Flux Graphs for 2D Shape Analysis

Morteza Rezanejad

Master of Computer Science

School of Computer Science

McGill University

Montréal,Quebec

2012-12-13

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

DEDICATION

I would like to dedicate this thesis to my wonderful parents who have always made me feel strong and confident.

ACKNOWLEDGEMENTS

My greatest thanks are to my special advisor Prof. Kaleem Siddiqi, who helped me during these years with the challenges I have faced. He made me feel supported all the time and he helped me in all areas of my graduate life gently. I strongly believe that working under his supervision is one of the very rare chances that a graduate student can be given and for this I am grateful.

I am deeply thankful to all the shape analysis group for creating a stimulating environment for discussion. In particular, I thank Svetlana Stolpner for her openness to sharing her ideas. Special thanks also go to Parya MomayyezSiahkal, Emmanuel Piuze-Phaneuf, Maxime Boucher, and Hervé Lombaert for their help and prompt feedback.

I am specially grateful to Diego Alejandro Macrini for allowing us to use his code to set up the matching framework for flux graphs. Furthermore, my heartfelt thanks go to my dear friend Babak Samari for helping me a lot with his insightful comments, and support.

I would like to thank all of my friends in Montréal city who have been supporting me on a daily basis: Mohammad Bagher Motevali, Danesh Ghafoorzadeh, Vahid Raissi Dehkordi, Danial Alizadeh, Bahareh Ghotbi, Sara Moghtadernejad, Abolfazl Mohebbi, and Sadegh Bolouki. You have made me feel at home!

Last, but not least, my deepest gratitude goes to my family for their unflagging love and support throughout my life; this dissertation would not be possible without

them. Thank you Maman, Baba, Mohammad, Mojtaba, Fereshteh, Azar, and Daee Arash.

ABSTRACT

This thesis considers a method for computing skeletal representations based on the average outward flux (AOF) of the gradient of the Euclidean distance function to the boundary of a 2D object through the boundary of a region that is shrunk. It then shows how the original method, developed by Dimitrov *et al.* [17] can be optimized and made more efficient and proposes an algorithm for computing flux invariants which is a number of times faster. It further exploits a relationship between the AOF and the object angle at endpoints, branch points and regular points of the skeleton to obtain more complete boundary reconstruction results than those demonstrated in prior work. Using this optimized implementation, new measures for skeletal simplification are proposed based on two criteria: the uniqueness of an inscribed disk as a tool for defining salience, and the limiting average outward flux value. The simplified skeleton when abstracted as a directed graph is shown to be far less complex than popular skeletal graphs in the literature, such as the shock graph, by a number of graph complexity measures including: number of nodes, number of edges, depth of the graph, number of skeletal points, and the sum of topological signature vector (TSV) values. We conclude the thesis by applying the simplified graph to a view-based object recognition experiment previously arranged for shock graphs. The results suggest that our new simplified graph yields recognition scores very close to those obtained using shock graphs but with a smaller number of nodes, edges, and skeletal points.

ABRÉGÉ

Ce mémoire propose une méthode pour calculer des représentations squelettiques en fonction du flux moyen décrit par le gradient de la fonction de distance Euclidienne aux limites d'un objet 2D qui rétrécit. La méthode originale développée par Dimitrov *et al.* [17] est ensuite optimisée afin de calculer des invariants de flux plus rapidement. Une relation entre l'AOF et l'angle de l'objet aux extrémités (aux points de branches et des points réguliers du squelette) est exploitée afin d'obtenir une reconstruction plus précises des limites de l'objet par rapport aux travaux précédents. En utilisant cette implémentation optimisée, de nouvelles mesures de simplification de squelettes sont proposées selon deux critères: l'unicité d'un disque inscrit comme un outil permettant de définir la saillance, et la limitation de la moyenne du flux á l'extérieur. Il est démontré que le squelette simplifié, abstrait par un graphe orienté, est beaucoup moins complexe que des graphes squelettiques conventionnels mentionnés dans la littérature, tel que le graphe de choc. Les mesures de complexité de graphe comprennent le nombre de nuds, le nombre de bords, la profondeur du graphe, le nombre de points du squelette et la somme des valeurs du vecteur des signes topologiques (TSV). La thèse se finit en appliquant le graphe simplifié au problème de reconnaissance d'objets basée sur la vue, préalablement adapté pour l'utilisation de graphes de choc. Les résultats suggèrent que notre nouveau graphe simplifié atteint des performances similaires à celles des graphes de choc, mais avec moins de nuds, de bords et de points du squelette plus rapide.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure page

CHAPTER 1
Introduction and Background

Visual shape analysis plays a fundamental role in perception by man and by computer and allows for inferences about properties of objects and scenes in the physical world. Although the problem of form analysis is not always mathematically well defined, researchers have tried to tackle by dividing it into more particular tasks which are easier to solve. Once these subproblems are solved, the solutions can be put together to solve more complex visual shape analysis problems.

One of the essential steps to solve these subproblems is to find a good representation for a shape. It is obvious that some information is lost when a 3-D real world object is projected onto an image plane. This makes the task of finding a robust image-based description for an object's shape challenging. Moreover, developing a geometry to outline the representation in mathematical notations is non trivial.

One of the important ways to recover the shapes of objects in 2D and 3D in computer vision is to develop skeletal or medial representations. In 1967 Harry Blum [7] opened this new chapter in computer vision by introducing medial loci, which were later generalized to skeletons [8], [9], [10]. After that, mathematicians and computer scientists developed these initial ideas and further extended them to describe objects in 2D and 3D images. This chapter presents background material on this subject and is organized as follows: In the following subsection, we focus on the problem of object representation in computer vision. We then review some of the existing methods for computing skeletons in the literature. Afterwards, in Section 1.4, we survey different approaches to handle instability in the computation of skeletons. Finally, in Sections 1.5 and 1.6, we go over the problem of medial graph abstraction and shape matching by reviewing some of the popular approaches in the literature.

1.1 Object Representations

One may ask what features a reliable object representation method should have. A feature that has been always sought is the ability to cover a domain of more general examples. Therefore, the potential to cover a bigger class of instances is an important factor. In addition, it should be possible to extract parts and subparts from the shape representation. In many applications, the mathematical description of the model needs to deform to fit a different setting which requires the description to preserve its fundamental elements in the presence of some deformations. Dimitrov [19] enumerates some of these benchmarks and suggests the following constraints: Completeness, Hierarchy, Invariance, Noise, Metric, and Language.

Contour based representations of outlines for object recognition, which are a popular choice for categorization/recognition using shape information alone, can use either boundary information or boundary information combined with information about the interior of the object. Contour-based approaches use boundary information to extract features that define an object. The Boundary of an object can be broken into parts using a specific measure, or can be used as a whole to derive some feature vectors. Since small variations in the boundary are captured throughout these methods, noise is destructive, and some regularization such as Gaussian smoothing must be utilized. Moreover, defining a notion of a shape part is difficult with representations that exploit just boundary data.

In the region-based approaches, the interior of an object is taken into account, which eases the task of describing parts. In addition, these approaches have the advantage of using topological information of an object structure to define relations between object parts. The biggest problem with region based approaches is that they have a lot information to handle in comparison with contour-based approaches.

Based on different specifications of a visual shape problem, a number of representation methods have been developed in these two categories. Medial representations are one of the popular choices because they take and combine assets from both contour-based and region-based approaches. Medial representations or skeletons were firstly introduced by Blum [7] in 1967, along with the process of generating the medial axis based on his well known *grassfire* analogy. Here the boundary is set on fire and the front advances inward at a constant speed, and as fire fronts meet, skeletal points are created. (see Figure 1–1).

The output of applying a grassfire process on an object is a structure called the *Medial Axis Transform*(MAT) which is a set of medial points within an object along with their distances to the boundary. Since the grassfire process is applicable to all bounded shapes, the MAT is to be considered as a comprehensive representation in visual shape problems. In this thesis, the medial axis is chosen to be the object representation approach. The geometry and methods of computing the medial axis are discussed in the following sections.

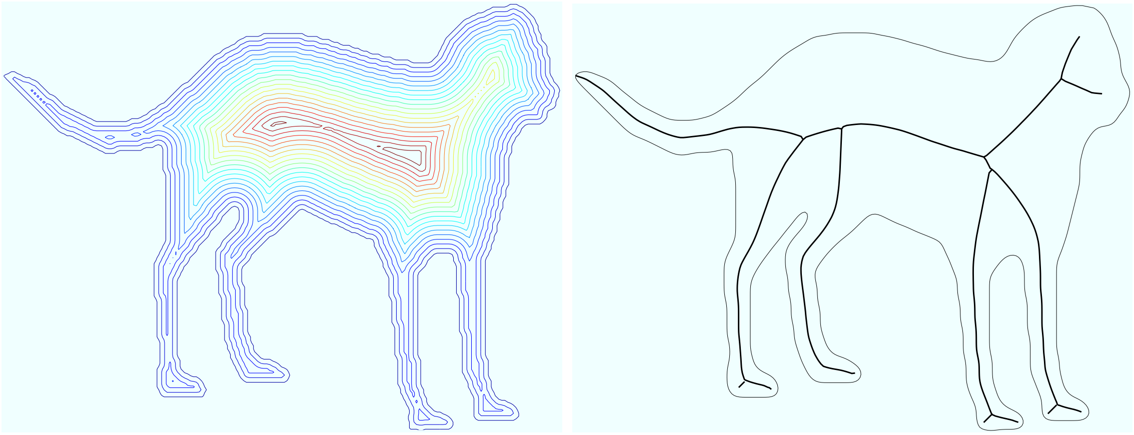


Figure 1–1: LEFT: Iterations of the grassfire process. RIGHT: The resulting skeleton.

1.2 Geometry of Medial Representations

The medial representation is an approach that tries to describe an object as a set of the singularities or quench points of the grassfire transform. These form medial curves about which the outline of the object is locally mirror symmetric. Each point in the interior of such a curve is the result of collision of two distinct boundary points. This set is a union of the centers of the spheres that touch the boundary in two or more locations. In other words, a skeleton consists of a set of points lying inside the shape of an object with the property of having more than one closest point from the boundary. In this section we will discuss the mathematical geometry of a medial axis of an object and we will introduce some notations for the medial representation which will be used frequently later through this thesis.

Definition 1.2.1. Assume an n-dimensional object denoted by Ω with its connected closed boundary denoted by $\partial\Omega \in \mathbb{R}^n$. A closed disk $D \subset \mathbb{R}^n$ is a *maximal inscribed* *disk* in $\partial\Omega$ if $D \subset \Omega$ and for any another disk $D'(\neq D)$ such that $D' \subset \partial\Omega$ then $D' \subset D$.

Definition 1.2.2. Let Ω be an n-dimensional object in $\mathbb{R}^n$. The *Blum medial locus* or *skeleton*, denoted by $Sk(\Omega)$, is the union set of all points representing centers of maximal inscribed disks contained $\partial\Omega$.

The term *Skeletal point* is used to refer to a point on the skeleton of a given shape of an object characterized by a location $\mathbf{p}$ associated with radius r, object angle θ, direction of the unit tangent vector $\mathbf{T}$, and corresponding boundary points $\mathbf{b}^{\pm\mathbf{1}}$ at that point:

$$\theta = \arccos\left(-\frac{dr}{ds}\right) \tag{1.1}$$

where s is the arc length along the medial curve.

The projection $\mathit{\Pi}(x)$ is the set of closest points on the boundary $\partial\Omega$ to x, or $\mathit{\Pi}(x) \overset{\triangle}{=} \{q \in \partial\Omega \ : \ \|x - q\| = \min\{\|x - q\| \ : q \in \partial\Omega\}\}$. For a skeletal point $\mathbf{p}$, the projection set $\mathit{\Pi}(\mathbf{p})$ is equal to the set of points from the boundary touched by the maximal inscribed disk centered at $\mathbf{p}$ (see Figure 1–2).

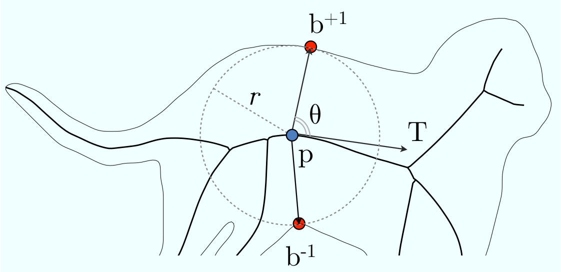
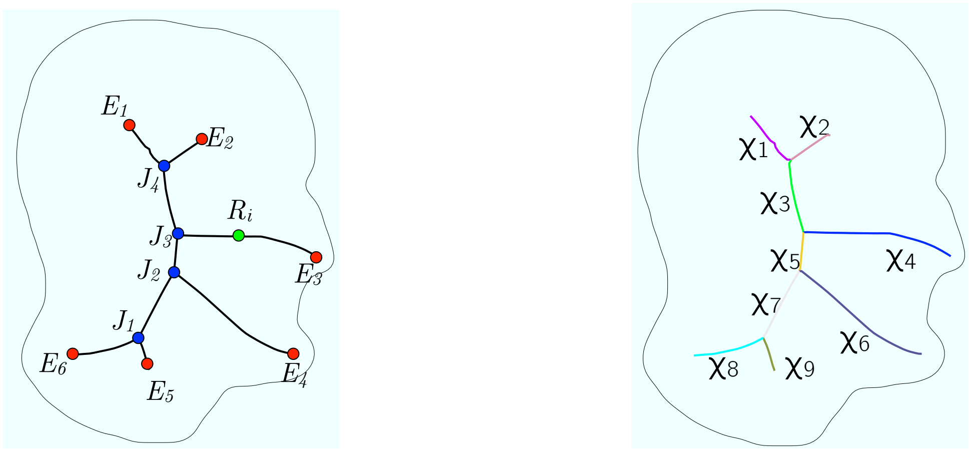


Figure 1–2: Local geometry of a maximal inscribed disk centered at the skeletal point $\mathbf{p}$ with radius r and the object angle θ. The maximal inscribed disk touches the boundary at two points $\mathbf{b}^{\pm\mathbf{1}}$ ($\mathit{\Pi}(\mathbf{p}) = \{\mathbf{b}^{+\mathbf{1}}, \mathbf{b}^{-\mathbf{1}}\}$) (adapted from [35]).

