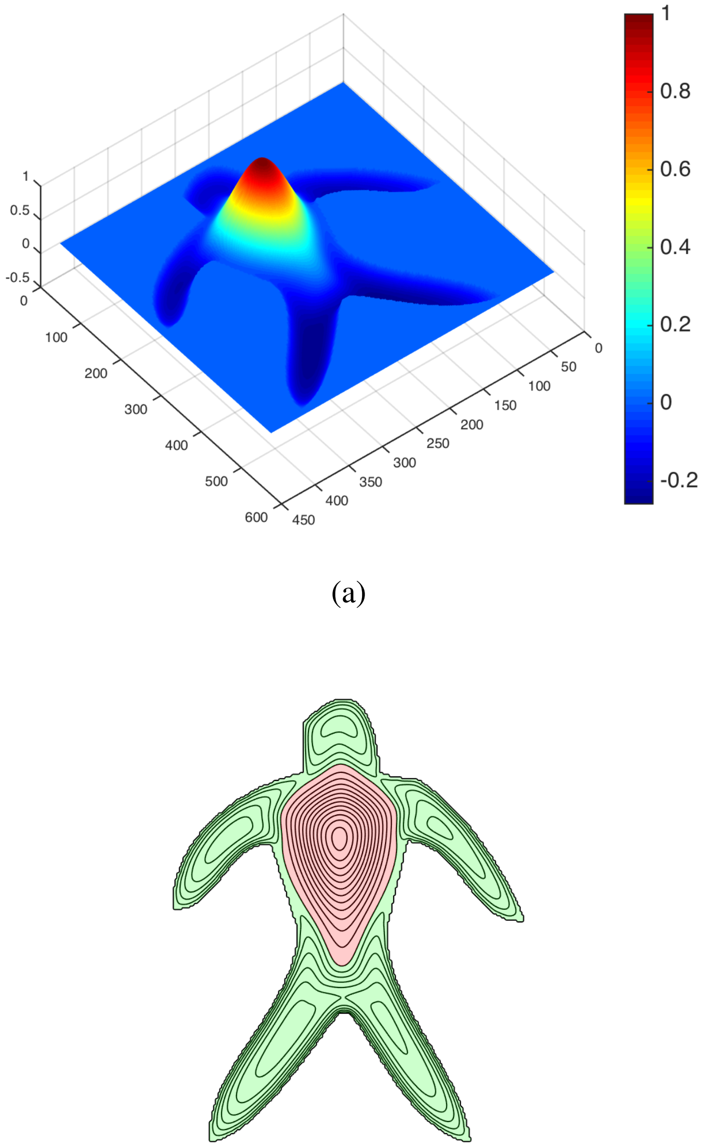
Figure 3.4: Incremental carving of the shape domains where the number of centers from which the carving is initiated is (a) one, (b) two, (c) one, and (d) five.

3.2 The Connection to the Distance Transform of Tarı [1, 2]

In 2009, Tari proposed a part-coding distance field [1, 2] which is called the fluctuating distance field since it is a smooth distance field composed of both positive and negative values (see Figure 3.5 (a)). The fluctuating distance field is computed by solving a dense linear system of equations modeling local and global interactions between the shape points. Accordingly, the field exhibits highly nonlocal behavior (see Figure 3.5 (b)). Inner zero-level curve divides the shape into central (positive) region corresponding to a coarse form of the shape body and peripheral (negative) region including limbs, protrusions, boundary texture and noise. Level curves of the field inside negative and positive regions provide further partitioning of the shape. In Figure 3.5 (b), we see that level curves of the field encode meaningful shape parts corresponding to body, head, and each of legs and arms.

Considering v_k field presented in this chapter, central/peripheral separation of the shape domain is achieved by its incremental carving. By solving the PDE 3.1 on incrementally carved forms of the shape domain, we obtain a shape flow from the inside out. Carving starts from inner point(s) and proceeds gradually until the whole shape domain is covered. Carving at each level is guided by the smooth distance field obtained via solving the PDE 3.1 on the respective shape domain. Despite its local computation, level curves of the field exhibit highly nonlocal behavior by providing partitioning of the carved shape domain. Carving expands towards the field extrema which roughly mark part centers.

Observe that the shape model obtained via v_4 shown in Figure 3.1 (e) is similar to the one obtained via the fluctuating distance field shown in Figure 3.5 (b) as both models provide similar partitioning of the shape domain.



(a)

(b)

Figure 3.5: Fluctuating distance field [1, 2]. (a) 3D topographic view. (b) Level curves where positive/negative regions are shown with red/green colors, respectively.

CHAPTER 4

CURVE SKELETON OF 2D AND 3D SHAPES

In this chapter, we present how curve skeleton of 2D and 3D shapes is constructed using v_k field.

4.1 Constructing Curve Skeleton of 2D and 3D Shapes

Regardless of the dimension of the shape, whether it is 2D or 3D, our curve skeleton model is computed by following the same steps for both 2D and 3D shapes. Below, for simplicity, we describe the construction of our curve skeleton by illustrating it on example 2D shapes.

4.1.1 Determining Skeleton Points

As shown in Figure 3.1 (also, in Figure 4.2), the enclosure of the watershed boundary at a particular level grows to form the enclosure of the watershed boundary at the following level. The watershed boundary at each level is expected to pass through the local maxima of the corresponding v_k field which roughly mark part centers. This brings the idea that we can define skeleton points at each level as local maxima of v_k field (see Figure 4.1 (a)). However, the local maximum points do not always mark the shape parts through which the watershed boundary passes. Therefore, the skeleton points at each level k are defined as the local maximum points of v_k field whose associated parts intersect with the watershed boundary W_k. In order to illustrate this situation, in Figure 4.2, we visualize v_k field of an example 2D shape at sample levels $k = 0, 1, 2, 3, 4$. Each v_k is shown via its level curves. The red contour denotes the watershed boundary W_k. The black dots mark the local maximum points of v_k. The parts associated with the local maximum points are shown in different colors so that they can be differentiated from each other. We see that, in v_1, although there are five local maximum points, only two of them are defined as skeleton points since the parts associated with the remaining three do not intersect with the watershed boundary. Similarly, in v_2/v_3, although there are five/seven local maximum points, only two/four of them are defined as skeleton points, respectively, since the parts associated with the remaining ones do not intersect with the watershed boundary.

Before checking whether the local maximum points mark the shape parts intersecting the watershed boundary, there is another step for determining the skeleton points reliably. It is possible that v_k field has local maximum points that are unconnected but very close to each other marking the same shape part. This situation can be observed in Figure 4.2 (e) as there are two local maximum points marking the back leg of the elephant. To overcome this issue, we group the local maximum points (and their associated parts) so that two points are in the same group if the distance between them is smaller than half of the minimum of their distances to the nearest boundary. After grouping, the local maximum point associated with each group is chosen as the one whose distance to the nearest boundary is the largest.

As exemplified in Figure 4.1 (a), at some levels, the watershed boundary may fail to pass through the local maxima precisely. Thus, geodesic projection of the local maxima points onto the watershed boundary may be necessary (see Figure 4.1 (b)). By geodesic projection, we mean finding the nearest point along the watershed boundary where the distance between the local maxima point and each point along the watershed boundary is computed as the length of the shortest path between them through the shape domain. Specifically, for implementing geodesic projection, we use bwdistgeodesic function of Matlab.

4.1.2 Connecting Relevant Skeleton Points

After determining skeleton points, we need to attach relevant ones to obtain a skeleton model of the shape. We connect skeleton points at consecutive levels if their associated parts, which are obtained as watershed zones of the corresponding v_k field, intersect with each other (see Figure 4.1 (c)). Given a skeleton point s at a level k, we determine its parent as the skeleton point at the inner level $k-1$ as follows. First, we find all the skeleton points at level $k-1$ whose associated parts intersect with the associated part of s. If there is more than one such point, we choose the one closest to the projection of the skeleton point s onto the watershed boundary W_{k-1}. Notice that links from skeleton segments modeling boundary texture and noise to skeleton segments modeling limbs or protrusions may arise.

4.1.3 Pruning

The aim of the pruning step is to remove the connections between the skeleton segments modeling boundary texture/noise and the skeleton graph modeling the shape body, limbs and protrusions. We apply pruning by examining the link from each skeleton point to its parent as follows. We project the skeleton point onto the watershed boundary of its parent, and we remove the link if Euclidean distance from the projection point to the parent point is greater than Euclidean distance from the skeleton point to the nearest point of the shape boundary $\partial\Omega$ (see Figure 4.1 (d)).

4.1.4 Reconnection

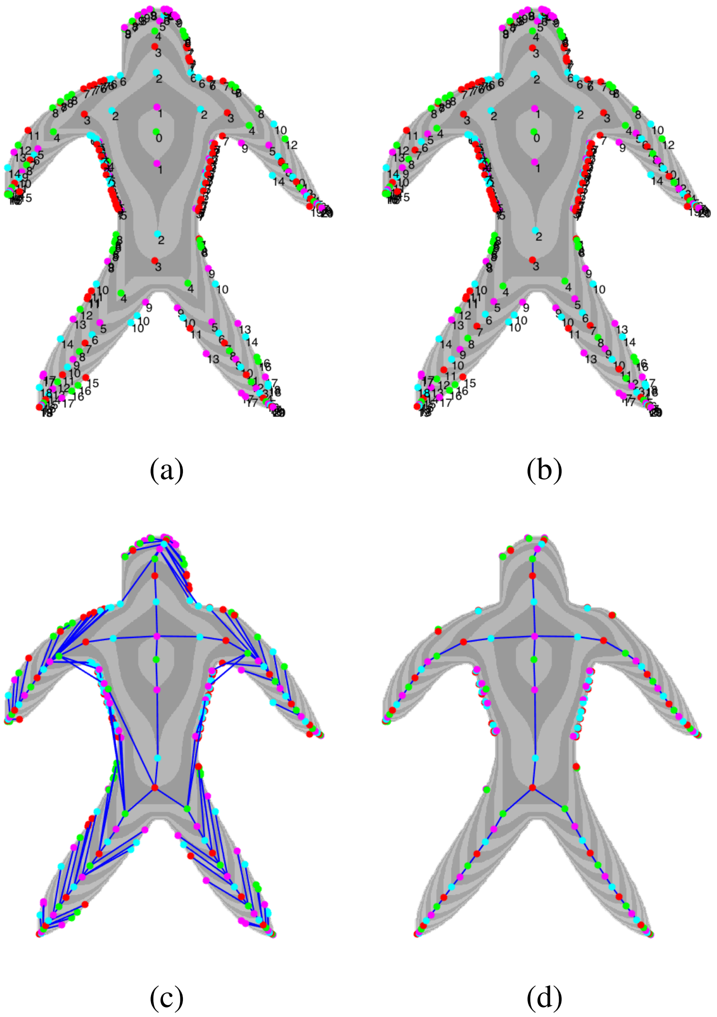
After the first pruning step we see that limb/protrusion can sometimes be disconnected from the main skeleton (see Figure 4.3 (b)). To eliminate disconnections to main skeleton, we come up with the following steps to connect these missing links.

- Find out if our pruned skeleton has disconnected skeleton groups based on the calculations we describe below.
- In case there are disconnected skeleton groups, decide whether they are disconnected limb/protrusions or boundary noise.
- If skeleton groups should be connected, find missing connections from fully connected skeleton, which is the skeleton created by skipping the pruning step of our method, then apply those connections to our pruned skeleton.