Another definition of the medial axis is given by the set of interior points of a shape with more than one corresponding closest boundary point in the sense of Euclidean distance. This definition is known as “Maxwell set” definition of the medial locus [28]. According to this definition, each skeletal point $\mathbf{p} \in Sk(\Omega)$ must have at least two closest boundary points ($|\Pi_{\Omega}(\mathbf{p})| \geq 2$).

Assume a computed skeleton $Sk(\Omega) \in \mathbb{R}^2$ for a 2-dimensional object Ω. Each skeletal point $\mathbf{p}$ on this skeleton can have one, two, or more skeletal points as its neighbors. In chapter 2, a method of skeletonization in $\mathbb{R}^2$ is discussed and each skeletal point is labeled as one of three different types depending upon the number of neighbors that it can have: regular points, junction points and end points. Figure 1–3a shows these different types on a given shape. Properties of these different types are reviewed in detail in chapter 2.



(a)

(b)

Figure 1–3: LEFT: Different types of a skeletal point are shown on the skeleton for a shape representing a head. E_i’s drawn in red represent endpoints, J_i’s drawn in blue represent junction points and all other skeletal points R_i are regular points. RIGHT: Skeletal branches of the same medial axis are represented as a set of χ_i’s.

From a different point of view, the $Sk(\Omega)$ consists a set of single segments called skeletal branches that join to each other at branch points and form the complete skeleton. A *skeletal branch* denoted by χ is a set of contiguous regular points from the skeleton that lies between a pair of junction points, a pair of end points or and end point and a junction point,

$$\chi = \{\mathbf{p}_1\mathbf{p}_2 \ ... \ \mathbf{p}_{k-1}\mathbf{p}_k\}, \tag{1.2}$$

where all points other than the ends are regular points. In Figure 1–3b skeletal branches of a shape are illustrated.

A full skeleton representation is a union of all skeletal branches:

$$Sk(\Omega) = \bigcup_{i=1}^{n} \chi_i, \tag{1.3}$$

where n is the number of skeletal branches. It is obvious that except for the case that the skeleton is a single skeletal branch, all skeletal branches share at least one junction point with another skeletal branch.

1.3 Computing Skeletons

In the past decades, there has been a lot of interest in developing new algorithms of skeletonization for computer vision and shape analysis applications. A wide variety of methods have been suggested which explore boundary information in order to come up with medial loci which closely represent the real skeleton, therefore, discretizing the boundary is the first step for many of these approaches. A big challenge which the skeletonization algorithms face is the sensitivity of their final medial representation to the extracted boundary information (e.g. [12], [29], and [30]). Consequently, a fine representation of the boundary is a necessity for all skeletonization arrangements. Most of the time, the goal is not to find the exact locations of medial loci, but rather to find nearby points which are fairly close to the real skeleton with the topology of the skeleton being stable. The concern to find a stable medial branching for a shape comes along with the mentioned goal. The challenge exists and efforts to address this issue have led to many different methods of skeletonization. It is not possible for us to review all the existing approaches, but here we list four popular categories of methods.

1.3.1 Thinning Approaches

The first category of approaches proposed to compute skeletons is based on using topological thinning to generate medial loci. The general idea behind these methods is to reduce object volume gradually to get a thinned shape in the form of a skeleton ([33],[20]). These ideas are usually implemented iteratively, and in each iteration an external volumetric layer of the object is peeled away from the shape. At the final phase, the remaining locus of voxels represents the medial locus. Since in these algorithms the topology of the main shape will not change and the original shape is maintained, their output representations are good means for object recognition and classification. They are also very efficient in the sense of computation time. Figure 1–4 provides an example of a skeleton generated by a thinning approach.

1.3.2 Distance Map Methods

The distance map, also known as the distance transform, is one of the popular object representation approaches. In a distance map representation, each pixel is labeled with the distance to the nearest point from the boundary. Whether the

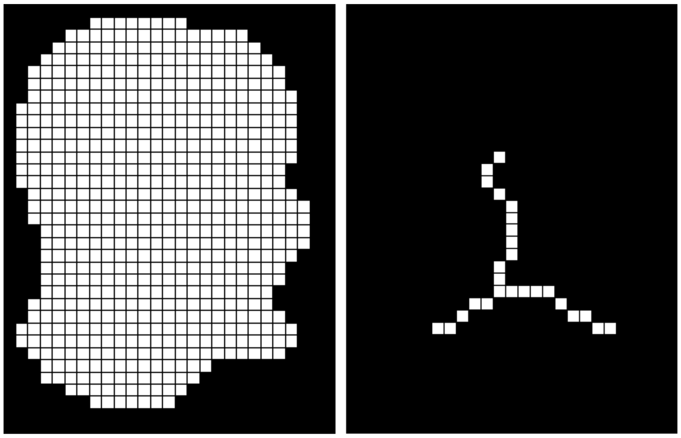


Figure 1–4: An example of skeletonization with a thinning approach. The algorithm has been applied on a binary image and the resulting connected skeleton is shown. The figures above were generated by the thinning skeletonization package implemented by Zhengguo [40].

considered pixel is inside or outside of the object's boundary, the distance labels can be positive or negative. Methods computing skeletons based on the distance map usually carry out the following steps:

1. The boundary of the input object (Ω) is computed ($\partial\Omega$).
2. Assuming a distance metric (e.g. Euclidean distance), a distance map representation (D) is derived for the given object based upon distances to the computed boundary. The distance map must satisfy the condition that boundary points get zero distance ($\mathbf{p} \in \partial\Omega \rightarrow D(p) = 0$).
3. The distance map (D) is explored to find candidate skeletal points.

1.3.3 Voronoi Diagram

To understand methods which use the Voronoi diagram to extract the medial axis, we briefly review this construction. The Voronoi diagram is a configuration between a set of points and the space around them. Give a finite set of points such as $\{p_1, p_2, \ldots, p_n\}$, the following rules are used to divide the space into a set of partitions $\{S_1, S_2, \ldots, S_n\}$ where each space S_i is associated with a particular point p_i.

- S_i is associated with point p_i, therefore $p_i \in S_i$
- Representing the union of all spaces as $M = \bigcup_{i=1}^{n} S_i$, the following rule is applied:

$$S_i = \{x \in M | distance(x, p_i) \leq distance(x, p_j) \text{ for all } j \neq i\}$$

As an example see Figure 1–5.

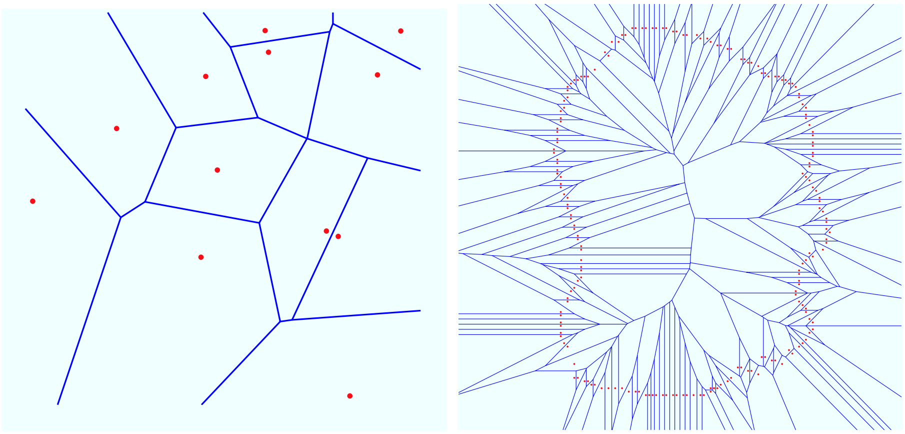


Figure 1–5: Examples of Voronoi diagrams. LEFT: Voronoi Diagram of 12 randomly generated points. RIGHT: Voronoi Diagram of points sampled from the boundary of a shape. The internal portion of the Voronoi Diagram approaches the skeleton as the boundary is sampled densely. These figures are generated using algorithms implemented in MATLAB.

Kirkpatrick *et al.* were the first to propose methods for skeletonizing arbitrary n-line polygonal figures using generalized Voronoi diagrams [22]. Approaches using Voronoi diagrams first generate a set of sampled points from the boundary of a given object, then make the skeleton based on the Voronoi diagram of that set (e.g. [37], [31], and [2]). When the set of input points for Voronoi is sampled from a shape contour, then the relationship between the Voronoi diagram and the skeleton becomes visible. As shown in Figure 1–5, by increasing the number of sampled boundary points the internal Voronoi edges converge towards skeletal points. Topologically, when the gap between sampled boundary points goes to zero, the remaining diagram represents a rather complete skeleton ([11]).

One of the advantages of Voronoi diagram approaches is that the generated skeleton in connected. Besides, the reconstructed shape from the skeleton is very close to the original one, which means that a duality criterion is being addressed by such approaches. Due to these properties and considering the fast processing time, Voronoi diagram approaches are good candidates for skeletonization in many visual shape problems. One of the key points to get an accurate medial representation in this category of skeletonization methods is to sample points of the bounding contour as finely as possible.

1.4 Finding Instabilities

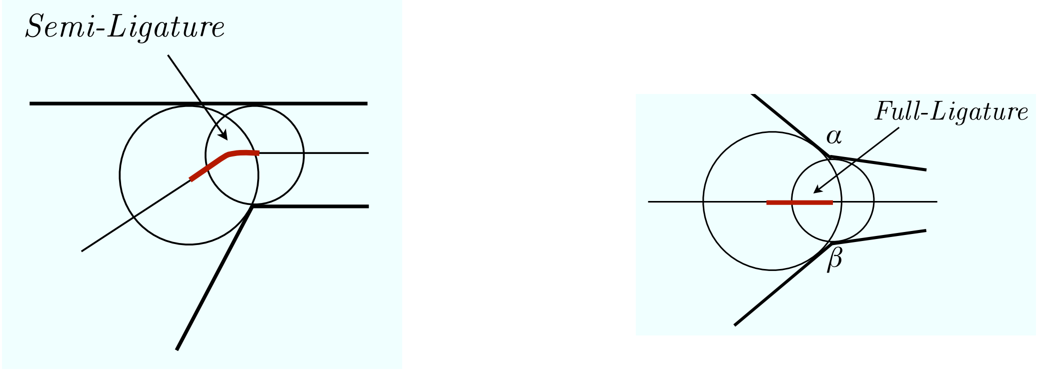
Deploying the MAT (Medial Axis Transform) in problems of computer vision requires stable computation of the skeleton. As previously mentioned, many skeletonizations methods suffer from instability such that small deformations can change the branching structure of the output. This challenge suggests that attention should be paid to the problem of characterizing instabilities on a medial axis. This section reviews methods which address this problem. We first start with a definition for stability and instability in the context of analyzing skeletons, and later review some of the basic approaches suggested in the literature.

1.4.1 What is Ligature?

In stability analysis of the medial axis, the main question is what key attributes should a reliable shape representation have and how we can go from an informal definition to a topological one which is also mathematically justifiable? There have been different studies in the literature which have investigated this subject analytically ([31],[32],[4], and [36]).

The notion of ligature was introduced by Blum [8] for the first time in 1973. He defined finite length fragments of the medial axis that were generated from a single boundary point as *ligature parts*. This definition can be extended to include all sensitive portions of the skeleton which are not topologically stable in the presence of small deformations of the shape. Generally, ligature structures include parts of the medial axis that form loci from the skeleton that are associated with concave chunks of the boundary of the shape. If the boundary of that object is reconstructed upon the remaining salient parts of the medial axis, the new boundary would be approximately the same as the original one.

Different approaches have been proposed in the literature that try to identify points and segments that are more likely to be unstable. These methods try to reduce the defect of ligature points by pruning the skeleton to get a set of reliable parts that can be organized later into a data structure suitable for visual shape problems. In the following sections, we review some of the topics and themes which are closer to the subject of this thesis.



(a) Semiligature

(b) Full ligature

Figure 1–6: Examples of Semi-Ligature and Full-Ligature Parts (adapted from [4]).

1.4.2 Blum's ligature analysis

In this subsection we review Blum's ligature idea [8] by analyzing the behavior of curvature of boundary points. August *et al.* [4] formulated this problem in the context of perceptual shape abstraction. The algorithm makes a set of all concave corners on the boundary with local minimum curvature (denoted by V) at the firs step. Then, for each skeletal point $\mathbf{p}$ that has two closest boundary point $\mathit{\Pi}(\mathbf{p}) = \{a, b\}$, one of the three following conditions are applied:

1. $\mathbf{p}$ is non ligature: V does not contain either a or b ($a \notin V$, $b \notin V$).
2. $\mathbf{p}$ is semi-ligature: Either a or b is in V and the other one is not ($a \notin V$, $b \in V$ or $a \in V$, $b \notin V$).
3. $\mathbf{p}$ is full-ligature: Both a and b belong to V ($\{a, b\} \in V$)

Simplifying the above statement, we may label portions of thes medial axis that correspond to at least one concave corner, a ligature part (See Figure 1–6).

The results of Blum's ligature are promising since they highlighted points with low level of representation information. Despite the important contribution they have made, the proposed method suffers from a serious problem of local sensitivity. Applying derivatives to the values of curvature to compute a local minimum along a discretized boundary, can be sensitive to noise and discretization artifacts. Besides, there are cases where the topology of branching will break down the stability of the method and will cause wrong or imprecise identification of ligature parts.

1.4.3 Boundary to Axis Ratio

Another approach to handle instabilities in a medial axis is to use the *boundary to axis* (BAR) ratio, which is the ratio of the associated boundary length to that considered medial segment length, as the interval on the skeleton shrinks to zero, is called BAR ratio (denoted by δ).

Topologically, a precise skeleton is equivalent to its original shape. This means that having the skeleton of an object Ω, the original can be completely reconstructed. A question which arises is that, if we take a small fragment on the skeleton, $\Delta Sk(\Omega)$, how long is the corresponding length segment on the boundary, $\Delta\partial\Omega$? In real applications, the BAR ratio $\left(\delta = \lim\limits_{\Delta\partial\Omega\to 0} \dfrac{\Delta Sk(\Omega)}{\Delta\partial\Omega}\right)$ varies according to the convexity or concavity of the original boundary segment. The following scenarios may actually occur for each side of the boundary with respect to the skeleton:

1. $\delta < 1$: at the considered interval, there is a local concavity at that side of the boundary. This suggests that eliminating such portions of the skeleton results in a smaller portion of the boundary being lost. Therefore, these points are to be considered as good candidates for ligature elements.
2. $\delta \geq 1$: at the considered location, we have a local elongated part, or a local convexity. This suggests that eliminating the corresponding skeletal points

results in a larger portion of the boundary being lost. Therefore, these points are considered to be more important for shape representation than the previous category.

An example of ligature analysis with the BAR method is shown in Figure 1–7. The skeletal points drawn have BAR ratios bigger than a certain threshold value between [0,1].

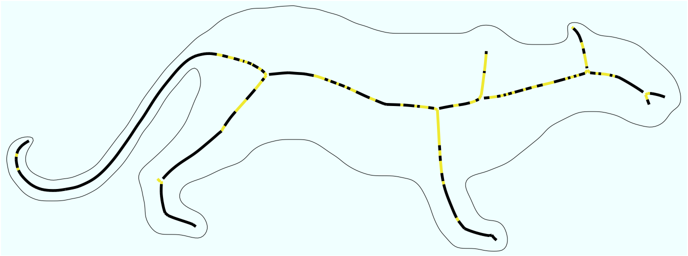


Figure 1–7: An example of ligature analysis using the BAR ratio on a shape of a dog. The yellow points show parts from the boundary where the BAR ratio is lower than a chosen threshold ($\delta \leq 0.75$ for this example).

Some methods of ligature analysis ([3], [24]) combine BAR ratio with other measures, so they would be able to consider other criteria for saliency.

1.4.4 Bone Graph Analysis

One of the reliable methods reported in the literature that uses BAR analysis is the *Bone Graph* ([24],[23], and [26]). This method of finding instabilities consists of two main steps:

1. Initial Ligature Detection: After computing the skeleton of the given shape, skeletal points that form ligature are identified as follows: First, all points that have BAR value less than a certain threshold (less than one) are found. The smaller the threshold the more points there are that are considered for removal.

Second, the algorithm tries to retrieve the longest intervals from the set with negative curvature for all points. Segments with negative curvature at both sides are assumed as full ligature segments and ones with negative curvature at one side are assumed as semi ligature segments.

2. Restoring Over- and Undersegmented Parts: According to the branching topology of the skeleton and different ligature segments detected, the algorithm looks for candidate situations, in which, branches at either side of a junction can merge. This post-processing procedure is termed *Detecting Protrusions* in bone graph analysis. The first step of detecting protrusions is the labeling all junction points. The main intuition of labeling is to find ligature parts extracted from intervals derived from incident branches. To achieve this goal, Macrini *et al.* [26] try to formulate all the different possibilities for junction points and derive all possible situations under which two branches can merge. At the final stage, a dependency graph is made according to the topology of the skeleton and the links created through the merging process.

1.5 Medial Graph Abstraction

Having reviewed methods to handle skeletal instabilities in the previous section, we now attend to the next step which is to define a part-structure for a given object, based on salient portions of the medial axis. We shall refer to this process as *Medial Graph Abstraction.*

In mathematics, a *graph* is a tool to represent a set of objects where some of them are connected to one another by links called edges. Formally a graph $G = (V, E)$, consists of two elements. The first one V is a set of vertices or nodes which represents objects, whereas the second element E is a set of edges or links between those nodes.

Each medial axis includes a number of skeletal branches connected to each other, where each of them represents a part of its corresponding object. This means that the medial axis representation divides objects into a union of parts. Figure 1–8 shows a shape of dog, in which, the shape silhouette is decomposed into some connected regions. Intuitively, to represent this shape as a graph, for each of these parts a node is assigned and connections between parts form edges between these nodes.

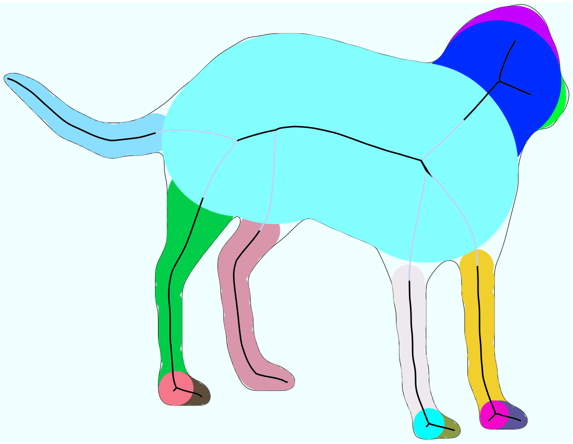


Figure 1–8: A shape of a dog represented in a set of separate parts. The figure is generated by our implementation for flux graph abstraction, which will be developed in chapter 3 in detail.

Three different approaches that try to illustrate a medial axis as a graph are addressed in subsections 1.5.1, 1.5.2, and 1.4.4.

1.5.1 Skeletal Graphs

Given a medial representation for a shape, a trivial approach to extract a graph out of the skeleton is to consider a node for each skeletal branch and an edge for each connection between pairs of these skeletal branches:

$$G = \{(V, E)\} \tag{1.4}$$

where

$$V = \{v_i | v_i = \chi_i \in Sk(X)\} \tag{1.5}$$

and

$$E = \{(v_i, v_j) | \text{ where } (\chi_i \cap \chi_j) \neq \phi\} \tag{1.6}$$

This approach results in a simple, but also powerful graph abstraction (Dimitrov *et al.* [18]).

1.5.2 Shock Graph

Blum *et al.* [8] explained how skeletal points appear when the grassfire flow algorithm is applied to the object contour. Siddiqi *et al.* [36] classified skeletal points into different shock level elements according to their evolutionary appearance as singularities or shocks during the grassfire flow. These four levels of shocks are defined in bellow:

1. First-Order: A skeletal point is considered as a first-order shock if the radii of inscribed maximal disks within a neighborhood around that skeletal point changes monotonically (is increasing or decreasing).
2. Second-Order: A skeletal point is considered as a second-order shock if the radius of its inscribed maximal disk approaches a local minimum along a skeletal branch.

3. Third-Order: A skeletal point is considered as a third-order shock if the radii of inscribed maximal disks within a neighborhood around that skeletal point does not change.
4. Fourth-Order: A skeletal point is considered as a fourth-order shock if the radius of its inscribed maximal disk is a local maximum.

Figure 1–9 shows an example of different shock orders.

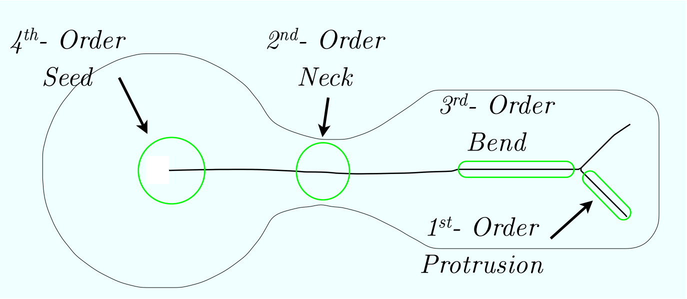


Figure 1–9: Different shock types shown on an arbitrary object.

Through a merging process, adjacent skeletal points of the same shock order are merged to make a node. Each node is labeled based on the shock order of its skeletal points. The union of these nodes gives the set of vertices for the shock graph. For the set of edges, connectivity between nodes is determined according to the order of shocks and their topological structure in the skeleton [36]. This is closely connected to a notion of entry-level categories for shape [36].

1.5.3 Bone Graph

The third graph reviewed in this thesis is the *Bone Graph*. Previously in section 1.4.4 we discussed how the bone graph analysis of ligature detection is applied on a given skeleton. Here once the protrusion detection process is completed and also the dependency graph is created, the merging of host branches begins. After this, a branch merging process is completed and the shape decomposition process is then completed by recovering shape parts.

Like for any other graphs, we must define a set of nodes and also a set of edges to construct a bone graph for a given shape. Each salient segment in the bone graph is called a bone part and represents a node in the derived graph. To represent the hierarchical relations between bones, a map of bone to bone attachments must be made. Each attachment between a pair of bones makes an edge for the bone graph. Attachments are according to the topology of nodes and their connectivities on the skeleton. The parent-child relation for each edge is defined according to the magnitude of the radius of the skeletal point closest to the ligature point from the bones on either side of the attachment, which is a local estimate of part size and is similar to what is used for the shock graph. Besides, each edge is weighted according to the position of nodes and their spatial properties in the skeleton. The attachments are formed based on particular conditions: those formed by segments that are placed at both sides of a junction point, those formed by a merging process in ligature analysis, or those which lie at different sides of a neck shape part. According to the condition by which each attachment is formed, its corresponding edge is weighted by an attribute between $[-1\ 1]$.

1.6 Skeletal Shape Matching

Skeletal Graphs are good candidates to tackle the problem of shape matching. In the previous section, we reviewed different methods of skeletal abstraction using graphs. The first step in these approaches, is to develop a matching process between different graphs, where a measure of similarity or dissimilarity is determined when the matching is done. In the following section, we review some of the skeletal shape matching approaches in the literature.

1.6.1 Shock Graph and Bone Graph Matching

The general aim in shock graph matching [36] is to develop a matching algorithm that can be applied to the shock graphs extracted from binary shapes. The idea is briefly described bellow:

- First shock trees are generated according to the extracted shock graphs. Existing loops (without considering directions over edges) in the shock graphs are eliminated by duplicating tips of loops.
- An iterative process is used to compute the largest common subtree between the two trees.

In terms of eigenvalue characterization, the connectivity between nodes can be represented as a $\{0, 1\}$ adjacency matrix with the property that the rooted trees extracted can have any reordering because they are invariant to similarity transforms.

For bone graphs, Macrini *et al.* [26] followed the same matching framework used for shock graphs. Although this framework ignores edge attribute weights presented in bone graphs, they carried out the comparison by partitioning bone nodes into shock parts and building a look-alike node similarity function.

1.6.2 Path Similarity Approaches

Many approaches to the problem of shape matching fall into the group of methods in which the shapes are matched by comparing their representing graphs. But other than these approaches, there are some reported methods in the literature that tackle the problem in a different way. One category of these methods are the *path similarity approaches* [39], [6], [5], [38].

The general idea is to match the shortest paths that connect skeletal end points. This is done by mapping the dissimilarity between two given skeletons to the dissimilarity measures computed between each of the two sets of nodes. It is obvious that for two given examples with respectively m and n nodes, a dissimilarity matrix $(C(G, G'))$ is made by $m \times n$ elements, where G and G' represent skeletal graphs extracted from two given objects.

To compute the elements of this matrix first a matrix is computed for every two matching nodes, (v, v'), as $pd(v, v')$. Each element of $pd(v, v')$ is a dissimilarity value, e_{ij}, measuring the distance between two paths, one originating from v and ending at the node with index i from graph G and the other one originating from v' and ending at the node with index j from graph G'. Matrix $pd(v, v')$ is a dissimilarity matrix from which the best candidate paths for the matching process can be found. To choose the best nominee, an elastic matching algorithm named *optimal subsequence bijection* has been proposed and is used, which is in the spirit of other dynamic programming algorithms. The best matching path is a sequence of nodes matched along the matrix $C(G, G')$ with a matching score that is considered as a similarity measure.

In fact, in order to obtain a similarity measure between two shapes, these methods consider the shortest path between skeletal endpoints. These shortest paths are compared while considering that the structural graphs of different examples may vary. Although, the shortest path descriptors can result in a precise matching for many examples, the approach is more of a technically engineered system than a topologically justified method. It is unclear whether such methods are capable of handling transformations which can change the topology of the skeleton to a great extent. Besides, it is not well established how these methods handle the structure where two skeletal graphs having different numbers of nodes.

1.6.3 Many to Many Correspondence Graph Matching

Most approaches proposed for skeletal graph matching in the literature are based on a one-to-one correspondence between local elements. There is a category of matching approaches that consider more elements for comparison purposes (e.g. [16], [14], [15], [13], and [21]) when matching graphs. These techniques fall under the category of many-to-many correspondence matching methods. The many-to-many matching process iteratively assigns zero or more nodes from the second example to a node considered from the first examples. Then according to the matching result a pair-wise region abstraction procedure is applied to generate templates by combining the matched examples.

One of the main contributions of this approach is its strength to learn class prototypes, because deformations that separate volumetric nodes resulting in skeletons with more branches and nodes are easily abstracted, are hierarchical and are learned in the context of constructing class prototypes.

1.7 Contributions and Outline of this Thesis

The previous subsections have reviewed several topics of relevance to medial representations and their use in graph-based approaches to object recognition. We now enumerate the main contributions of the present thesis. First, previous approaches to compute flux-based skeletons and to use them for boundary reconstruction are not entirely complete and are computationally expensive. This thesis proposes to address these problems by suggesting a new extended version with full boundary representation, and by restricting the set of potential closest boundary points for the brute-force algorithm used in computation of the distance transform. These developments are discussed in Chapter 2. Second, a new approach of ligature analysis for simplifying skeletons, and a novel abstracted graph representation based on the simplified skeleton is presented in Chapter 3. This ligature analysis is computed based on a new measure of saliency based on uniqueness of inscribed medial disks combined with limiting average outward flux value. Finally, contributions made by this thesis and future research directions are reviewed and discussed in Chapter 4.