The decision to connect the disconnected skeleton groups are made based on the following criterias. For every skeleton group, we calculate the total area they represent. For this representation we take every skeleton point in the skeleton group and we calculate maximally inscribed circle/ball in 2D/3D at that point meaning that the largest circle/ball we can draw without crossing the boundary of the shape, which in summary means that the radius of the circle/ball at a skeleton point is equal to its smallest distance to the boundary as shown Figure 4.3(c). We add up these circular areas and get the total area for the skeleton group. The group with the largest total area is our main skeleton, our longest link. We consider that the skeleton groups with area smaller than a given ratio κ of the main skeleton correspond to boundary texture/noise and the rest of the skeleton groups corresponds to limbs/protrusions which should to be connected to the main skeleton. After decision to connect these missing links are made, we transfer our linking information to an adjacency list graph structure. By using this graph we recursively find our missing connections and add them to the pruned skeleton see Figure 4.3 (d).

For completeness, in Figure 4.4, we present our skeleton model after each step for the 2D shape in Figure 4.2.

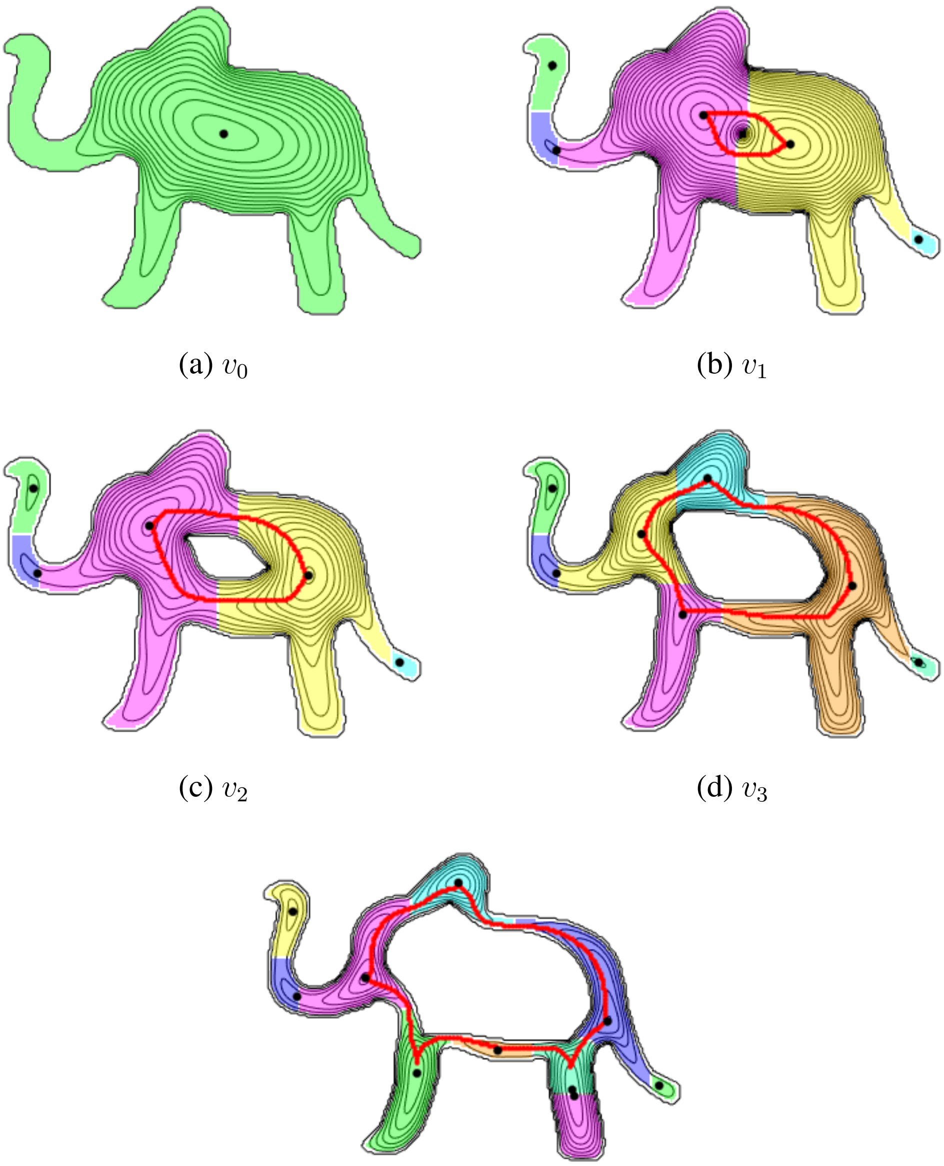
In Figure 4.5, we show curve skeleton for an example 3D shape.



(a) (b)

(c) (d)

Figure 4.1: (a) Local maxima of the family of v_k fields where each point is shown together with its level. (b) Skeleton points obtained via geodesic projection of the local maxima points onto the watershed boundary. (c) Skeleton points at consecutive levels are connected if their associated parts (or watershed zones) intersect with each other. (d) Pruning of noisy links from skeleton segments modeling boundary texture and noise to skeleton segments modeling limbs and protrusions. For this shape, reconnection step yields the same result as pruning.



(a) v_0

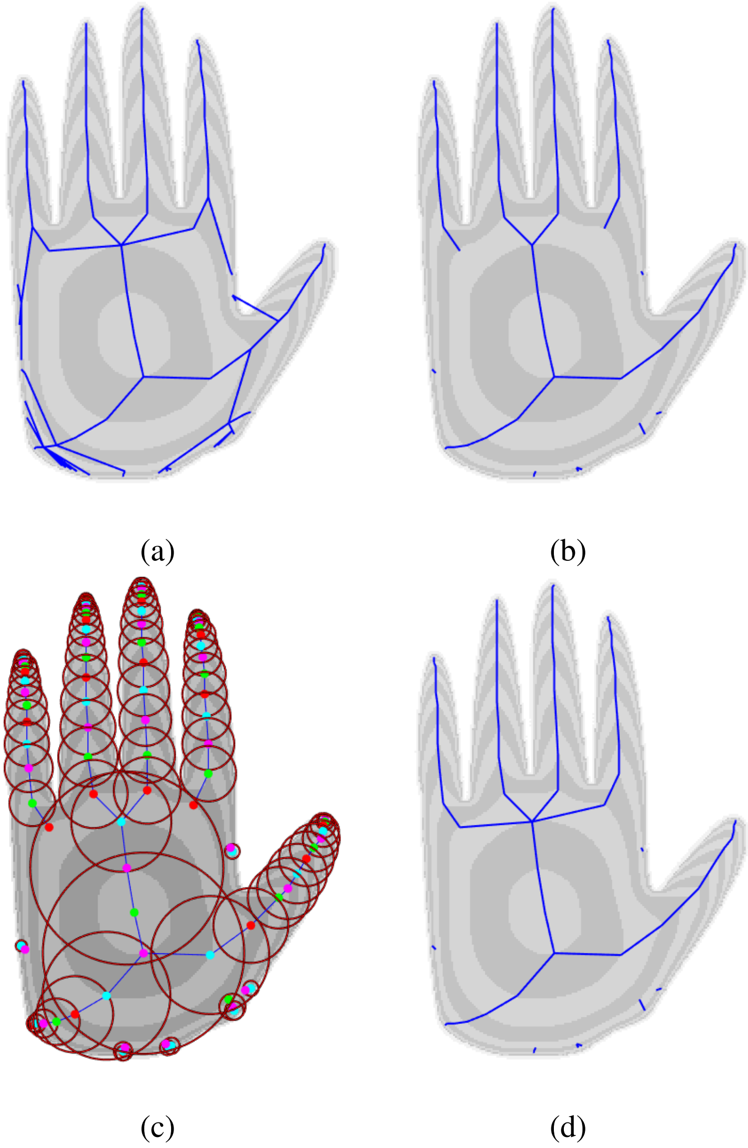
(b) v_1

(c) v_2

(d) v_3

(e) v_4

Figure 4.2: (a)-(e) For a 2D elephant shape, level curves of v_k field at sample levels $k = 0, 1, 2, 3, 4$ where $k_{max} = 31$. In each visualization, the red contour denotes the watershed boundary W_k, the black dots mark the local maximum points, and the parts associated with each local maximum are shown in different colors.



(a) (b)

(c) (d)

Figure 4.3: For a 2D hand shape, our skeleton model after each step (a) fully connected result (b) pruning result (c) skeleton points and maximally inscribed circles associated with them (d) reconnection result where $\kappa = 1/50$.

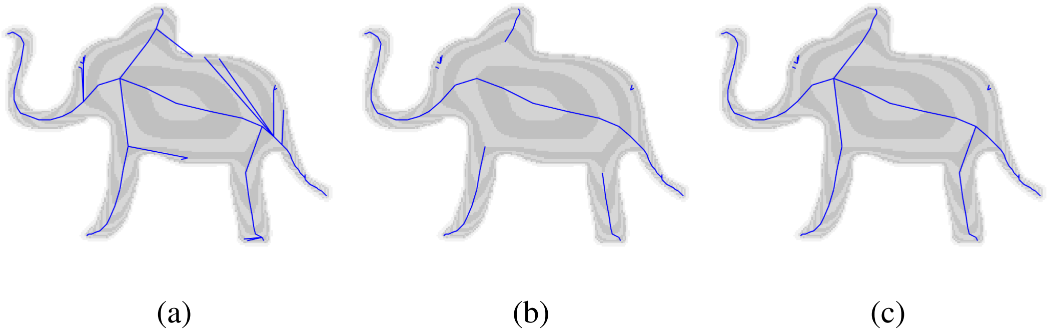


Figure 4.4: For the shape in Figure 4.2, our skeleton model after each step (a) fully connected result (b) pruning result (c) reconnection result where $\kappa = 1/50$.

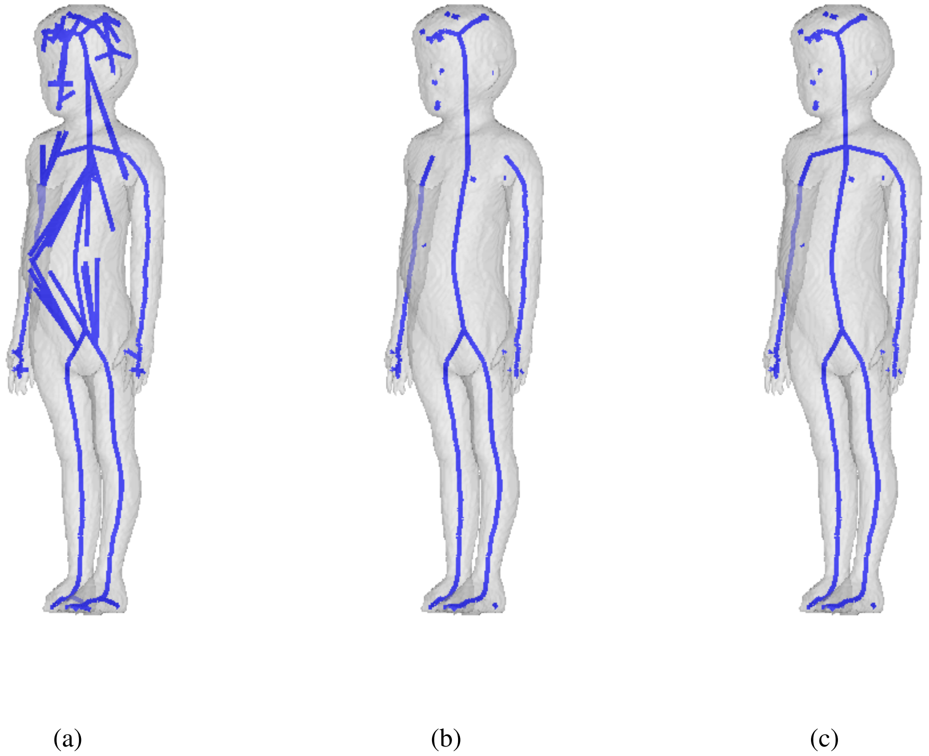


Figure 4.5: For a 3D human shape from the dataset [5], our skeleton model after each step (a) fully connected shape (b) after pruning step (c) after reconnection step where $\kappa = 1/5$.

CHAPTER 5

RESULTS AND DISCUSSION

In §5.1 and §5.3, we present and discuss our curve skeleton model computed for several 2D and 3D shapes. For all of the results, we take $\kappa = 1/50$ unless otherwise stated. In §5.2, we present limitations of our method.

5.1 2D Shapes

We present our results for 2D shapes under three categories, namely, rectangular shapes, annulus, and shapes from 56Aslan dataset [3].

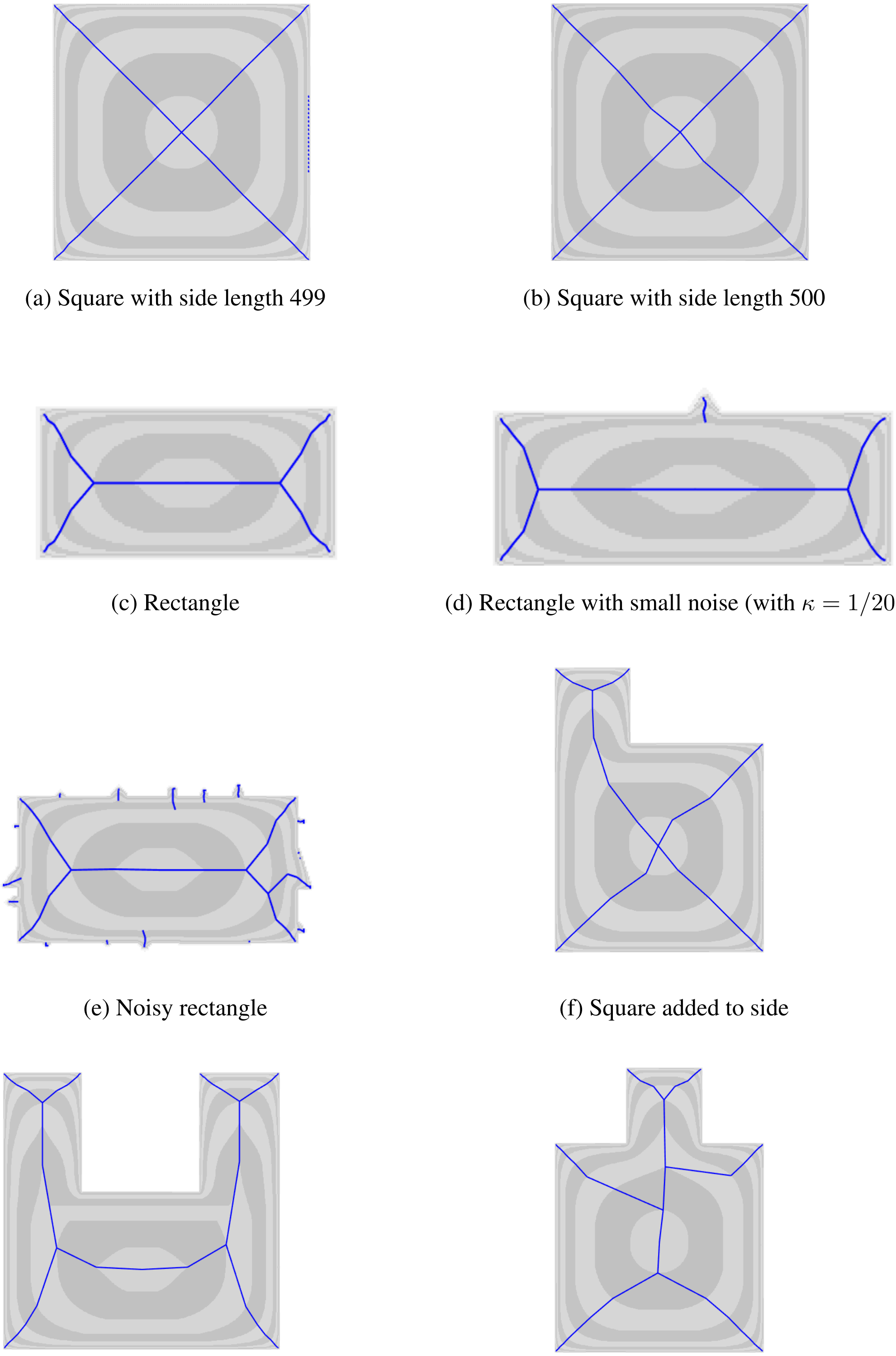
5.1.1 Rectangular Shapes

Rectangular shapes are the shapes that we experiment with to understand the nature of our skeleton model. This category includes eight rectangular shapes: one square with an odd side length, one square with an even side length, a rectangle, a rectangle with small noise, a noisy rectangle, a smaller square added to the top-left of a larger square, a smaller square removed from the top-middle of a larger square, and a smaller square added to the approximately top-middle of a larger square. In Figure 5.1, we present our skeleton model for the shapes in this category. For comparison, in Figure 5.2, we present skeletons of the shapes obtained using bwmorph function of Matlab that uses Medial Axis Transform (MAT). We choose MAT for comparison since it is an accurate, easy-to-explain, and well-known method. Also, its available implementation in Matlab makes the shown results reproducible.

We observe that our skeleton is similar to the MAT result for the squares and the rectangle (see Figure 5.1 and Figure 5.2 (a)-(c)).

Considering the rectangle with small noise and the noisy rectangle, we see that our skeleton is not affected by the boundary noise (see Figure 5.1 (d) and (e)). Our skeleton is composed of a connected skeleton modeling the main shape and a number of small skeleton segments modeling the boundary noise. However, considering the skeletons obtained using MAT (as shown in Figure 5.2 (d) and (e)), we see that the skeletal branches modeling the boundary noise are connected to the main skeleton and the boundary noise affects the main skeleton making it shift towards the noisy boundary. This behaviour of MAT makes pruning of the skeleton challenging and, even after pruning, since the main skeleton is shifted, the resulting skeleton has left with deformations.

For the remaining three shapes, our skeleton results are similar to MAT results (see Figure 5.1 and Figure 5.2) (f)-(h)). Adding a small square to a larger one introduces a new arm so that the entire shape is covered as shown in Figure 5.1 (f). Removing a square from top-middle part of a larger square results in a symmetric skeleton as shown in Figure 5.1 (g). Adding a square to the middle with an offset, shifts connection point of the top right skeleton branch and connects it one level outer than the top left branch as shown in Figure 5.1 (h).



(a) Square with side length 499

(b) Square with side length 500

(c) Rectangle

(d) Rectangle with small noise (with $\kappa = 1/20$)

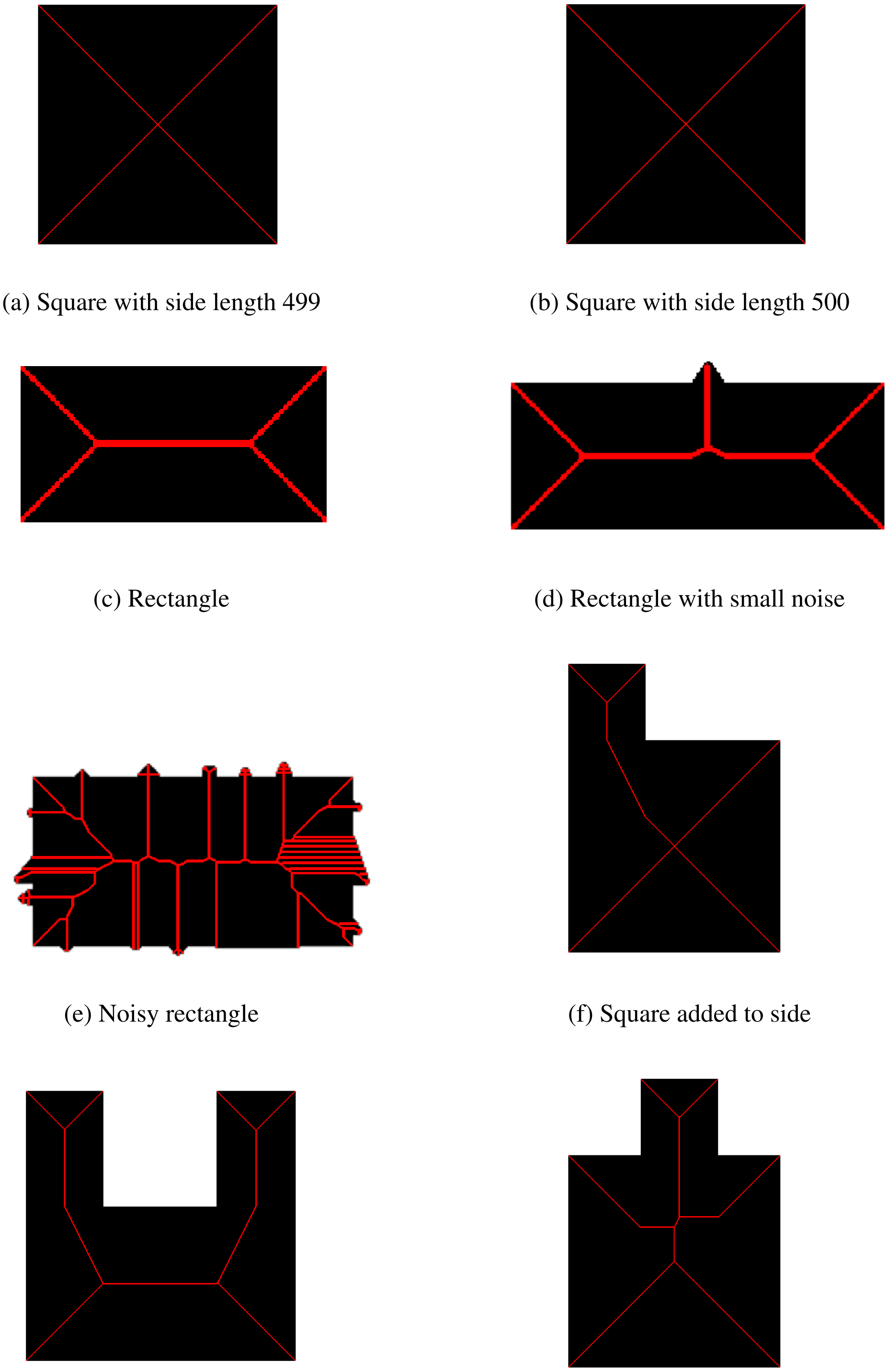
(e) Noisy rectangle

(f) Square added to side

(g) Square removed from middle

(h) Square added to top

Figure 5.1: Our skeleton model for rectangular shapes.



(a) Square with side length 499

(b) Square with side length 500

(c) Rectangle

(d) Rectangle with small noise

(e) Noisy rectangle

(f) Square added to side

(g) Square removed from middle

(h) Square added to top

Figure 5.2: Skeletons obtained using Medial Axis Transform (MAT) of the shapes.