CHAPTER 2
Flux Based Skeletons

This chapter begins by reviewing the computation of the skeleton using the notion of average outward flux of the gradient of the Euclidean distance function through a boundary [34], [19], [18], [17]. In order to be self contained we provide necessary mathematical notation and definitions in Section 2.1. We then review the approach of flux based skeletonization proposed by Dimitov *et al.* [17] in Section 2.2. Section 2.3 then discusses boundary representation through flux based skeletons.

The original boundary reconstruction approach reviewed in Subsection 2.3.1 uses only regular points (ignoring junction- and end points) to reconstruct the original object's boundary. The method is imperfect and also has the limitation of being sensitive to local noise. To prove that flux-based skeleton is a faithful representation, equivalent to the original object, we present a more precise method to compute a more comprehensive boundary through the skeleton where the mentioned limitations are addressed in Section 2.3.2.

Finally, in Section 2.4 we present a new method used to optimize the distance function calculation on a rectangular lattice so as to make the original implementation by Dimitrov *et al.* [17] more practical, and then to further study its use in boundary reconstruction.

2.1 Definitions and Properties

This section gives the basic definitions and properties which are necessary to understand the rest of the chapter. The reader familiar with these concepts can move directly to section 2.1.6. In this section, we shall give precise definitions for shape, distance transform and, in the process, we shall also introduce the vast majority of concepts needed throughout this thesis.

2.1.1 Shape

A 2-D *shape* Ω is a set of geometrical loci that the projection of an object occupies in 2-D space. Its boundary, $\partial\Omega$, consists of a finite number of mutually disjoint closed curves, each being a connected path in this space that does not intersect itself.

2.1.2 Euclidean Structure

We shall start by introducing the notation that will be used in Euclidean space. In mathematics, Euclidean distance represents the common distance between two points that one could measure with a ruler. The *Euclidean metric* $d(P, Q) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is:

$$d(P, Q) = ||P - Q||$$

A *transformation* is a mathematical function that maps a set of algebraic elements to another set, $f : X \to Y$, where f is the transformation function, and X and Y can be members of any spaces such as Euclidean space. Rotations, translations and reflections include some examples of transformations.

A transformation is called rigid if the distance between every pair of points remains the same. A *rigid transformation* can be proved to be a rotation, a translation, or a combination of the two. Let $Q \in \mathbb{R}^n$ be a rigid transformation of $P \in \mathbb{R}^n$, then there must be a rotation matrix, $R \in \mathbb{R}^n \times \mathbb{R}^n$, and a translation vector, $T \in \mathbb{R}^n$, for which $Q = RP + T$. Note that $R^T = R^{-1}$ (i.e., R is an orthogonal matrix). Shape, size and distances remain the same after a rigid transformation. Two shapes represent the same object in Euclidean space if one of them can be transformed to the other one by a rigid transformation.

2.1.3 Distance transform

For each point P, and a given shape Ω, a distance metric, $d_\Omega(P)$, can be defined as follows:

$$d_\Omega(P) = \inf_{Q \in \partial\Omega} d(P, Q).$$

The *distance transform* of a shape Ω is a signed distance function that specifies how close a given point P is to the boundary of that shape $\partial\Omega$:

$$\mathcal{D}_\Omega(P) = \begin{cases} d_\Omega(P) & \text{if } P \text{ is inside } \Omega \\ 0 & \text{if } P \in \partial\Omega \\ -d_\Omega(P) & \text{if } P \text{ is outside of } \Omega \end{cases}$$

Assume that on the boundary $\partial\Omega$, there exists only one point Q of minimum distance to P ($\mathit{\Pi}_\Omega(P) = \{Q\}$). We would then define the *distance function gradient vector* for point P as:

$$\dot{\mathbf{q}}_\Omega(P) = \frac{Q - P}{||Q - P||}$$

In the case of $|\mathit{\Pi}_\Omega(P)| > 1$, one can not define the closest boundary point uniquely, and therefore the distance function gradient vector can not be defined for point P. As a result, the distance function gradient vector $\dot{\mathbf{q}} = \nabla\mathcal{D}$ is not defined for skeletal points, because skeletal points have more than one closest boundary points according to the "Maxwell set" definition of medial axis [28]. Except for at skeletal points, $\dot{\mathbf{q}}$ is continuous almost everywhere on its domain and it satisfies the eikonal equation:

$$|\dot{\mathbf{q}}| = 1.$$

Figure 2–1 shows an example of the distance function gradient vector.

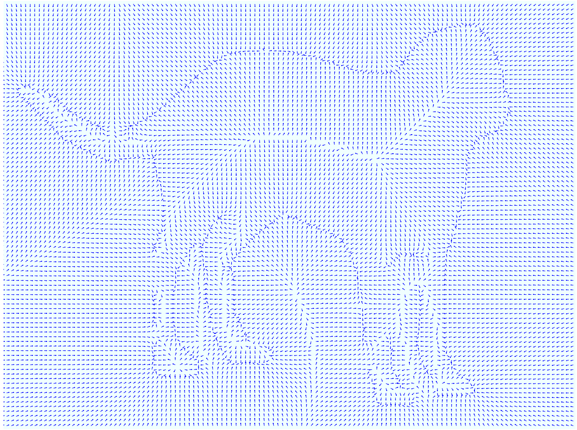


Figure 2–1: A distance function gradient vector field computed based on the boundary of a dog shape.

2.1.4 The Divergence Theorem

Strictly speaking, the divergence theorem relates the integral of the divergence of a vector field within a simply-connected region to the outward flux of that vector field through this boundary.

Assume R is a region in $\mathbb{R}^n$ with boundary ∂R, and $\mathbf{F}$ is a vector field differentiable on a neighborhood of R, then the standard form of the divergence theorem states that

$$\int_R (\nabla . \mathbf{F}) dv = \oint_{\partial R} \langle \mathbf{F}, \mathbf{N} \rangle ds. \tag{2.1}$$

2.1.5 An Extension of the Divergence Theorem

Dimitrov *et al.* [19] extended the divergence theorem to regions that include parts of a skeletal curve. Assume $\mathbf{F} = \dot{\mathbf{q}}$ is the distance function gradient vector field applied to a shape Ω, and R is an arbitrary region from the shape that includes a part of the skeletal curve C (see Figure 2–2). They showed that the following equation applies

$$\int_R \operatorname{div}(\dot{\mathbf{q}})dv = \int_{\partial R} \langle \dot{\mathbf{q}}, \mathbf{N} \rangle ds + 2 \int_C \langle \dot{\mathbf{q}}, \mathbf{N} \rangle ds. \tag{2.2}$$

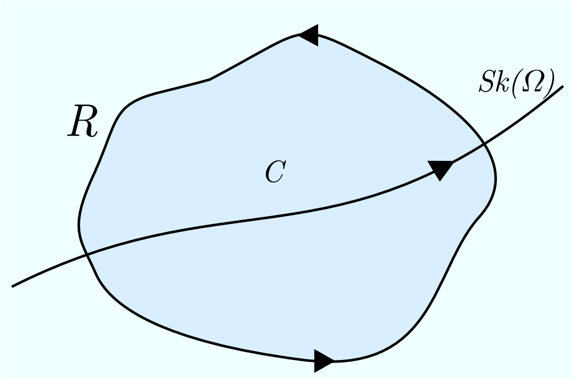


Figure 2–2: An arbitrary region R including a branch segment of the skeleton $Sk(\Omega)$ (adapted from [17]).

This equation will be used later in 2.1.7.

2.1.6 The Outward Flux of a Vector Field

As has been discussed previously, except for at skeletal points, $\dot{\mathbf{q}} = \nabla \mathcal{D}$ is a continuous and differentiable vector field on the shape domain; denoted as R. The *outward flux of* $\dot{\mathbf{q}}$ *through* ∂R is defined as

$$\int_{\partial R} \langle \dot{\mathbf{q}}, \mathbf{N} \rangle ds \tag{2.3}$$

and the *average outward flux of* $\dot{\mathbf{q}}$ *through* ∂R is defined as

$$\mathcal{AOF} = \frac{\int_{\partial R} \langle \dot{\mathbf{q}}, \mathbf{N} \rangle ds}{\int_{\partial R} ds}. \tag{2.4}$$

2.1.7 Intrinsic Meaning of the Outward Flux

Take a shape Ω, and a bounded circular region with radius $\epsilon \to 0$ around a point P for the divergence theorem, and let $\mathcal{F}_\epsilon(P)$ represent the outward flux vector field through that circular area. Then,

$$\mathcal{F}_\epsilon(P) = \int_R \operatorname{div}(\dot{\mathbf{q}}) dv = \int_0^L \langle \dot{\mathbf{q}}_P + \delta(s), \mathbf{N} \rangle ds, \tag{2.5}$$

where $\delta(s)$ is the quantity function value which has been added to the center of the gradient distance map vector field value at point P to get the value of $\dot{\mathbf{q}}$ at neighboring points on C, and L is the length of the boundary of the neighboring circular area. Dimitrov *et al.* [19] proved that when P does not lie on a skeletal branch,

$$\lim_{L \to 0} \left(\int_0^L \langle \dot{\mathbf{q}}_P, \mathbf{N} \rangle ds + \int_0^L \langle \delta(s), \mathbf{N} \rangle ds \right) \to 0 \tag{2.6}$$

and thus

$$\lim_{L \to 0} \frac{\left(\int_0^L \langle \dot{\mathbf{q}}_P, \mathbf{N} \rangle ds + \int_0^L \langle \delta(s), \mathbf{N} \rangle ds \right)}{\int_{\partial R} ds} \to 0 \tag{2.7}$$

which means that the right-hand side in equation 2.5, and also its division by $\int_{\partial R} ds$ (the average outward flux ($\mathcal{AOF}$) value) goes to zero as $\epsilon \to 0$.

When P lies on a skeletal branch, equation 2.2 yields

$$\int_{\partial R} \langle \dot{\mathbf{q}}, \mathbf{N} \rangle ds = \int_R \mathrm{div}(\dot{\mathbf{q}}) dv - 2 \int_C \langle \dot{\mathbf{q}}, \mathbf{N}_C \rangle ds \tag{2.8}$$

where C is the part of the skeleton limited in R. Since $R \to 0$, It can be easily verified that the first integral on the right hand side goes to zero in the limit. Using the proved property they showed that

$$\int_{\partial R} \langle \dot{\mathbf{q}}, \mathbf{N} \rangle ds = -2 \int_C \langle \dot{\mathbf{q}}_P + \delta(s), \mathbf{N} \rangle ds = -2 \times (\inf_C \langle \dot{\mathbf{q}}_C, \mathbf{N}_C \rangle) \times \mathrm{length}(C) \tag{2.9}$$

Since $R \to 0$ all elements located inside R will vanish therefore $C \to 0$ and:

$$\lim_{\epsilon \to 0} \mathcal{F}_\epsilon(P) = \int_{\partial R} \langle \dot{\mathbf{q}}, \mathbf{N} \rangle ds = -2 \times (\inf_C \langle \dot{\mathbf{q}}_C, \mathbf{N}_C \rangle) \times \mathrm{length}(C) = 0 \tag{2.10}$$

and

$$\begin{aligned} \mathcal{AOF} &= \frac{\int_{\partial R} \langle \dot{\mathbf{q}}, \mathbf{N} \rangle ds}{\int_{\partial R} ds} && (2.11) \\ &= \frac{-2 \times (\inf_C \langle \dot{\mathbf{q}}_C, \mathbf{N}_C \rangle) \times \mathrm{length}(C)}{\mathrm{length}(C)} && (2.12) \\ &= -2 \times (\inf_C \langle \dot{\mathbf{q}}_C, \mathbf{N}_C \rangle). && (2.13) \end{aligned}$$

When the region shrinks to a skeletal point P,

$$\mathcal{AOF} = -2 \times \langle \dot{\mathbf{q}}_P, \mathbf{N}_P \rangle. \tag{2.14}$$

Table 2–1 summarizes the results obtained by Dimitrov *et al.* [17] which have been reviewed in this subsection.

	$P \in Sk(\Omega)$	$P \notin Sk(\Omega)$
$\lim_{\epsilon\to 0} \mathcal{F}_\epsilon(P)$	0	0
$\mathcal{AOF}$	$-2 \times \langle \dot{\mathbf{q}}_P, \mathbf{N}_P \rangle$	0

Table 2–1: This table summarizes the results of the limiting behaviour of the outward flux and $\mathcal{AOF}$ of the gradient of the Euclidean distance function to the boundary of a 2D object. (Results are adapted from [19].)

2.2 Flux Based Skeletons

Based on the above results Dimitrov *et al.* [17] developed a flux based medial axis computation method. The key idea behind this approach is to examine different situations that can happen at a skeletal point. In the remainder of this section, we will review properties that they derived and the overall method.

2.2.1 Limiting to Circular Neighborhoods

According to the results of table 2–1, the $\mathcal{AOF}$ value of all points not located on the skeleton is equal to zero. Therefore, the skeletonization algorithm will look for those points that have a non zero value of the $\mathcal{AOF}$. There are three different classes of skeletal points that altogether make for the skeleton of a given shape. Dimitrov *et al.* [17] reviewed the properties of each of class by analyzing the behavior of the $\mathcal{AOF}$, when shrunk to a circular neighborhood. Assuming a shape Ω and a skeletal point $\mathbf{p}$, their results are as follows:

1. $\mathbf{p}$ is a *regular point* if the maximal inscribed disc at $\mathbf{p}$ touches the boundary at two corresponding boundary $|\Pi_\Omega(P)| = 2$. The computed $\mathcal{AOF}$ at a regular point $\mathbf{p}$ is given by

$$\lim_{\epsilon\to 0} \frac{\mathcal{F}_\epsilon(\mathbf{p})}{2\pi\epsilon} = -\frac{2}{\pi} \sin\theta. \tag{2.15}$$

See Figure 2–3a.