5.1.2 Annulus

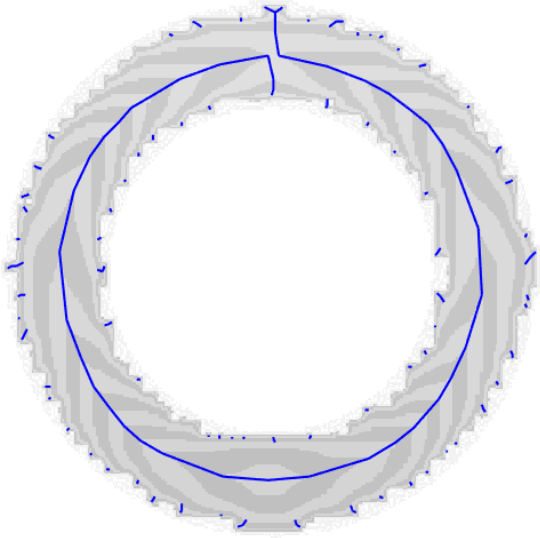


Figure 5.3: Our skeleton model for an example annulus shape.

Annulus is another shape that makes us curious because it is a shape with a hole and its boundary is jagged and noisy due to its discretization in a regular 2D grid. We present our skeleton model for an example annulus shape in Figure 5.3. We see that, starting from the middle of the bottom of the shape, there are two skeleton branches passing through the shape where the left branch is in the clockwise direction and the right one is in the counter-clockwise direction. These two branches meet at a point in the middle of the top of the shape and then they continue to grow in opposite directions so that they reach the shape boundary. We see that our skeleton model is composed of a piece-wise continuous skeletal line capturing the centerline of the main shape and a number of small skeletal segments representing the boundary texture/noise. However, our skeleton model is not topology-preserving. In order to make our skeleton topology-preserving, the skeleton branches meeting at a common point should be connected and this is one of the future works.

5.1.3 Shapes from 56Aslan dataset [3]

56Aslan dataset [3] consists of silhouettes from fourteen shape categories where there are four shapes from each category. The shapes within the same categories have differences with respect to their orientation, scale, articulation, and boundary detail. Our skeleton results for all of the shapes from 56Aslan dataset are presented in Figures 5.4 and 5.5.

For each shape, our skeleton model is composed of a connected skeletal graph modeling the shape body and the limbs/protrusions attached to it. We observe that our skeleton model is robust to boundary texture/noise. Moreover, for the shapes with boundary texture/noise, our method yields a number of disconnected skeletal segments modeling the boundary texture/noise (see Figure 5.6).

Overall, we observe that our skeleton results are consistent for the shapes within the same categories in spite of the differences between the shapes as mentioned above. The difference between the skeletons of the shapes within the same categories is due to the order and the location in which the skeletal branches associated with the limbs/protrusions are connected to each other while reaching the initial skeleton point in the middle of the shape body. In order to better illustrate this situation, in Figure 5.7, we re-present our results for the first two horse shapes in Figure 5.4. For the first horse shape in Figure 5.7 (a), the skeletal branches modeling the front legs are first connected to each other. After growing towards the center of the body, the resulting skeletal tree is connected to the branch modeling the head. However, for the second horse shape in Figure 5.7 (b), the skeletal branches modeling the front legs and the head are connected at the same time. The reason behind this difference is that the front legs of the first horse are occluding each other so that they branch out later compared to the front legs of the second horse.

As mentioned in Chapter 3, when the shape resolution is very low, v_k field and, hence, our skeleton model does not cover thin sections of the shapes such as legs of some horse and cat shapes (see the results for the third horse shape in Figure 5.4 and the second cat shape in Figure 5.5).

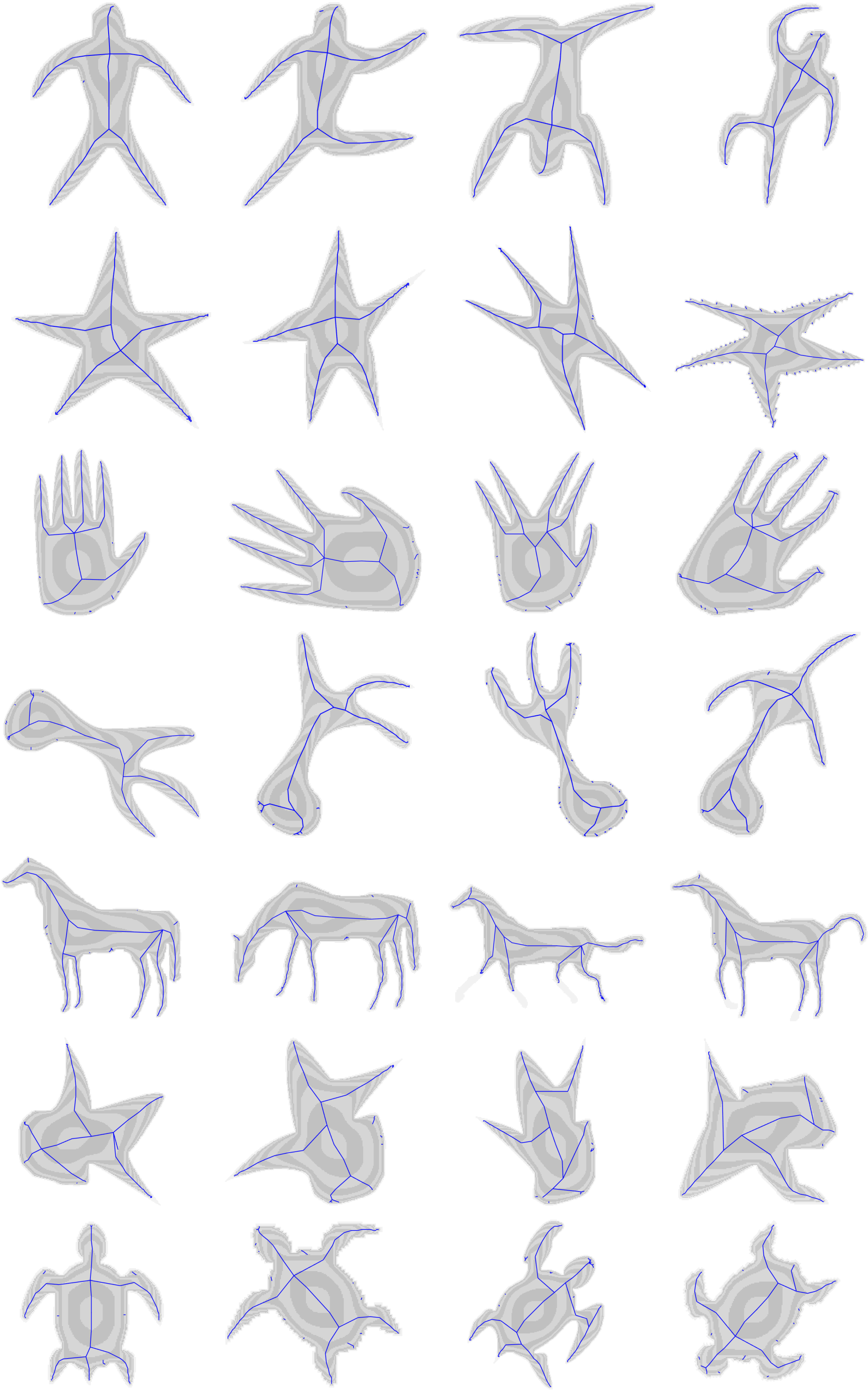


Figure 5.4: Our skeleton model for 2D shapes from the first seven categories of 56Aslan dataset [3].

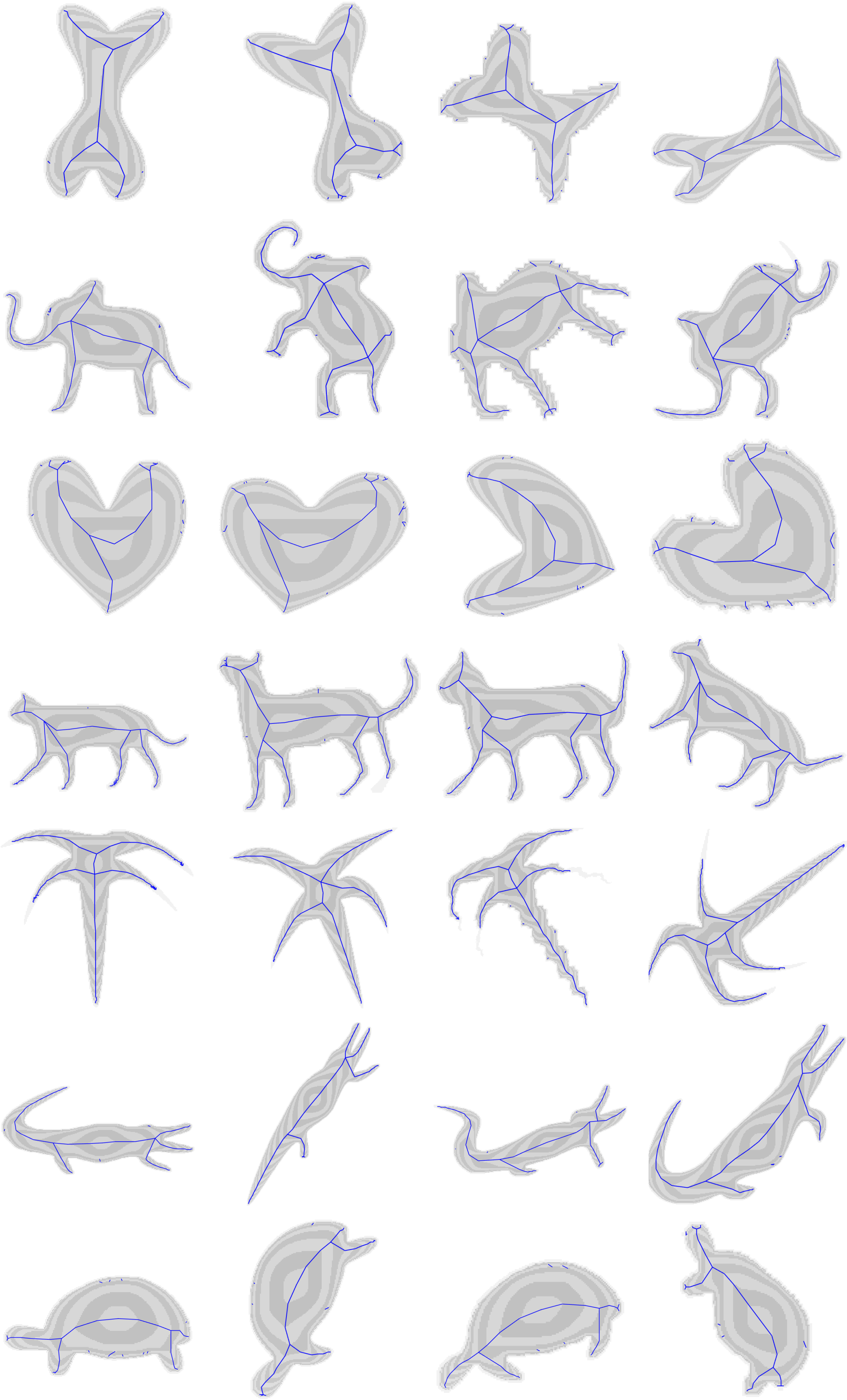


Figure 5.5: Our skeleton model for 2D shapes from the remaining seven categories of 56Aslan dataset [3].

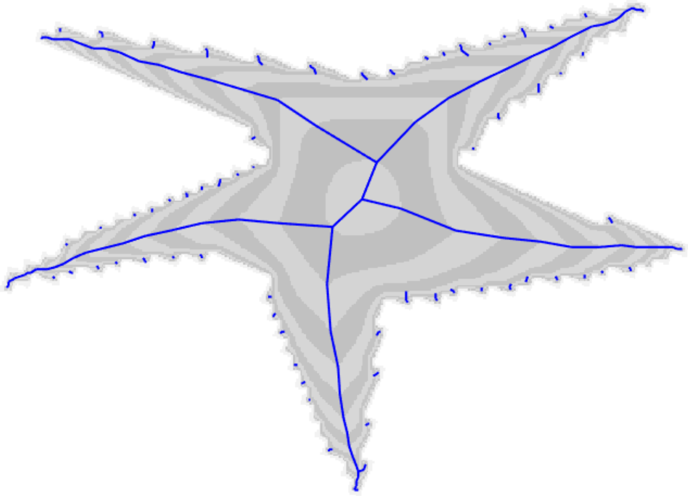


Figure 5.6: In addition to a connected skeletal graph modeling the shape body and limbs/protrusions attached to it, our method yields a number of disconnected skeletal segments modeling the boundary texture/noise.

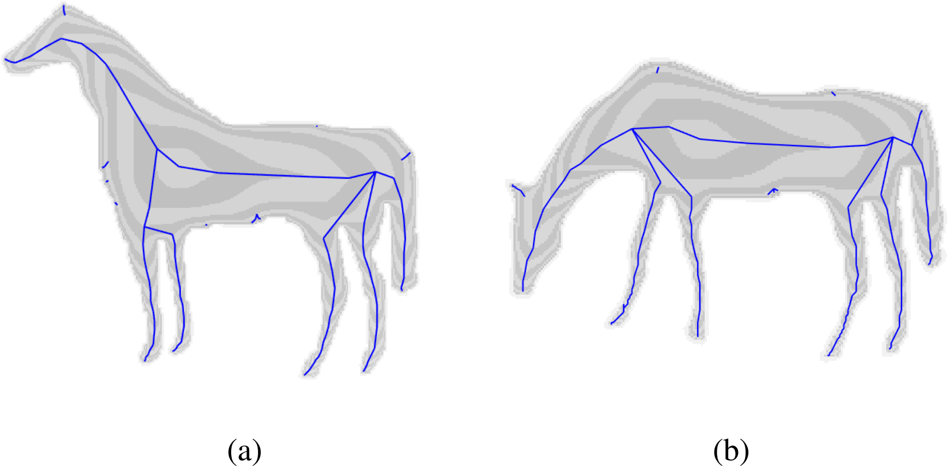


Figure 5.7: For the first two horse shapes in Figure 5.4, skeleton branches modeling the front legs are connected differently to the main skeleton.

5.2 Limitations of Our Method

The limitations of our method are as follows:

- It requires a regular 2D/3D grid as the input where the pixels/voxels belonging to the shape is 1 and the remaining background pixels/voxels are 0.
- When the resolution of the input shape is low, v_k field and, hence, the skeleton does not cover the whole shape.
- It is not topology-preserving i.e. topology of the shape and the skeleton might not be the same.
- It does not guarantee the centeredness of the skeleton.
- The skeleton might not be smooth i.e. it might have sharp turns.
- It involves a parameter κ, which is used to determine details of the shape represented by the skeleton.

5.3 3D Shapes

We present our results for 3D shapes under two categories, namely, shapes composed of rectangular prism and general shapes.

5.3.1 Shapes Composed of Rectangular Prism

This category is composed of the shapes that we experiment with to test the behavior of our skeleton model and to address the shapes that are challenging to other skeletonization methods. This category includes five shapes composed of rectangular prism: cube, rectangular prism, rectangular slab, a shape where two cubes are combined, and a comb-like shape which is a combination of rectangular prisms. In Figure 5.8, we present our skeleton model for the shapes in this category.

We observe that our skeleton results for cube and the rectangular prism are similar to the results in 2D shapes of square and rectangle respectively.

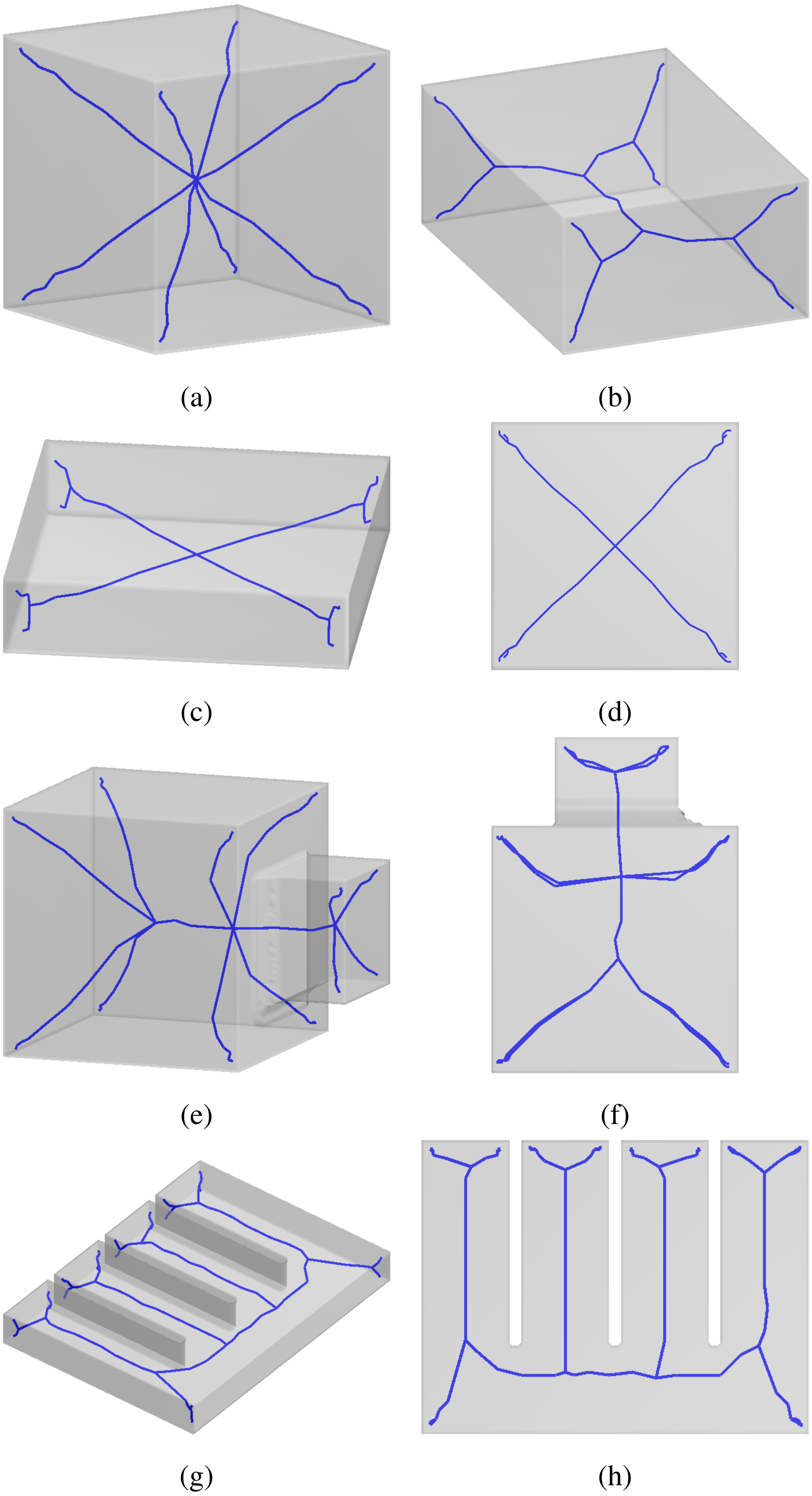
Rectangular Slab refers to a rectangular prism where one of its dimensions only serves the purpose of depth. In [25], the authors mention that, for any curve skeleton method, slab is the most difficult object, because, while we expect methods to form curve skeletons, they instead produce medial surfaces. We see that our method treats rectangular slab as a regular rectangular prism and produces a skeleton accordingly as shown in Figure 5.8 (c)-(d).

Considering the shape with two cubes combined shown in Figure 5.8 (e)-(f), we see that our skeleton method handles the shape as it handles a cube or a rectangular prism, thus our skeleton result reflects the fact that the shape is composed of combination of two cubes. However considering the skeletons obtained using Starlab program from the work [7] in Figure 5.9, we see that Starlab requires four parameters named, omega_L_0, omega_H_0, omega_P_0 and edgelength_TH, and depending on these parameters it produces either missing skeleton branches or extra skeleton branches. For all the skeletons produced by using Starlab, we use the same values for the first three parameters because changing their default values results in not producing a skeleton at all for this shape. Tuning the edgelength_TH parameter results in different skeleton calculations as shown in Figure 5.9 (b)-(e), with larger edgelength_TH values like 0.00421 creates only one curve while decreasing this value gradually introduces new branches to the skeleton, looking at the result produced with edgelength_TH $= 0.00121$ skeleton branches represent the shape itself.

For the comb-like shape shown in Figure 5.8 (g)-(h), our skeleton model produces a skeleton that looks like actual combination of its smaller parts which all five of them are similar to rectangular prism's skeleton that we produce.

5.3.2 General 3D Shapes

General 3D shapes are the shapes that we include to understand our method's strengths and weaknesses on the shapes of objects from different domains.



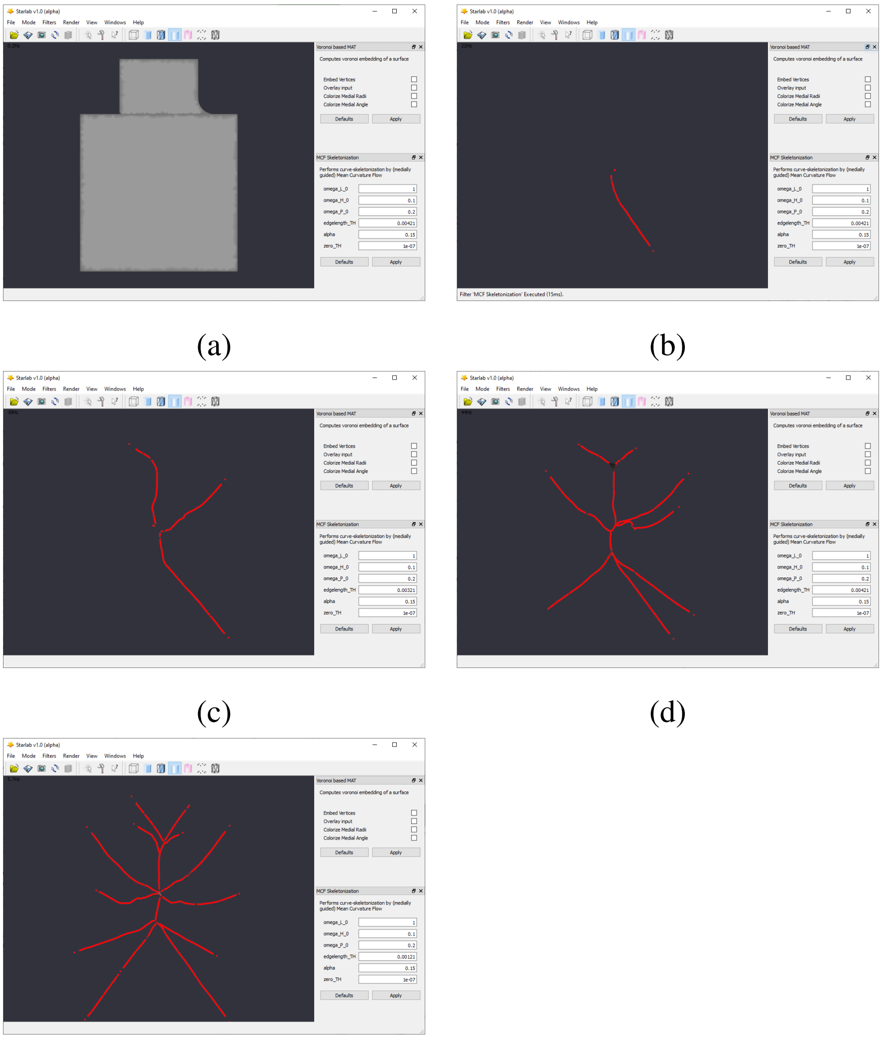
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 5.8: Our skeleton results for (a) Cube (b) Rectangular Prism (c)-(d) Rectangular slab from two different viewpoints (e)-(f) Shape that looks like two cubes combined (with $\kappa = 1/100$) from two different viewpoints (g)-(h) Comb like shape from two different viewpoints.



omega_L_0 1
omega_H_0 0.1
omega_P_0 0.2
edgelength_TH 0.00421
alpha 0.15
zero_TH 1e-07

(a)

edgelength_TH 0.00421

Filter 'MCF Skeletonization' Executed (15ms).