2. **p** is an *end point* if these exists δ $(0 < \delta < r)$ such that for any ϵ $(0 < \epsilon < \delta)$ the circle centered at **p** with radius ϵ intersects $Sk(\Omega)$ just at a single point (r is the radius of the maximal inscribed disc at **p**). The computed $\mathcal{AOF}$ at an end point **p** is given by

$$\lim_{\epsilon \to 0} \frac{\mathcal{F}_\epsilon(P)}{2\pi\epsilon} = -\frac{1}{\pi}(\sin\theta_P - \theta_P). \tag{2.16}$$

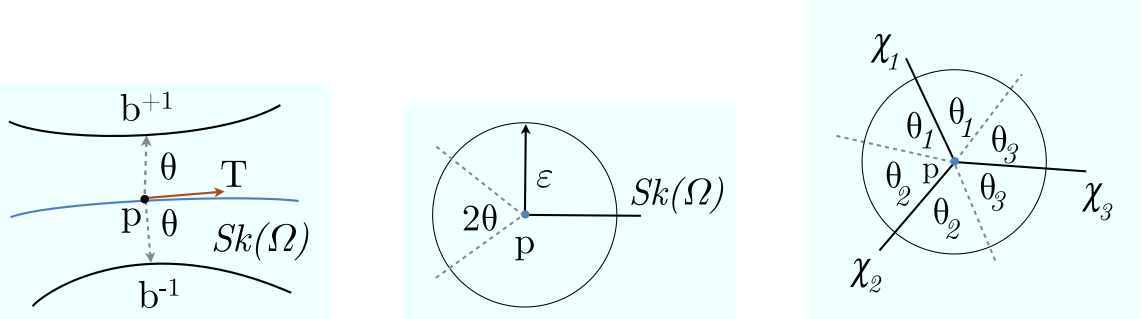
   See Figure 2–3b.

3. **p** is a *junction point* if $\Pi_\Omega(\mathbf{p})$ has three or more corresponding closest boundary points. Generically a junction point has degree 3. All other branch points are unstable. The computed $\mathcal{AOF}$ at a junction point **p** is given by

$$\lim_{\epsilon \to 0} \frac{\mathcal{F}_\epsilon(P)}{2\pi\epsilon} = -\frac{1}{\pi}\sum_{i=1}^{n} \sin\theta_i. \tag{2.17}$$

   See Figure 2–3c.

These different classes of skeletal points are shown in Figure 2–3.



(a) A regular point

(b) An end point

(c) A junction point

Figure 2–3: LEFT: Different types of a skeletal point shown separately on an arbitrary segment of the skeleton $Sk(\Omega)$ of a given shape Ω (adapted from [17]).

2.2.2 Computational Setup of Flux Based Skeletonization

To compute the skeleton, we need to calculate the limited average outward flux value for all points on the shape space, and find those with non-zero $\mathcal{AOF}$ values. To achieve this goal, we should first compute $\mathcal{AOF}$ for a circular area of radius r around every point P

$$\mathcal{F}_r(P) = \frac{\int \langle \dot{\mathbf{q}}_{\widetilde{P}}, \mathbf{N}(s) \rangle ds}{2\pi r}, \tag{2.18}$$

where $\widetilde{P} = \{P + r\mathbf{N}(s)\}$, and $\dot{\mathbf{q}}$ is the distance function at point $\widetilde{P}$. By discretizing the circular area boundary into n equal arcs, the numerator is approximated by:

$$\int \langle \dot{\mathbf{q}}_{\widetilde{P}}, \mathbf{N}(s) \rangle ds = \frac{2\pi r}{n} \sum_{k=0}^{n-1} \langle \dot{\mathbf{q}}_{\widetilde{P}}, \mathbf{N}(k) \rangle, \tag{2.19}$$

where ds is approximated by division of the perimeter $2\pi r$ by n $(\frac{2\pi r}{n})$, and $\mathbf{N}(k) = (\cos(\frac{2\pi k}{n}), \sin(\frac{2\pi k}{n}))$.

2.3 Boundary Reconstruction

According to the Maxwell set definition of the medial axis, each point on the skeleton has 2 or more corresponding boundary points. Therefore, given a mapping between boundary points to skeletal points, it is possible to invert that mapping to reconstruct the boundary purely from skeletal points and their properties. Dimitrov *et al.* [17] attempted to do this by exploiting the relationship between regular points of the medial axis and the object angle. In this section we will review the basic algorithm for doing this and then extend it to obtain a more complete boundary reconstruction by properly considering the case of endpoints and junction points (in addition to regular points). Throughout the remainder of this subsection, we let $\mathbf{p}$ represent a skeletal point on the parametrized skeleton $Sk(\Omega)$ of the given object Ω, which is characterized by the inscribed maximal disk radius r, the object angle at that point θ, and the unit tangent vector $\mathbf{t}_\mathbf{p}$ to the skeleton at $\mathbf{p}$ (see Figure 2–3a). The proceeding subsections propose methods attempting to compute corresponding boundary points for $\mathbf{p}$.

2.3.1 Boundary Representation through Regular Points with First-order Approximation of the Tangent Vector

Dimitrov *et al.* [17] studied the boundary reconstruction idea by computing the inverted map from the regular points to their corresponding boundary points, considering that almost all skeletal points are regular except the small number of end points and junction points at the ends of skeletal branches. Therefore, taking a regular point namely $\mathbf{p}$, they outlined the reverse transform from the skeletal points to corresponding boundary points by

$$R_{1,2} = \mathbf{p} + r\mathrm{Rot}(\pm\theta)\mathbf{t}_\mathbf{p}, \tag{2.20}$$

where $R_{1,2}$ represent the two closest corresponding points. To reconstruct $R_{1,2}$ from a regular point on a parameterized skeleton using 2.20, the following parameters of a skeletal point ought to be numerically computed: the coordinates of the point $\mathbf{p}$, the radius value r, the object angle θ, and the unit tangent vector $\mathbf{t}_\mathbf{p}$. During the skeletonization process, a parameterized skeleton is computed where each of its skeletal points include the position $\mathbf{p}$ the radius at that point r, and the limiting average outward flux value $\mathcal{AOF}$. For the missing numerical value of object angle, they proposed a numerical estimation for θ calculated based on equation 2.15 :

$$\theta = \arcsin\left(-\frac{\mathcal{F}_\epsilon(P)}{4\epsilon}\right).$$

They also estimated the tangent vector by the slope of the line that connects the prior skeletal point $\mathbf{p}_{-1}$ to posterior skeletal point $\mathbf{p}_{+1}$ (see Figure 2–4)

$$\mathbf{t_p} = \frac{\mathbf{p}_{+1} - \mathbf{p}_{-1}}{\|\mathbf{p}_{+1} - \mathbf{p}_{-1}\|}.$$

Figure 2–5 provides some results from their skeletonization and boundary reconstruction algorithms.

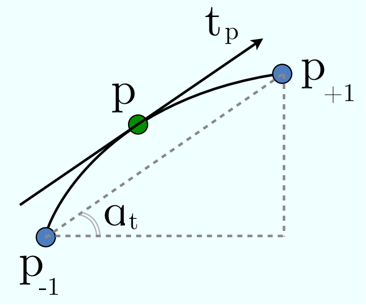


Figure 2–4: Estimation of the tangent vector at a skeletal point is shown using the prior and the posterior skeletal points.

2.3.2 Full Boundary Reconstruction

As can be seen in the lowest row of shapes in Figure 2–5, the boundary representation through regular points reviewed in 2.3.1 does not provide a full boundary reconstruction and is incomplete. In this subsection, we extend the previous boundary reconstruction approach to a more general one that includes all types of skeletal points and gives a better numerical approximation of the parameters required for reconstruction.

To achieve this aim, three limitations from the boundary reconstruction method developed by Dimitrov *et al.* [17] are considered and addressed in our approach:

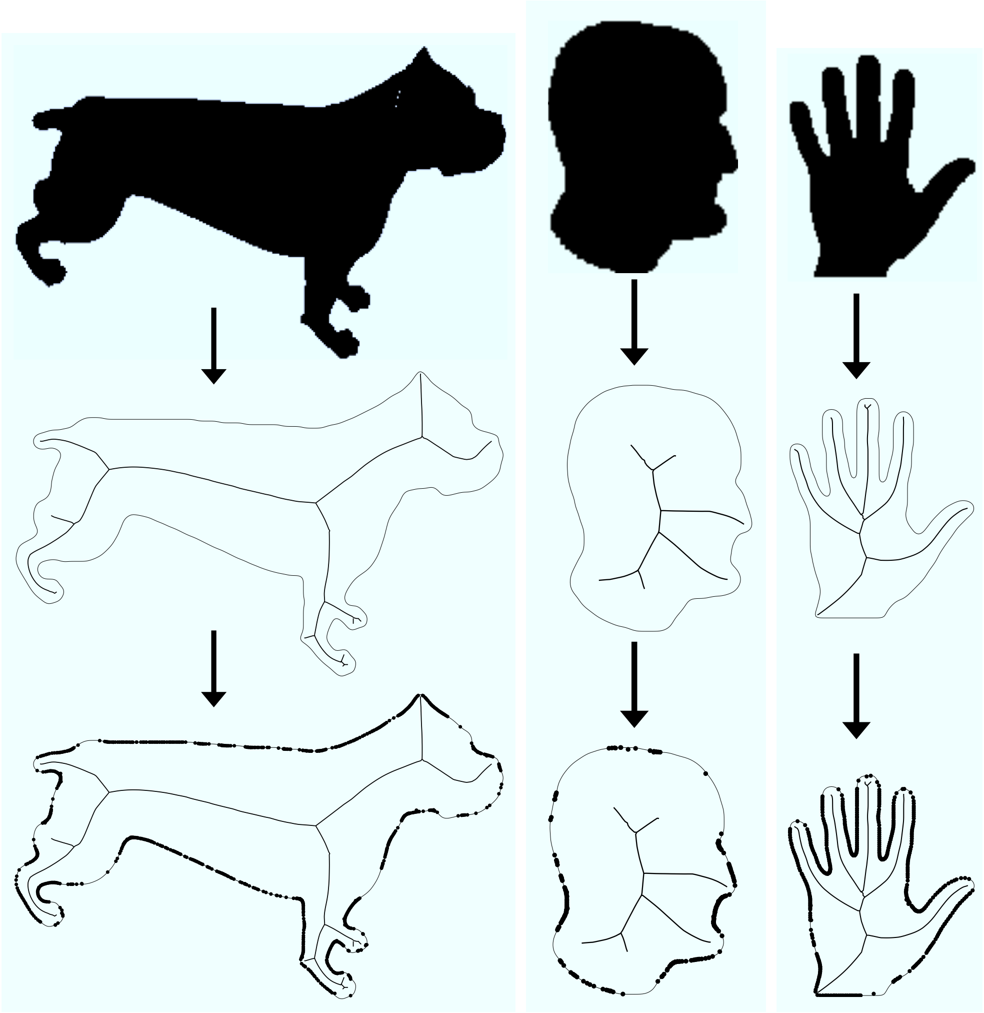


Figure 2–5: The top level figures are binary images of some given examples. Their derived skeletons are shown in the middle level using flux based skeletonization. The bottom level figures provide the reconstruction results using the method of Dimitrov *et al.* [17]. The small black disks represent reconstructed boundary points.

1. **Sensitivity of first-order approximation of tangent estimation**: The two point stencil computation of tangent is very sensitive to discretization effects along the skeleton, and results show that the estimation fails for many regular points. To avoid local numerical errors, we deploy higher order methods

for approximating the unit tangent. For those medial loci for which the two points method fails, we use four points stencil approximation [1] given by

$$\mathbf{t}_{\mathbf{p}} = \frac{2}{3}\left(\frac{\mathbf{p}_{+1} - \mathbf{p}_{-1}}{\|\mathbf{p}_{+1} - \mathbf{p}_{-1}\|}\right) + \frac{1}{3}\left(\frac{P_{+2} - P_{-2}}{\|P_{+2} - P_{-2}\|}\right) \tag{2.21}$$

where $\mathbf{p}_{+2}$ and $\mathbf{p}_{-2}$ represent the subsequent and the prior skeletal points to $\mathbf{p}_{+1}$ and $\mathbf{p}_{-1}$ respectively.

Using the second-order of approximation of tangent estimation results in a number of newly reconstructed boundary points (see Figure 2–6).