(b)

edgelength_TH 0.00321

(c)

edgelength_TH 0.00421

(d)

edgelength_TH 0.00121

(e)

Figure 5.9: Since Starlab is an external program, we are not able to put the resulting skeletons on the top of the shapes for visualization purposes. (a) Original shape for visualization. (b) Result with edgelength_TH $= 0.00421$. (c) Result with edgelength_TH $= 0.00321$. (d) Result with edgelength_TH $= 0.00221$. (e) Result with edgelength_TH $= 0.00121$.

5.3.2.1 Articulated Objects

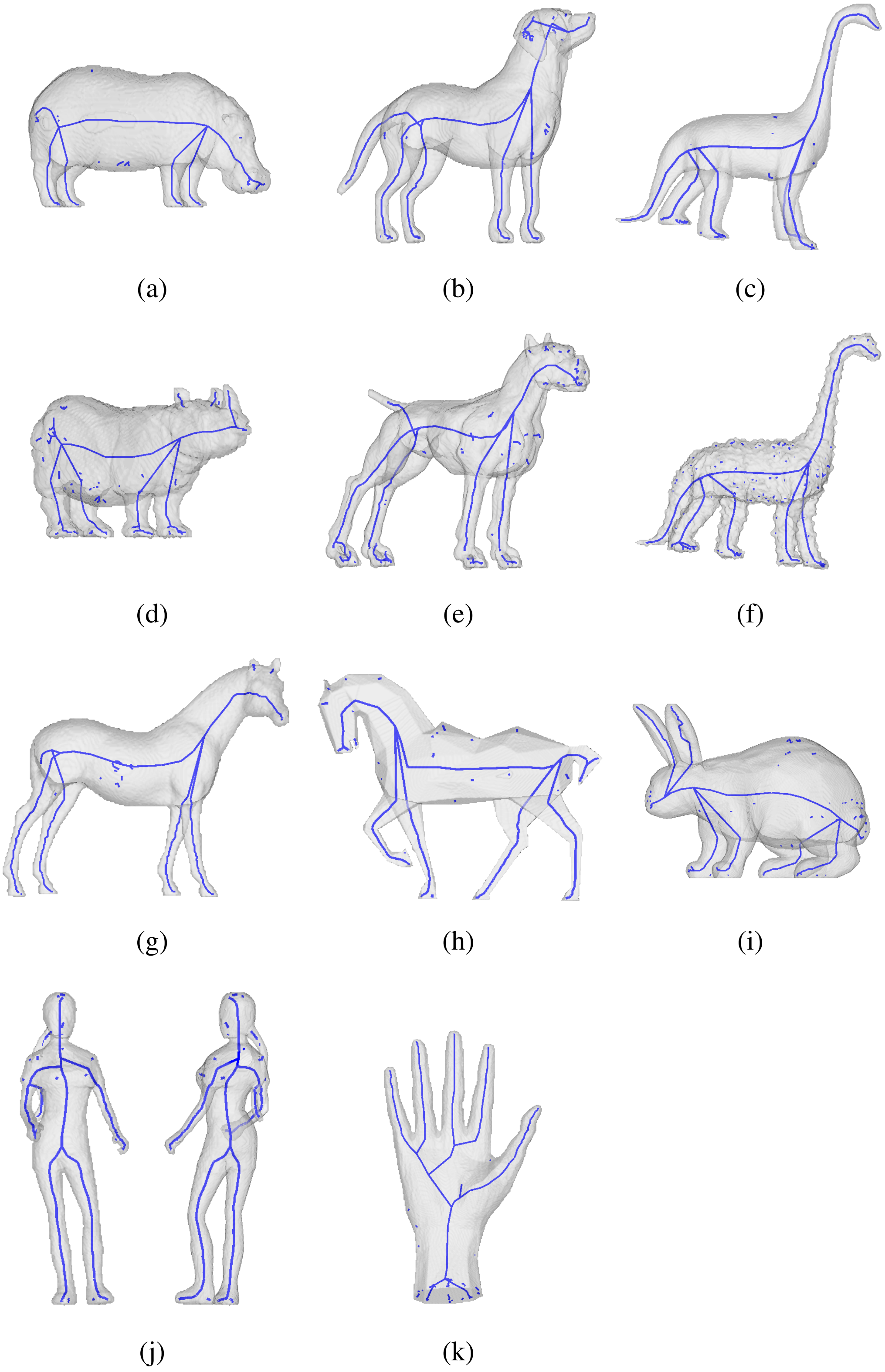
The first group of this section is composed of articulated objects i.e. objects with limbs/protrusions attached to a body e.g. animals, human, hand and etc. (see Figure 5.10 and Figure 5.11). Overall, we observe that main shape components such as body, legs, head, and tail of animal shapes (Figure 5.10 (a)-(h)), body, legs, head, ears and tail of rabbit shape (Figure 5.10 (i)), body, arms, legs, and head of human shapes (Figure 5.10 (j) and Figure 5.11), and fingers, wrist, and palm of hand shape (Figure 5.10 (k)) are successfully captured by our skeleton model.

The shapes in Figure 5.10 (c) and (f) correspond to different poses of a dinosaur shape where the shape in Figure 5.10 (f) has a substantial amount of boundary noise. We see that our results for these two objects are similar and our skeleton model is not affected by the boundary noise. Instead, the boundary noise is modeled via tiny skeleton pieces that could be easily discarded since they are not connected to the main skeleton.

Considering the horse shapes in Figure 5.10 (g) and (h), their skeletons are very similar but the skeleton of the second horse has sharp corners. This difference is due to that the surface of the second horse shape has sharp corners and it is not as smooth as the surface of the first horse.

For better visualization, the skeleton of the woman in Figure 5.10 (j) is shown from two different viewpoints. The woman has a long ponytail whose middle part is combined with her right shoulder. Thus, there are disjoint skeletal branches modeling the beginning and the end of the ponytail. Since the end of the ponytail is as if it is protruding from the right shoulder, its skeletal branch is connected to the skeletal branch passing through the shoulder.

In Figure 5.11, we present our results for the 3D shapes from human category of the dataset [5]. We see that the skeletons obtained for different shapes or different poses of the same shape are consistent with each other except that arms of the human shape in Figure 5.11 (s) are not represented since they are merged with the body.



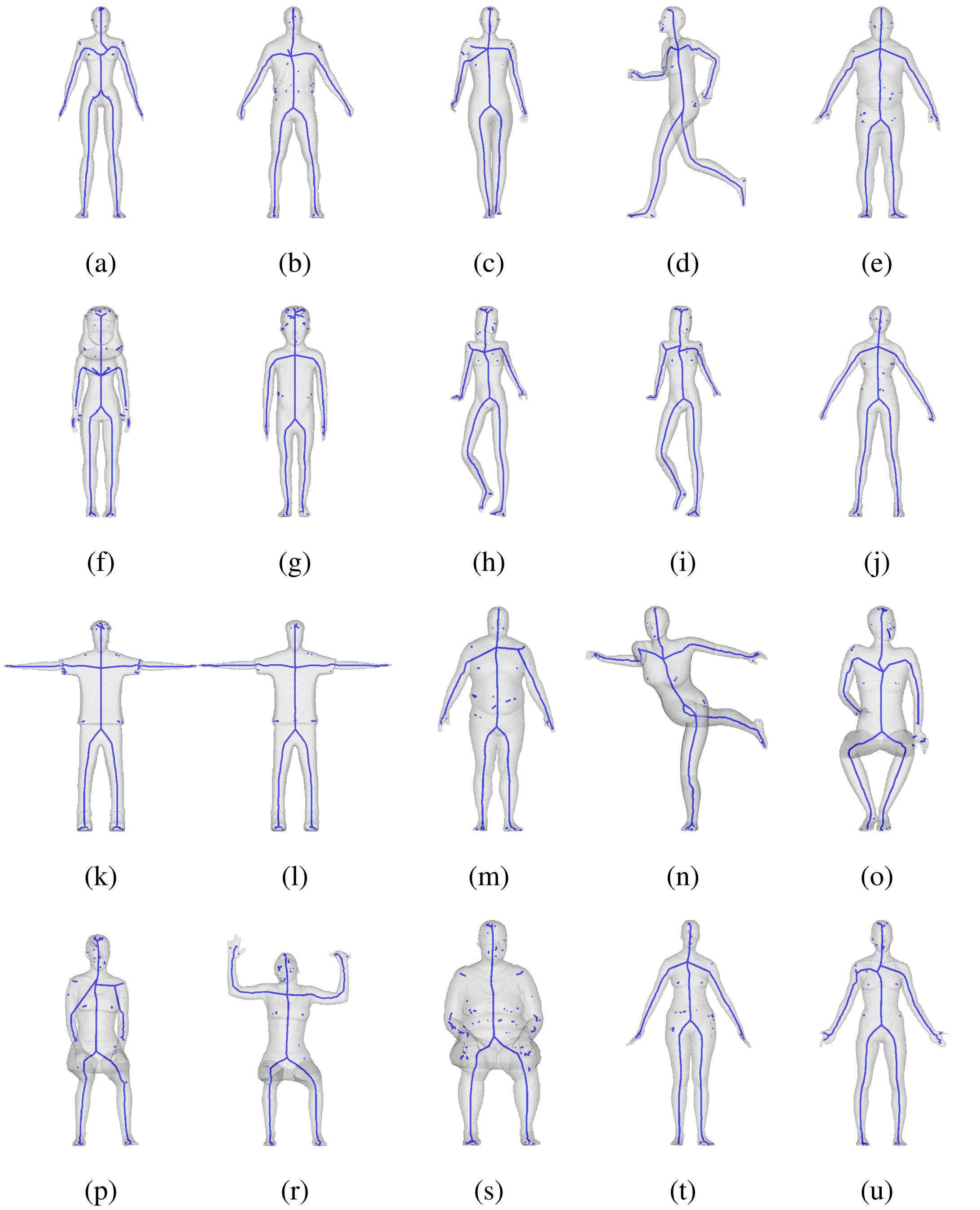
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 5.10: Our skeleton results for articulated 3D shapes.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (r) (s) (t) (u)

Figure 5.11: Our skeleton results for 3D shapes from human category of the dataset [5] where $\kappa = 1/60$ for (c), and $\kappa = 1/50$ for the remaining results.