2. **Boundary points that map to an end point**: The boundary reconstruction method by Dimitrov *et al.* [19] does not explicitly consider the other two types of skeletal points (end points and junction points). This decision results in a number of circular segments missing from the boundary which map to the end points. We present a numerical approach to recover those missing boundary points. Assume $\mathbf{p}$ is an end point such as the one shown in figure 2–3b. Then, there would be a circular arc segment from the boundary corresponding to this skeletal point. The osculating disk at $\mathbf{p}$ touches the boundary along that circular segment and the tangent vector to the skeleton at that point bisects the angle that sees the circular arc. Let γ represent the curve of that circular arc segment, then

$$\gamma : I \rightarrow \Omega \tag{2.22}$$

$$\gamma(\theta) = \mathbf{p} + r\mathrm{Rot}(\theta)\,\mathbf{t}_{\mathbf{p}} \tag{2.23}$$

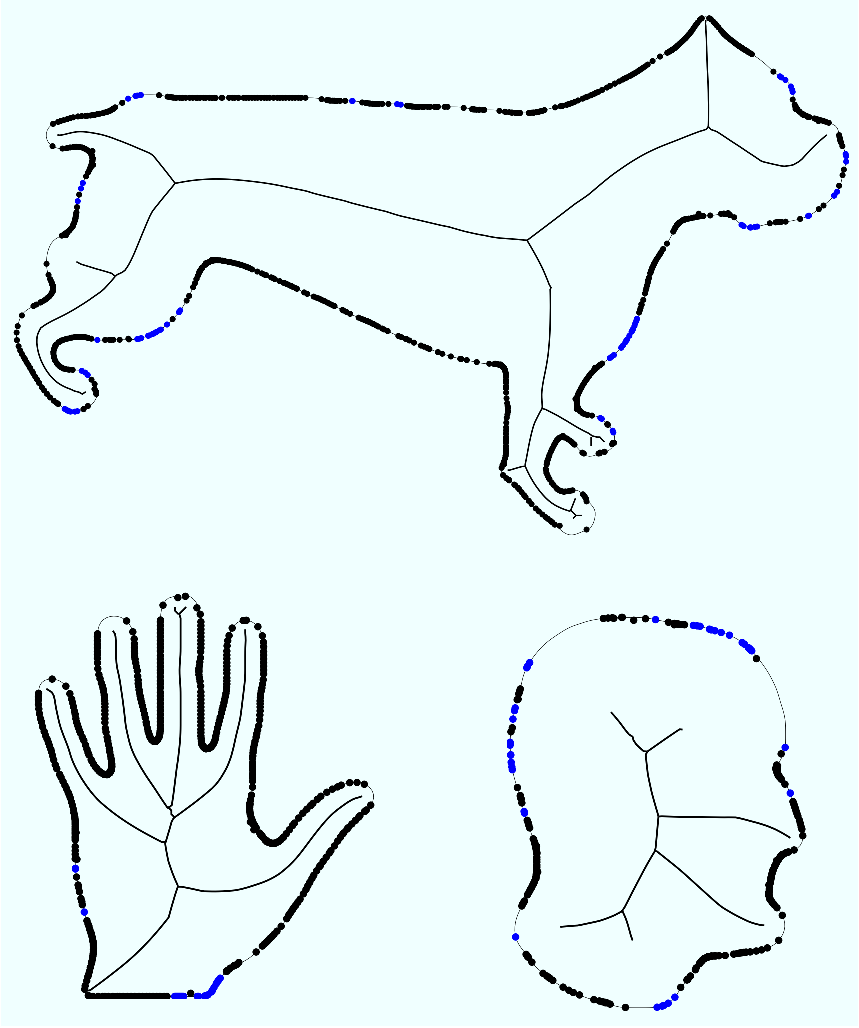


Figure 2–6: Along with initial reconstructed points (results shown in 2–5) shown with black disks, newly reconstructed points resulting from the improvement of tangent estimation are shown with blue disks.

where I is a closed bounded interval $[-\theta_{\mathbf{p}}, \theta_{\mathbf{p}}]$. The coordinates of the point $\mathbf{p}$, and the radius value r are parameters that are computed during the skeletonization process. To compute γ, the following parameters are needed to be computed numerically other than $\mathbf{p}$, and r: the object angle $\theta_{\mathbf{p}}$, and the unit tangent vector $\mathbf{t_p}$. To compute the object angle, we use the equation 2.16

$$\frac{F_\epsilon(P)}{2\pi\epsilon} = -\frac{1}{\pi}(\sin\theta_p - \theta_p) \Rightarrow \tag{2.24}$$

$$-\frac{F_\epsilon(P)}{2\epsilon} = \sin\theta_p - \theta_p. \tag{2.25}$$

It can be easily verified that the function

$$f(x) = x - \sin(x) - C$$

where C is an arbitrary constant real number, has only one root. Therefore, if we replace C by $\frac{F_\epsilon(P)}{2\epsilon}$, then the zero of the function $f(x)$ becomes a numerical estimation for the object angle at the end point $\mathbf{p}$. For the tangent vector $\mathbf{t}_P$, we simply use the tangent estimation of the prior point to the end point $\mathbf{t_p} = \mathbf{t_{p_{-1}}}$. Figure 2–7 shows some examples of boundary reconstruction with results of the newly found boundary circular segments corresponding to end points.

3. **Boundary points that map to a junction point**: Junction points are also not included in the initial boundary reconstruction method by Dimitrov *et al.* [19]. We compute the corresponding boundary points of a junction point the same way that we compute the corresponding boundary points of a regular point with the difference that the tangent vectors near junction points are approximated by their prior skeletal points on the skeleton. The rest of the procedure is the same as that for computing boundary points for a regular point $\mathbf{t_p} = \mathbf{t_{p_{-1}}}$. Figure 2–8 shows the improvement on reconstructed points with results of newly found boundary points corresponding to junction points.

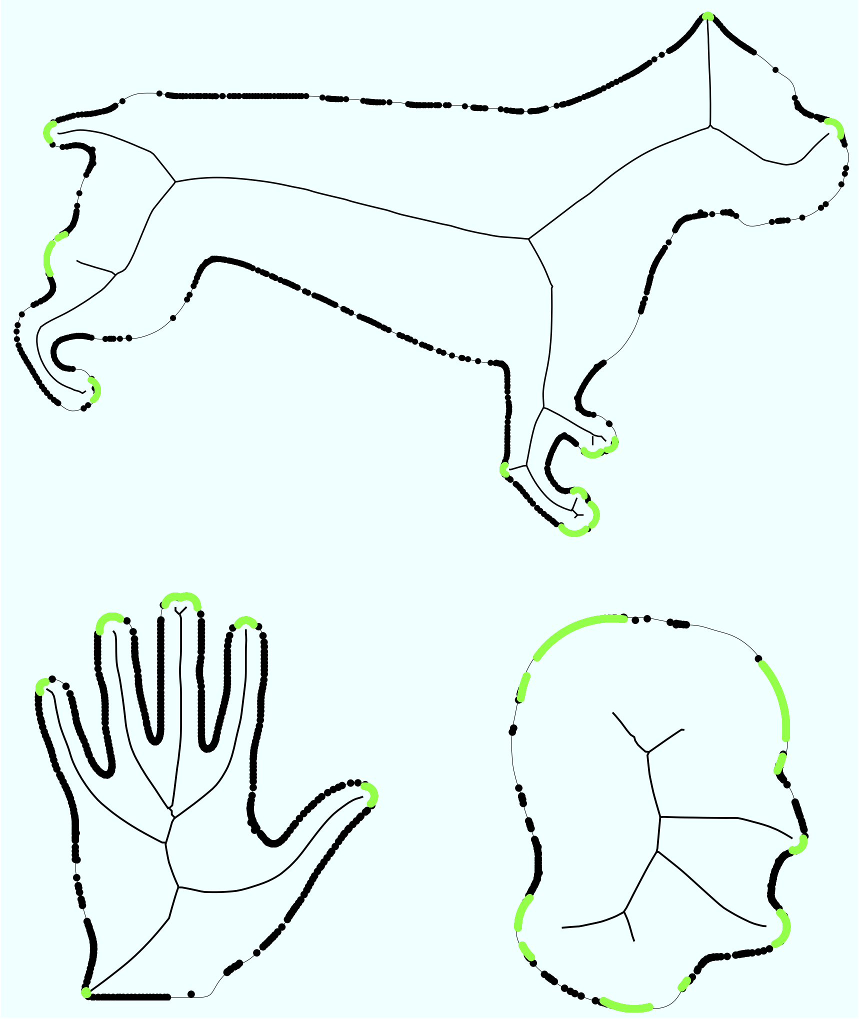


Figure 2–7: Along with initial reconstructed points (results shown in 2–5) shown with black disks, newly reconstructed boundary circular segments corresponding to end points are shown with green disks.

Figure 2–8 also shows that there are not that many points reconstructed from junction points. This is understandable since there are not usually many junction point and furthermore, unlike end points, these points can reconstruct at most two boundary points.

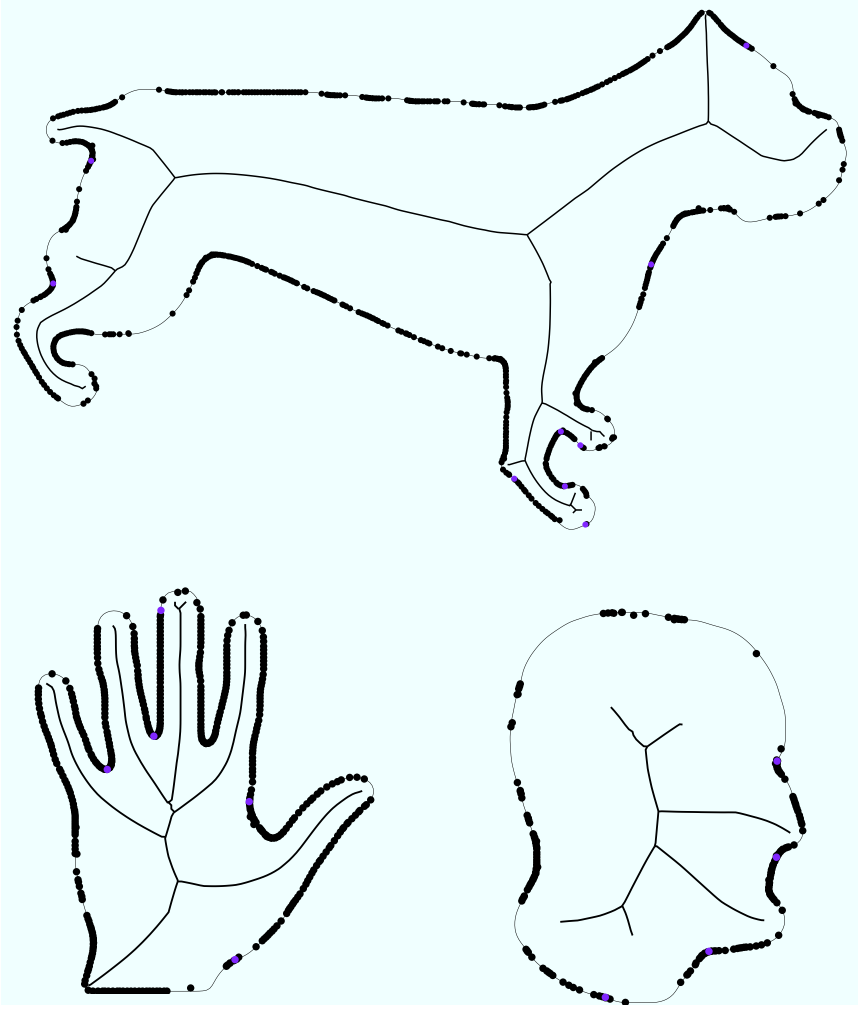


Figure 2–8: Along with initial reconstructed points (results shown in 2–5) shown with black disks, newly reconstructed boundary points corresponding to junction points are shown with violet disks.

The contribution of our approach in reconstructing boundary points is threefold: improved of approximation of tangents for many regular points of the skeleton, the computing of circular segments that correspond to end points of the skeleton, and the computing of extra boundary points from junction points. In figure 2–9 the contribution that this thesis has made to the boundary construction method that Dimitrov *et al.* proposed in [19] is shown.

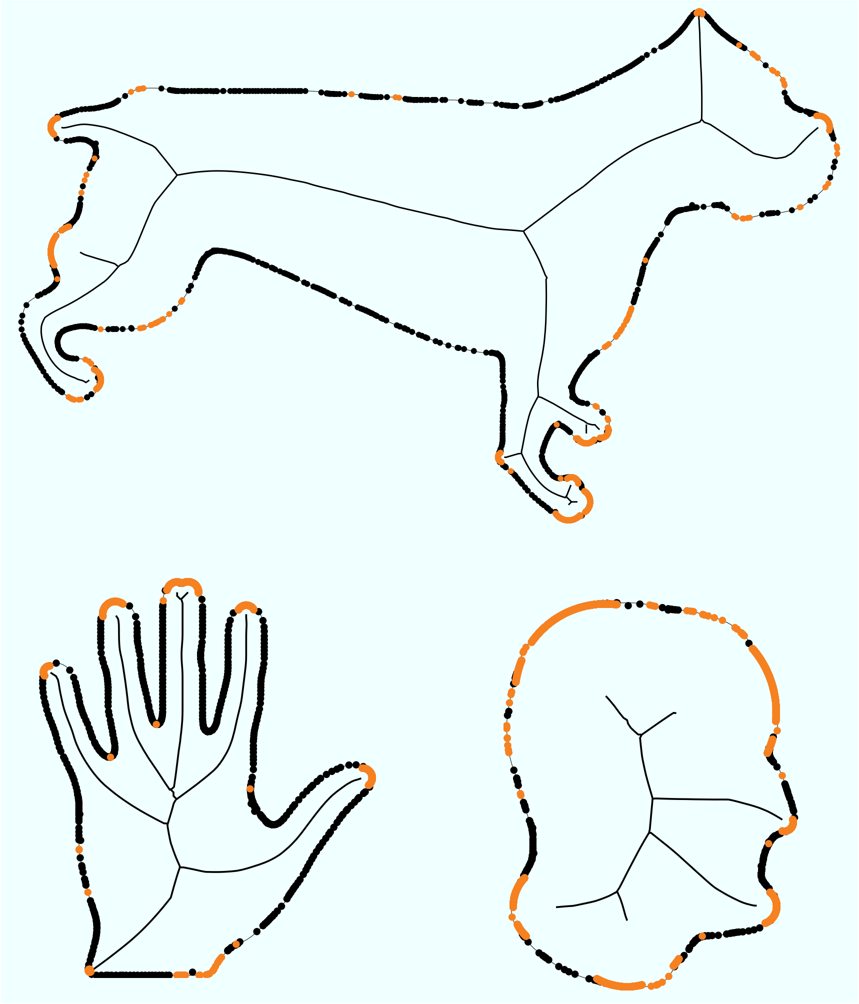


Figure 2–9: Along with point reconstructed using Dimitrov algorithm [19], all reconstructed boundary points contributed by this thesis are shown in orange. The fraction of the total number of (discretized) boundary points reconstructed for each of these examples is presented in table 2–2.