5.3.2.2 Objects from Medical Field

The second group of this section is composed of objects from the medical field such as coronary artery, bronchial tree, colon, humerus, and ball-and-socket joint (see Figure 5.12).

For the coronary artery and the bronchial tree, our results are visualized from four different viewpoints (see Figure 5.12 (a)-(d), and (e)-(h), respectively). The extracted curve skeleton of the coronary artery in Figure 5.12 (a)-(d) covers all vascular structures including very small arteries branching off the main artery. Our skeleton result for the bronchial tree shown in Figure 5.12 (e)-(h) preserves the order of the branches and it covers all branches of the tree except the region encircled with a red contour in Figure 5.12 (h). This is due to that two branches are combined to form a ring-like structure in that region and the skeleton branches coming from opposite directions are not linked to each other similar to the case in 2D annulus shape.

In Figure 5.12 (i)-(j) and (k)-(l), we present our results for the colon object in two different settings $\kappa = 1/15$ and $\kappa = 1/50$, respectively. The result in each setting is visualized from two different viewpoints. We see that the skeleton obtained with a smaller κ has less number of branches so it is possible to adjust the level of details of the skeleton by changing κ. In both settings, the main skeleton passes through the object from one end to the other end except the small part encircled with a green contour (see Figure 5.12 (j)), which is due to low resolution of the shape volume in that region. Also, there is a hole in the bottom of the shape encircled with a red contour. Our skeleton does not cover all the regions around the hole since it is not topology-preserving.

Considering our results for the humerus, and the ball-and-socket joint in Figure 5.12 (m) and (n), we see that our method enables modeling both the main axis of the shapes and all the bumps and the protrusions attached to them.

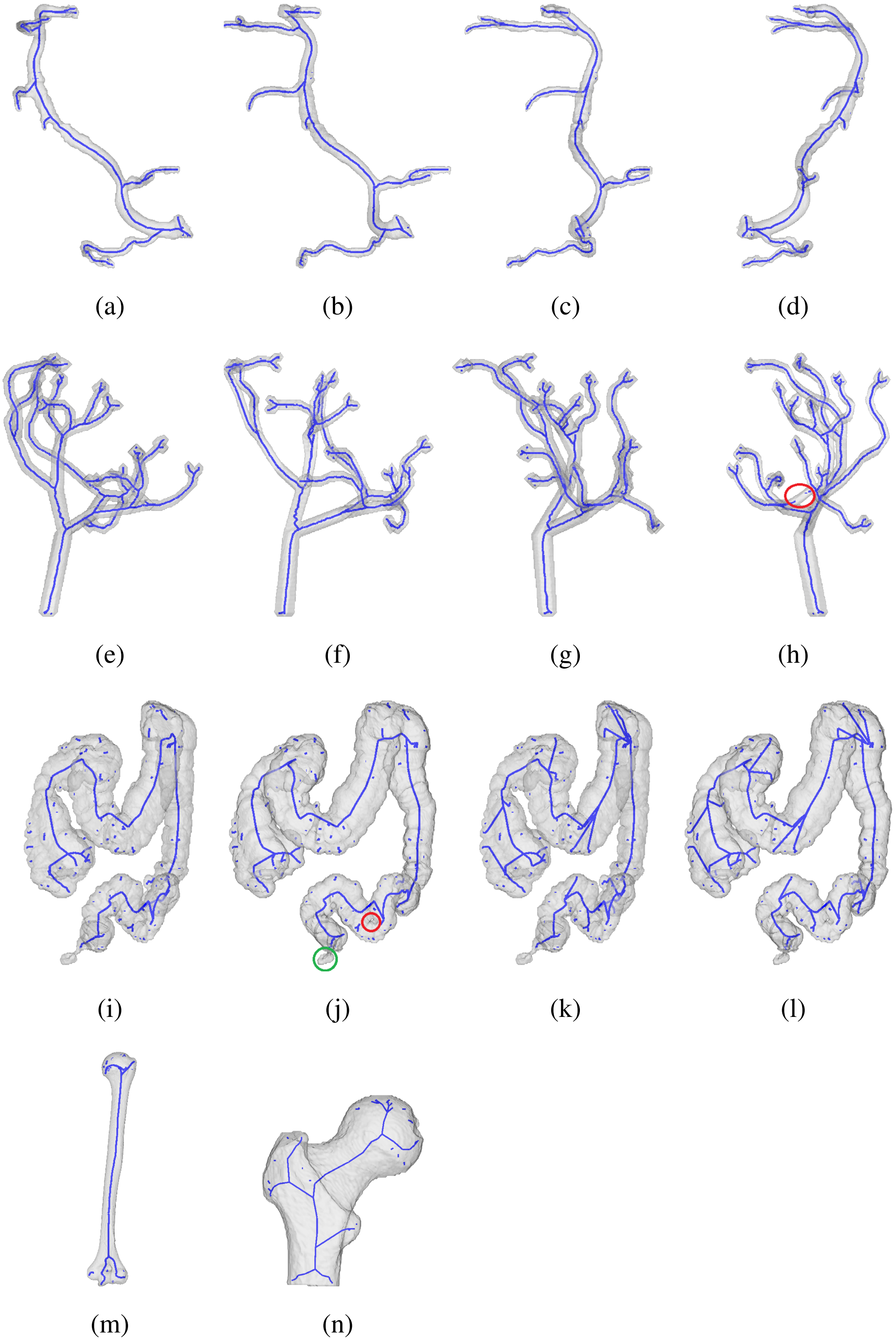


Figure 5.12: Our skeleton results for objects from medical field (a)-(d) coronary artery ($\kappa = 1/150$), (e)-(h) bronchial tree, (i)-(j) colon ($\kappa = 1/15$), (k)-(l) colon ($\kappa = 1/50$), (m) humerus, and (n) ball-and-socket joint.

5.3.2.3 Human-made Objects

The third group of this section is composed of human-made objects such as airplane, biplane, helicopter, hammer, screwdriver, bottle, bench, rocker arm, and rotor (see Figure 5.13).

In Figure 5.13 (a)-(c), we see that all the parts of airplane, biplane, and helicopter objects along with their connections are successfully represented by their skeletons. The only exception is that the skeleton of the helicopter in Figure 5.13 (c) is not connected in the region encircled with a red contour (also, in the region of its symmetric counterpart) due to the lack of topology-preserving property of our method. Also, notice that since the objects are symmetric, their extracted skeletons are symmetric as well.

The skeleton of hammer, screwdriver, and bottle object in Figure 5.13 (d)-(f) involves a long skeletal line corresponding to the symmetry axis of each object. Also, there are skeletal branches connected to the main skeletal line modeling the base of each object and the head of the hammer including both of its claws.

Considering the bench object in Figure 5.13 (g), which looks like a bent and thin rectangular prism, the resulting skeleton is also in accordance with the skeleton of a rectangular prism (see Figure 5.8 (b)). The skeleton of the rocker arm object in Figure 5.13 (h) provides a compact representation of it except that the skeleton must be connected in the region encircled with a red contour and this is one of the future works. The rotor in Figure 5.13 (i) is a thin object with many holes in it. We see that our skeleton covers every detail of the object.

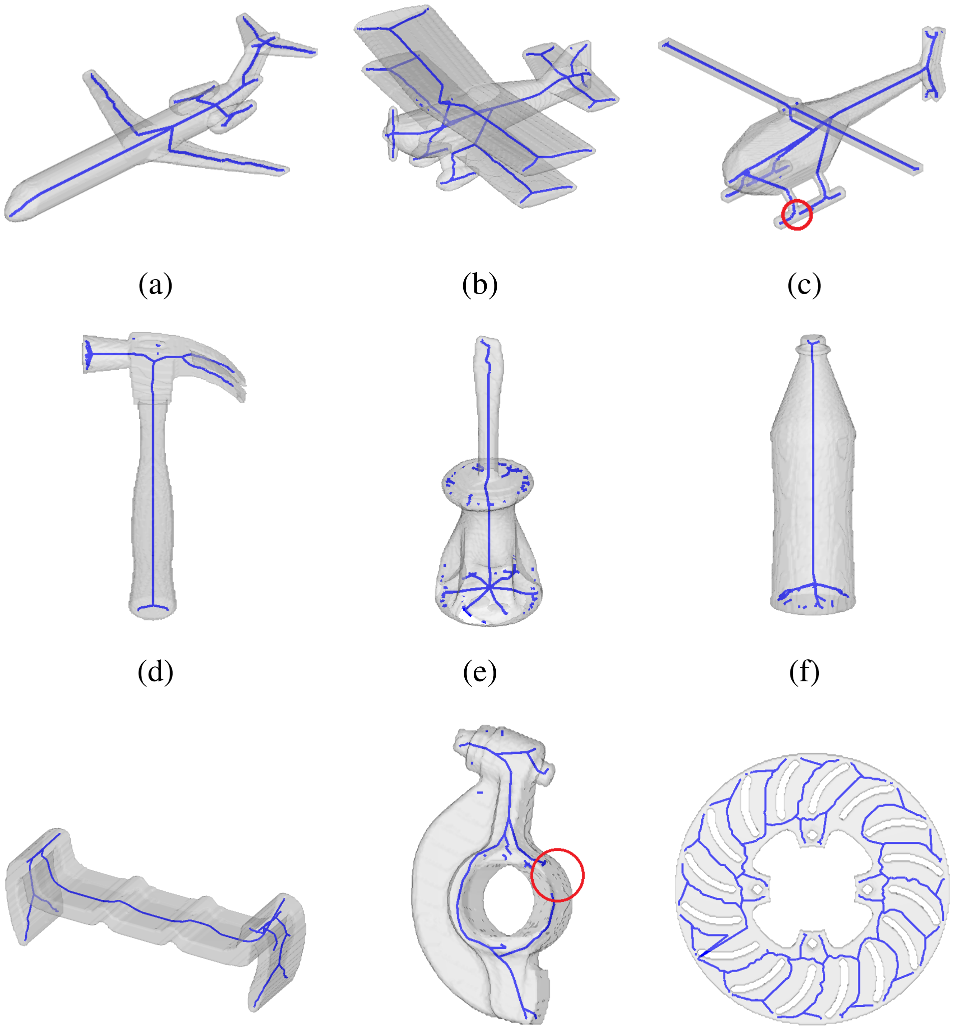


Figure 5.13: Our skeleton results for human-made objects (a) airplane ($\kappa = 1/250$), (b) biplane ($\kappa = 1/100$), (c) helicopter ($\kappa = 1/100$), (d) hammer, (e) screwdriver ($\kappa = 1/15$), (f) bottle, (g) bench ($\kappa = 1/100$), (h) rocker arm, and (i) rotor.

5.4 3D Result Comparison

In this section, we present our results in comparison with other skeletonization methods.

In Figure 5.14, we present the coronary artery object in Figure 5.12, our skeletonization result, and the result obtained using [6, 7]. We see that all vascular structures including very small arteries are represented in our result. However, the parts in the regions encircled by red contours in Figure 5.14 (a) are missed by the method [6, 7]. This is due to that the method [6, 7] involves boundary contraction and it reduces small protrusions or bumps into a single point [17].

In [11], the authors present a survey of the related works on extracting curve skeleton of 3D objects. They divide the methods into four classes, namely, the thinning methods, the methods using distance field, the geometric methods, and the methods using general fields. They provide an implementation of one method from each of the four classes. They note that only the core part of each method is implemented since it is very difficult to obtain the full implementation as each method involves several thresholds and parameter values. In Figure 5.15, we present our skeletonization result and the results of the three methods provided by [11] for the colon object in Figure 5.12. We do not present the result of one of the methods provided by [11] since we are not able to compile it. However, its result for the colon object could be seen in [11]. Also, we are not able to present the result of [6, 7] for the colon object because the program crashes when we run it using its default parameters.