Example	Old Reconstruction	New Reconstruction
Dog	88.11 %	98.82 %
Head	66.85 %	98.74 %
Hand	88.36 %	97.79 %

Table 2–2: This table presents the results of the improvement made by our boundary reconstruction process for the three examples used by this thesis: the dog, the head, and the hand. REMARK: These results are representative but not exact, because the boundary reconstruction process depends on the sampling of (discrete) points on the medial axis

2.4 Optimization of Distance Function Computation

In order to compute the flux value in flux-based skeletonization, Dimitrov *et al.* [17] used a brute-force algorithm to compute a distance transform, $\mathcal{D}_\Omega(P)$, for a given shape Ω. For each point P of the shape Ω, the brute-force algorithm computes $d_\Omega(P)$ by calculating distances from P to all boundary points $\partial\Omega$, and then choosing the distance to the point on the boundary with minimal distance value. This procedure is a time-consuming and exhaustive one, and therefore is extremely slow when dealing with high-resolution input shapes. Using a brute-force approach is not feasible for methods that deal with large databases of objects, and also is not pragmatic to use for online tasks that need flux-based representation.

One of the contributions of this thesis is an optimization of this process. A general idea to improve the efficiency of the brute-force distance function computation is to decrease the number of calculations of distances. In this section, we introduce a new method in which instead of computing distance from a considered point to each boundary point, the calculation is restricted to fewer boundary points.

Our optimized method consists of five mains steps:

1. Using a particular splitting factor k, the bounding box that contains the shape is split into a square grid with a certain width for squares (e.g., see Figure 2–10) .

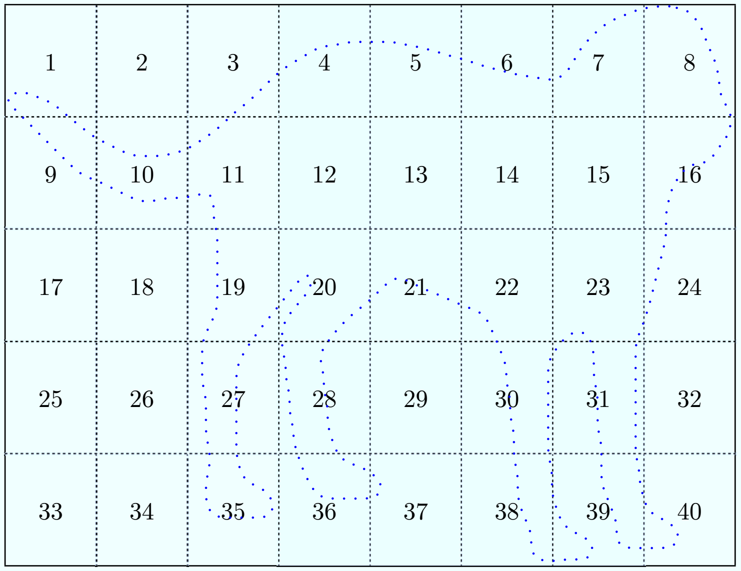


Figure 2–10: The grid for distance function optimization

2. An unordered map (hash map) is made from those grid cells that contain at least one boundary point to indexes of all boundary points located in the corresponding grid cell, denoted by $\mathcal{H}$.
3. For each index of a grid cell, all closest grid cells that may have the closest boundary point are computed, and stored in another map denoted by $\mathcal{I}$.
4. For each point P, using $\mathcal{I}$ and the index of the grid cell that contains P, all closest grid cells to P that have at least one boundary are computed.

5. Using $\mathcal{H}$, all boundary points from the closest grid cells found in the previous step are considered and their distances to P are calculated and the minimum is picked as the answer to $d_\Omega(P)$.

According to our method, Steps 1, 2, 3 are only carried out once, and this contributes to decrease the total number of operations dramatically in real applications.

2.4.1 Time Complexity

Based on the choice of a splitting factor, the number that the width and height of the original bounding box of the shape is divided by, and resolution of discretization of the shape and its boundary, the time for computing a distance function may vary a lot. We did an experiment on a database of 1664 given shapes of 13 objects with both the brute-force distance function approach and our optimized distance function on the same machine, and the results of average computation time for computing all skeletons of each object class are reported in Table 2–3.

	Brute-force	Our Method			
		$k = 4$	$k = 8$	$k = 12$	$k = 16$
Alien	26.88	10.96	7.41	3.83	5.46
Bull	32.11	14.26	8.38	4.82	6.53
Camel	29.31	11.24	7.30	4.70	5.52
Child	28.99	12.10	7.93	4.63	5.65
Dinasour	33.72	14.81	8.62	4.89	6.45
Dog	30.32	12.23	7.71	5.01	6.34
Dolphin	26.30	11.65	6.27	4.38	5.46
Eagle	22.06	9.51	5.83	3.11	4.36
Guitar	23.08	9.43	6.18	3.68	4.72
Horse	32.78	13.20	8.51	4.88	6.27
Kangoro	33.64	14.75	8.32	5.24	7.04
Knife	21.12	9.25	5.80	3.17	4.17
Pig	28.90	12.50	7.29	4.54	5.42
Average	28.40	11.99	7.35	4.37	5.64

Table 2–3: The above table provides the average run time of our method with different splitting factors in comparison with the brute-force algorithm. Average run time for computing a skeleton for each shape of a considered class is shown. The base unit of measurement for all numbers is seconds. All the experiments were performed on a MacBook pro with a 2.4 GHz Intel Core 2 Duo processor, 4 GB RAM, NVIDIA GeForce 320M 256 MB graphics card, and 250GB 5400 RPM SATA disk. For these measurements, we used the OS X Version 10.8.

CHAPTER 3
Flux Graphs for 2D Shape Analysis

Abstracting a graph from a medial axis and then using graph matching methods is a popular approach in shape matching. Extracting a graph from the skeleton of a given shape is followed by defining a set of nodes for disjointed parts of the object and a set of edges for their connectivities, where the abstraction is a reliable representation of the silhouette appearance. The objective of this chapter is to introduce a novel way for simplifying skeletons using salience measures for skeletal branches, and then extract a graph based on the simplified skeletal segments. To this end, we present a new method for simplifying a skeleton where the representation based on the incorporated parts preserves the majority of the shape's area.

More specifically, the contributions of this chapter are three-fold: The first contribution is to use the uniqueness of an inscribed disk as a tool for defining salience, combined with the limiting average outward flux value. This measure of salience is new and experiments show that it leads to an intuitive and simplified skeletal segments with no significant loss in boundary representation. Second, we present a method to abstract the simplified skeletal segments into a graph, which we call the "flux graph". We show via a number of experiments that this is much less complex than earlier approaches such as shock graphs and skeletal graphs. Finally, we demonstrate the potential of using flux graphs for shape matching.

We shall start in Section 3.1 by proposing a new approach for simplifying a medial axis using a new measure of salience based on uniqueness of disks combined with the limiting average outward flux value. We shall then introduce flux graphs based on the simplified skeleton in Section 3.2. Applicability of flux graphs for matching problems is investigated in Section 3.3. Finally, we evaluate our proposed method of abstraction of the simplified medial axis in Section 3.4.

3.1 Local Geometry of Salient Parts of the Medial Axis

To find salient parts, one must know about the properties of the local geometry of a skeletal point on the derived skeleton of a given shape. Initial properties of a skeletal point and some measures of saliency for the skeleton was reviewed in chapter 1. In chapter 2, we explained that the flux based skeletonization idea developed by Dimitrov *et al.* is characterized by the limiting average outward flux value $\mathcal{AOF}$, and then we verified that this characterization faithfully represents the object.

Numerically, the stable computation of the $\mathcal{AOF}$ and its proportionality the object angle directed us to use this measure combined with a new measure based on uniqueness of disks to compute a saliency range for skeletal points of a given shape. In this section, we present these two measures of saliency.

3.1.1 Uniqueness of Disks

In this subsection, we present a new saliency measure that has been used in our simplification and is called “Uniqueness of Disks”.

Definition 3.1.1. Assume a skeletal point on the skeletal branch of a given shape. The maximal inscribed disk of that skeletal point is called a *Unique Disk* if and only if it does not intersect with any other maximal inscribed disks of skeletal points from other branches. The center of a unique disk is called a *Unique Skeletal Point*.

As an example consider figure 3–1. Let $\mathbf{p}.\upsilon$ represents the uniqueness of a skeletal point $\mathbf{p}$, so that

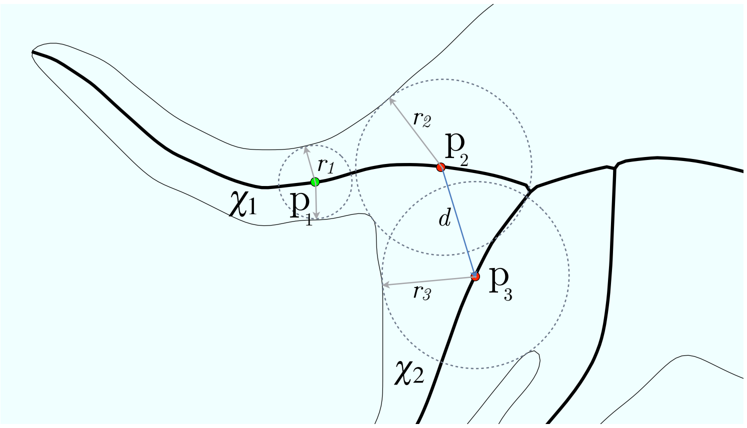


Figure 3–1: Uniqueness of Disks: Part of a dog shape is cropped and shown with illustrated disks corresponding to unique and non-unique skeletal points. The maximal inscribed disk centered at $\mathbf{p}_1(\in \chi_1)$ is a unique disk and $\mathbf{p}_1$ is unique skeletal point because it does not intersect with any other maximal inscribed disks from other branches than χ_1. At the same time, $\mathbf{p}_2(\in \chi_1)$ is not a unique skeletal point because its maximal inscribed disk centered at $\mathbf{p}_2(\in \chi_1)$ intersects with $\mathbf{p}_3(\in \chi_2)$, a maximal inscribed disk from χ_2.

$$\mathbf{p}.\upsilon = \begin{cases} 1 & \text{if } \mathbf{p} \text{ is a unique skeletal point.} \\ \\ 0 & \text{if } \mathbf{p} \text{ is not a unique skeletal point.} \end{cases}$$

The pseudo-code of the approach for finding the unique skeletal points is outlined in Algorithm 1.

Eliminating a unique skeletal point results in a loss part of a shape boundary because the corresponding boundary segment can not be reconstructed by any other

Algorithm 1 Computing Local Geometry of Salient Parts of the Medial Axis

```
Declaration of variables
$\Omega$: shape image
$Sk(\Omega) = \{\chi_1, ..., \chi_n\}$: Skeleton of $\Omega$ consisted of a set of branches
$Sk_\nu(\Omega)$: Simplified skeleton
$\chi_i = \{\mathbf{p}_{i1}, ..., \mathbf{p}_{il}\}$: A skeletal branch consisted of a set of skeletal points
$\mathbf{p}_j = (x, y, r, f, \upsilon)$: A skeletal point is characterized by its position $(x, y)$, radius $r$,
the flux value $f$ of a skeletal point, and the uniquenss of that skeletal point $u$.

procedure Compute Uniqueness
for $\forall \chi_i \in Sk(\Omega)$ do
  for $\forall \mathbf{p}_{ij} \in \chi_i$ do
    if $\exists\ \mathbf{p}_{kl} \in \chi_k \mid d(\mathbf{p}_{kl}, \mathbf{p}_{ij}) < \mathbf{p}_{kl}.r + \mathbf{p}_{ij}.r$ then
      $\mathbf{p}_{ij}.\upsilon \leftarrow 0$ {Set $\mathbf{p}_{ij}$ as non-unique skeletal point}
    else
      $\mathbf{p}_{ij}.\upsilon \leftarrow 1$ {Set $\mathbf{p}_{ij}$ as unique skeletal point}
    end if
  end for
end for
end of procedure

procedure Compute Normalized Flux
All flux values are
for $\forall \chi_i \in Sk(\Omega)$ do
  for $\forall \mathbf{p}_{ij} \in \chi_i$ do
    $\mathbf{p}_{ij}.f \leftarrow \frac{\mathcal{F}_\epsilon(\mathbf{p}_{ij})}{\max\{\mathcal{F}_\epsilon\}}$ $\{\lim \frac{\mathcal{F}_\epsilon(P)}{2\pi\epsilon} = -\frac{2}{\pi}\sin\alpha \rightarrow \max\{|\mathcal{F}_\epsilon|\} = \frac{4}{\epsilon}\ \}$
  end for
end for
end of procedure

procedure Simplifying the Skeleton
$Sk_\nu(\Omega) = \emptyset$
for $\forall \chi_i \in Sk(\Omega)$ do
  for $\forall \mathbf{p}_{ij} \in \chi_i$ do
    $\tau \leftarrow$ a certain threshold
    if $\max(\mathbf{p}_{ij}.\upsilon, \mathbf{p}_{ij}.f) > \tau$ then
      $Sk_\nu(\Omega) = Sk_\nu(\Omega) \bigcup \mathbf{p}_{ij}$

    end if
  end for
end for
end of procedure
```

skeletal points from other branches. In figure 3–2, parts of the skeleton of a dog example found to satisfy the uniqueness of disks criterion are shown.