In Figure 5.15, we present the results from two different viewpoints where the results in the same row are visualized from the same viewpoint. The geometric method that uses the orientation-dependent height function and Reeb graph produces a strange result (see Figure 5.15 (b)) since it requires its input oriented in the natural up-down direction. Even if an orientation-independent function is used, it is still possible that skeletal lines are outside the shape volume [11]. Considering the result of the thinning method in Figure 5.15 (c), we see that it is sensitive to the boundary noise and, hence, there are extra skeletal branches towards the details of the shape boundary. Also, note that the thinning methods require post-processing involving pruning and grouping the

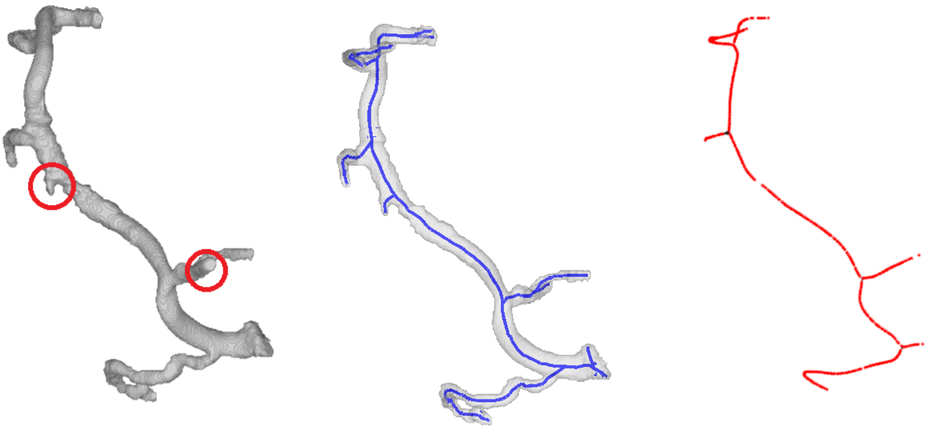


Figure 5.14: (a) Coronary artery object, (b) our result ($\kappa = 1/150$), and (c) the result obtained using [6, 7].

skeletal points. The result of the potential field method in Figure 5.15 (d) is clean and smooth but it involves additional connections between the skeletal lines modeling different parts of the shape. Considering our result in Figure 5.15 (a), we see that it is robust to boundary noise and it does not involve additional connections. However, our skeleton model does not preserve the topology of the shape and it does not represent the small part of the object at the bottom due to the low resolution of the shape volume in this part.

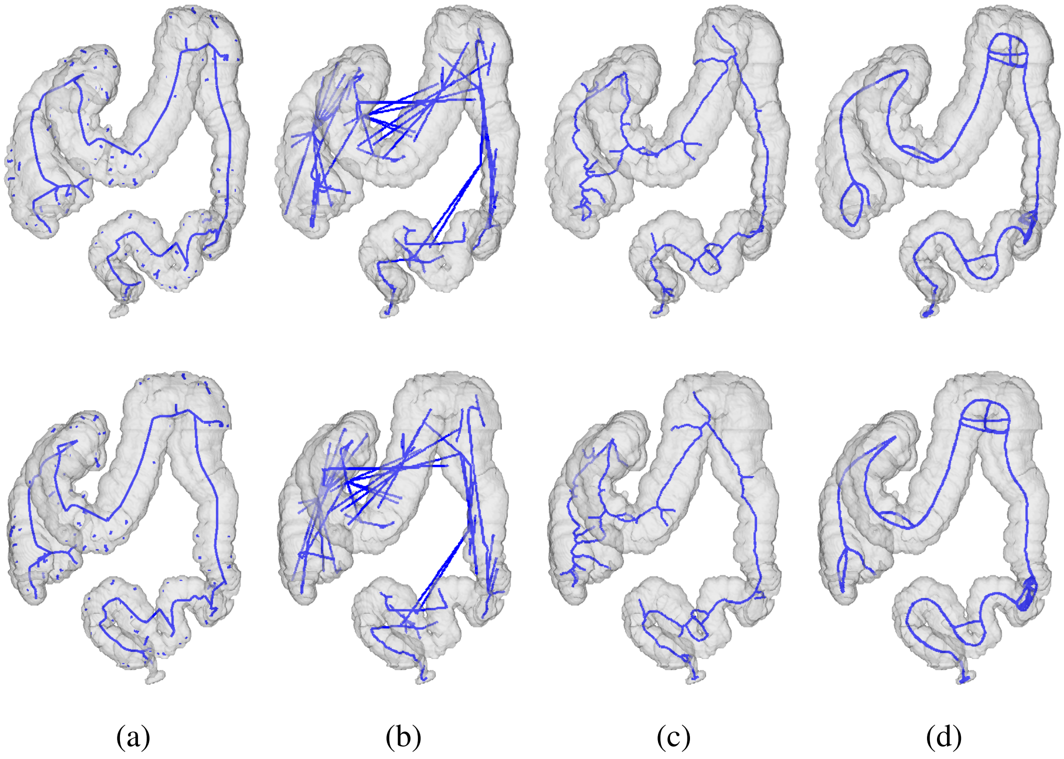


Figure 5.15: For the colon object, (a) our method, (b) geometric method, (c) thinning method, and (d) method using potential field.

CHAPTER 6

CONCLUSION

In this thesis, we present v_k field as a new family of smooth distance fields. The field at each level is obtained by solving the same PDE at the corresponding shape domain. However, the fields at different levels are not the same since the shape domain at level k is obtained by carving the shape domain at its previous level $k-1$ where the carving is guided by the field at level $k-1$. v_0 is defined on the whole shape domain and the initial carving is done from the point where v_0 attains its maximum value. In this way, the carving starts from inside the shape domain. In later levels $k > 1$, the carving is guided by the watershed boundary between the carved region inside the shape and the background region outside the shape considering the field at the corresponding level. The maximum level is reached when the carving stops i.e. either the whole shape domain is carved or the carving can not proceed into very thin regions.

We observe that level curves of the field at each level provide partitioning of the corresponding shape domain similar to the part-coding distance field [1, 2]. We also observe that the carving of the shape domain at level k proceeds toward the center of the parts provided by the level curves of the field at level k. Based on these properties of v_k field, we develop a new method for obtaining a skeletal representation of shapes. Our skeletonization method involves a sequence of intuitive and easy-to-understand steps with a single intuitive parameter κ controlling the level of shape details covered by the resulting skeletal representation. Our constructions are valid for both 2D and 3D shapes.

We experiment with our skeletonization method on several 2D and 3D shapes from different domains. It produces a thin and connected skeletal graph modeling the main shape components and a collection of skeletal segments modeling the boundary texture/noise, which could be easily discarded since they are not connected to the main skeleton. We show that our method produces compatible results when the shapes differ with respect to their orientation, scale, articulation, and boundary detail.

Our skeletonization method does not preserve the shape topology and this is one of the future works. Another future work is to overcome the limitation of the low shape resolution by considering the input shape in increasing resolutions until the whole shape domain is covered. Using the proposed method in an application such as shape matching, shape decomposition, virtual navigation is another future direction.

REFERENCES

[1] S. Tari, "Hierarchical shape decomposition via level sets," in *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pp. 215–225, Springer, 2009.

[2] S. Tari, "Extracting parts of 2D shapes using local and global interactions simultaneously," in *Handbook of Pattern Recognition and Computer Vision*, pp. 283–303, World Scientific, 2010.

[3] C. Aslan and S. Tari, "An axis-based representation for recognition," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, pp. 1339–1346 Vol. 2, 2005.

[4] D. Reniers, J. Van Wijk, and A. Telea, "Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 2, pp. 355–368, 2008.

[5] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3D mesh segmentation," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, aug 2009.

[6] A. Tagliasacchi and I. Alhashim, "Mean curvature skeletons." Accessed Jan. 9, 2023 [Online].

[7] A. Tagliasacchi, I. Alhashim, M. Olson, and H. Zhang, "Mean curvature skeletons," in *Computer Graphics Forum*, vol. 31, pp. 1735–1744, Wiley Online Library, 2012.

[8] H. Blum, "Biological shape and visual science (part i)," *Journal of Theoretical Biology*, vol. 38, no. 2, pp. 205–287, 1973.

[9] N. D. Cornea, D. Silver, and P. Min, "Curve-skeleton applications," in *VIS 05. IEEE Visualization, 2005.*, pp. 95–102, IEEE, 2005.

[10] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, and A. Telea, “3d skeletons: A state-of-the-art report,” in *Computer Graphics Forum*, vol. 35, pp. 573–597, Wiley Online Library, 2016.

[11] N. D. Cornea, D. Silver, and P. Min, “Curve-skeleton properties, applications, and algorithms,” *IEEE Transactions on visualization and computer graphics*, vol. 13, no. 3, p. 530, 2007.

[12] D. G. Morgenthaler, “Three-dimensional simple points: sereial erosion, parallel thinning and skeletonization,” *TR-1005*, 1981.

[13] K. Palágyi and A. Kuba, “A parallel 3D 12-subiteration thinning algorithm,” *Graphical Models and Image Processing*, vol. 61, no. 4, pp. 199–221, 1999.

[14] T. Ju, M. L. Baker, and W. Chiu, “Computing a family of skeletons of volumetric models for shape description,” *Computer-Aided Design*, vol. 39, no. 5, pp. 352–360, 2007.

[15] L. Liu, E. W. Chambers, D. Letscher, and T. Ju, “A simple and robust thinning algorithm on cell complexes,” in *Computer Graphics Forum*, vol. 29, pp. 2253–2260, Wiley Online Library, 2010.

[16] A. C. Jalba, J. Kustra, and A. C. Telea, “Surface and curve skeletonization of large 3d models on the gpu,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1495–1508, 2012.

[17] M. Wei, Q. Wang, Y. Li, W.-M. Pang, L. Liang, J. Wang, K. K. L. Wong, D. Abbott, J. Qin, and J. Wu, “Centerline extraction of vasculature mesh,” *IEEE Access*, vol. 6, pp. 10257–10268, 2018.

[18] Y. Xiao, P. Siebert, and N. Werghi, “A discrete reeb graph approach for the segmentation of human body scans,” in *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, pp. 378–385, IEEE, 2003.

[19] C. Arcelli, G. S. Di Baja, and L. Serino, “Distance-driven skeletonization in voxel images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 709–720, 2010.

[20] G. Borgefors, “On digital distance transforms in three dimensions,” *Computer vision and image understanding*, vol. 64, no. 3, pp. 368–376, 1996.

[21] S. Tari and J. Shah, “Local symmetries of shapes in arbitrary dimension,” in *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pp. 1123–1128, IEEE, 1998.

[22] Z. Tari, J. Shah, and H. Pien, “A computationally efficient shape analysis via level sets,” in *Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis*, pp. 234–243, 1996.

[23] Z. G. Tari, J. Shah, and H. Pien, “Extraction of shape skeletons from grayscale images,” *Computer Vision and Image Understanding*, vol. 66, no. 2, pp. 133–146, 1997.

[24] F. Meyer, “Topographic distance and watershed lines,” *Signal Processing*, vol. 38, pp. 113–125, 1994.

[25] M. S. Hassouna and A. A. Farag, “Variational curve skeletons using gradient vector flow,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2257–2274, 2008.