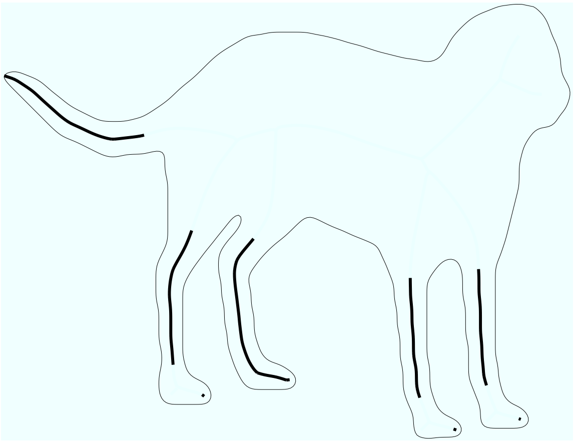


Figure 3–2: The skeletal points found to be unique are shown in black on the medial axis of a dog example.

3.1.2 Limited Average Outward Flux

As we reviewed flux based skeletons in chapter 2, except at junctions and end points, the flux value of the rest of the skeleton is computed by:

$$\mathcal{AOF} = \lim_{\epsilon \to 0} \frac{\mathcal{F}_{\epsilon}(\mathbf{p})}{2\pi\epsilon} = -\frac{2}{\pi} \sin \alpha \tag{3.1}$$

for a considered regular point $\mathbf{p}$. The equation determines a relation between the flux value and the object angle. The bigger the flux value, the higher the object angle gets. The higher the object angle at a portion of a skeleton the more likely the shape silhouette is to be elongated locally at that part. Elongated parts admit a simple and stable medial axis structure. Skeletal points with high flux value along a medial branch are likely to be appended in the set of candidate points for the simplified skeleton.

In figure 3–3, skeletal points of a dog example are colored form white to black (normalized between [0 1]) based on the normalized flux values which are in accordance to the object angle at these points.

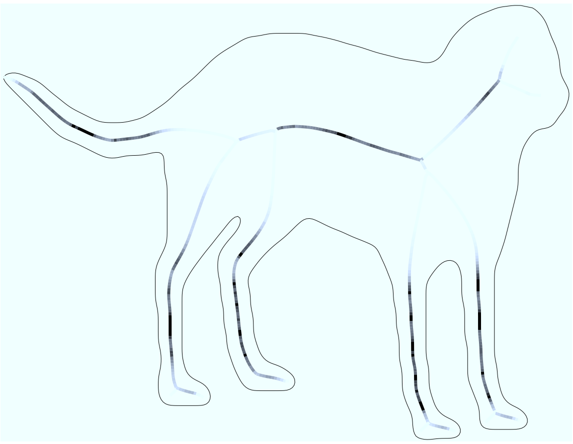


Figure 3–3: Normalized flux values of a skeleton are shown in a range starting from white, which has the least flux, and ending in black which has the maximum flux on a dog example.

In Algorithm 1 the procedure of setting flux values is outlined.

3.1.3 Simplifying the Skeleton

According to measures of saliency introduced in previous section (uniqueness of disks and flux values), we simplify the flux based skeletons. To decide whether a skeletal point should be kept or thrown away, we apply the following procedure: When the considered skeletal point is unique or its normalized flux value is greater than a certain threshold, then that skeletal point will be kept in the skeleton; otherwise it will be thrown away. In our experiments, we assume $\tau = 0.9045$ which means all non-unique skeletal points with object angle α greater than about 60° will be retained in the simplified skeleton. The procedure is coded in Algorithm 1. Figure

3–4 provides results of applying the simplification algorithm with a certain threshold on the dog example.

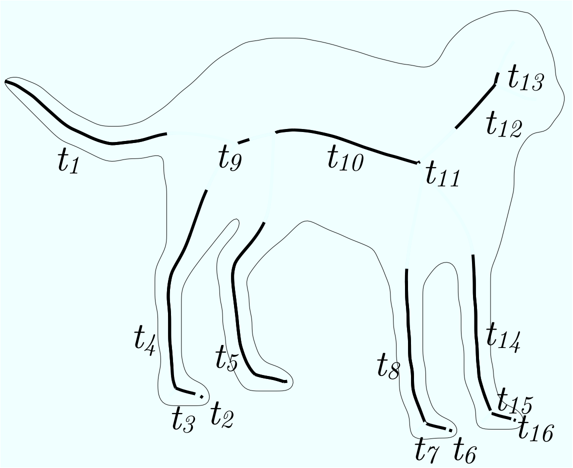


Figure 3–4: Several salient segments labeled as t_i are shown as the result of simplifying the medial axis of a dog example.

3.2 Flux Graphs

3.2.1 Introduction

The main reason for finding the local geometry of salient parts and simplifying the flux-based skeleton is to extract a graph representation which is simpler than but otherwise as complete as effective as the popular existing approaches. In this section, we propose a graph called “Flux Graph” that describes a shape as a set of separate parts each connected to a number of other ones, while preserving the topology of the object being represented.

3.2.2 Nodes and Edges

Each graph consists of a set of nodes. According to the simplified skeleton described before, the sets of nodes and edges are made for flux graphs. Simplifying the skeleton results in a number of fragmented parts. Not all these fragments should be apart from each other, therefore those ones that share a big portion of their volumes and are close enough to each other, are combined through a merging process (see Algorithm 2). The remaining departed fragments at the end of the merging process are treated as the nodes of a flux graph. This merging process is applied to get a more robust graph which is not sensitive to local fragmentation of nodes.

A critical condition that may occur is when a skeletal branch does not generate any salient skeletal points and the entire branch is thrown away during the simplification. Ignoring a complete skeletal branch changes the topology of the skeleton. On the other hand, playing with the threshold on the object angle to get some skeletal points is also not a good idea, because short segments of skeletal points with low salience have very low representational influence on the original shape's boundary. These considerations lead us to take the entire skeletal branch as a node for the flux graph. The results of merging fragmented parts on the dog shape are shown in figure 3–5.

The set of links between nodes are determined based on their connectivities on the original medial axis. This set of links is assumed to represent the set of edges in the flux graph. To direct edges, we considered the average radii of inscribing disks along two adjacent nodes and compared them. The one with magnitude is chosen as the parent and the other one as the child.

3.2.3 Equivalency of the Graph to the Main Object

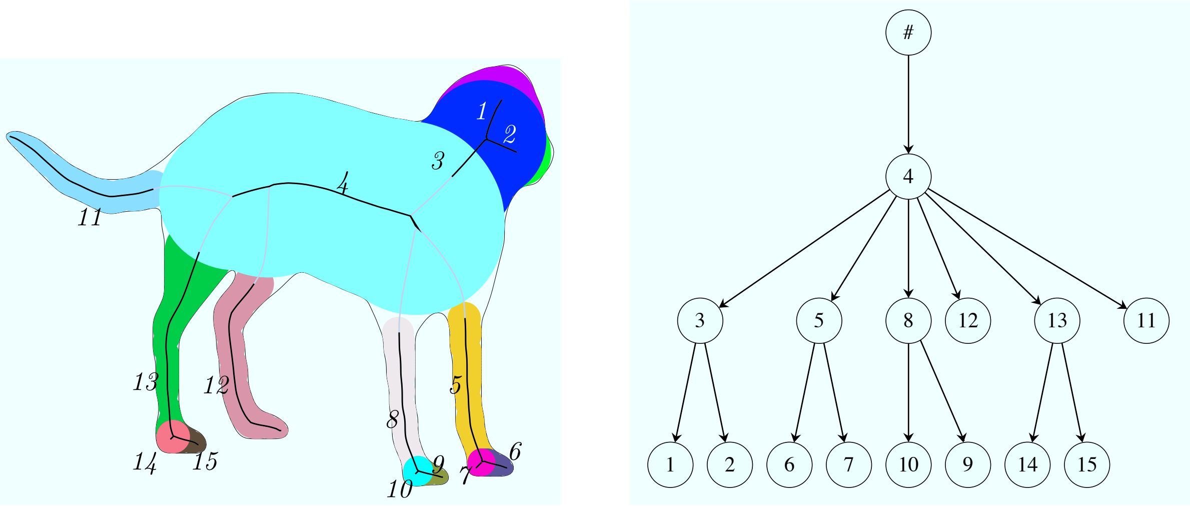
In this subsection we will discuss the equivalency of a flux graph to the original shape that it is representing. To study how accurately a flux graph represents its

Algorithm 2 Flux Graph Abstraction Using a Simplified Skeleton

```
Declaration of variables
$FG = (V, E)$: Set of nodes and set of edges of a Flux Graph
$Sk_\nu(\Omega)$: Set of fragmented parts of the simplified skeleton
$L_\nu(\Omega)$: Set of links between fragmented parts in the simplified skeleton
$vol(t_i)$: Volumetric area reconstructed by the set $t_i$

procedure Compute $L_\nu(\Omega)$
$\emptyset \leftarrow L_\nu(\Omega)$
for $\forall t_i \in Sk_\nu(\Omega) \mid t_i \in \chi_m$ do
  for $\forall t_j \in Sk_\nu(\Omega) \mid i \neq j$ and $t_j \in \chi_n$ do
    if $\nexists t_k \in Sk_\nu(\Omega) \mid t_k \in \{\chi_m \bigcup \chi_n\}$ and $(d(t_k, t_i) + d(t_k, t_j)) \leq d(t_i, t_j)$ then
      $L_\nu(\Omega) = L_\nu(\Omega) \bigcup (t_i, t_j)$
    end if
  end for
end for
end of procedure

procedure Merging Process - Compute $V$ and $E$
Compute $L_\nu(\Omega)$
for $\forall t_i \in Sk_\nu(\Omega)$ do
  for $\forall t_j \in Sk_\nu(\Omega) \mid i \neq j$ do
    $v \leftarrow |(vol(t_i) \bigcap vol(t_j))|$ {Compute the shared area of reconstruction of disks for $t_i$ and $t_j$}
    if $v > 0.95|vol(t_i)|$ or $v > 0.95|vol(t_j)|$ then
      if $(t_i, t_j) \in L_\nu(\Omega)$ then
        $t_{ij} = t_i \bigcup t_j$ {$t_i$ and $t_j$ will be merged. All skeletal points in between will be added to the new merged part}
        for $\forall l \in L_\nu(\Omega) \mid t_i \in l$ or $t_j \in l$ do
          replace $t_i$ (or $t_j$) with $t_{ij}$
        end for
        delete $t_i, t_j$ from $Sk_\nu(\Omega)$ and add $t_{ij}$ to $Sk_\nu(\Omega)$
      end if
    end if
  end for
end for
$(V, E) = (Sk_\nu(\Omega), L_\nu(\Omega))$
end of procedure
```



(a) (b)

Figure 3–5: The finalized flux graph of a dog example is shown. LEFT: The set of nodes is shown with the recovering parts depicted in different colors. RIGHT: The extracted directed acyclic graph corresponding to the flux graph of the dog example is drawn. There is a dummy node called $\sharp$ that is father to every nodes that do not have any fathers.

original shape, different parameters can be take into account. Some of them are explained in bellow:

1. Percentage of reconstructed shape: The percentage of the volume that a flux graph represents from the original shape is a favorable candidate to measure the accuracy of a skeletal graph. The higher this ratio is, the more capable the simplified skeleton is of representing the original object. Fortunately, flux graphs get excellent scores of equivalency to the main object. Experimental results shown in table 3–2 provide evidence for this claim (this is discussed in more detail in Section 3.4).
2. Hausdorff distance: This is a function to compute a distance measure between two subsets of a metric space, and is often used to measure how far two shapes are from each other. Assume a given shape X, and its reconstructed shape from its simplified flux graph Y. Taking the boundary of X as ∂X and of Y as ∂Y, the Hausdorff distance between them is defined as follows:

$$d_H(\partial X, \partial Y) = \max\{\sup_{x \in \partial X} \inf_{y \in \partial Y} d(x,y), \sup_{y \in \partial Y} \inf_{x \in \partial X} d(x,y)\}. \tag{3.2}$$

3.3 Graph Matching for Flux Graphs

3.3.1 Introduction

A skeletal graph abstraction can be used as a tool in many visual shape problems; one of them is the matching task. To be able to compare shapes, graphs are derived from their silhouettes and are matched with each other in a trial step. In this section, we explain how flux graphs are matched and compared against each other. To carry out a comparative experiment with shock graphs we used the same graph matching set up and database used for shock graphs [25], [36] as our matching framework in this thesis.

3.3.2 Topological Similarity

The topological structure similarity is a measure that says how much the derived graphs are similar to each other. Given two flux graphs, a bipartite graph is made between nodes of their DAGs. Each edge is weighted based on the structural similarity between nodes; the weight is the normalized length of difference of their topological signature vectors

$$w = \frac{|\mathbf{t}_1 - \mathbf{t}_2|}{\max(|\mathbf{t}_1|, |\mathbf{t}_2|)}$$

where $\mathbf{t}_1$ and $\mathbf{t}_2$ are representing corresponding TSV vectors of the two nodes of a considered edge. The best matching of a maximum weighted bipartite matching is when the sum of the values of the edges is maximized. The idea of using the TSV is adapted from [36]. Like shock graphs, flux graphs can be represented using a $\{0, 1\}$ adjacency matrix, with 1's indicating adjacent nodes in the tree form of the DAG. The eigenvalue-value sum of a sub-DAG of a given flux graph with computed eigenvalues of its corresponding submatrix is invariant to any similarity transformation applied to the submatrix. In a DAG representation, the TSV is defined as the vector of eigenvalue-sums derived from the corresponding adjacency matrix for the sub-DAG of the considered node. The matching algorithm used is a greedy algorithm [25] which has the benefit of finding a largest maximal matching in polynomial time. The similarity is computed by matching a query with a model node and then normalizing by the number of matched nodes according to the order of the model graph.

3.3.3 Geometrical Similarity

The geometric information investigates the relation of the local geometry of skeletal points on the nodes of each given DAG. To determine the similarity between nodes, Macrini *et al.* [25] tried to fit line segments to the skeletal points of a given node. Therefore, for a given query and a given model, the algorithm tries to fit the query to the model by allowing line segments of the model to shrink or grow to include the query data points. The main assumption here is that the derivative of the radius function can be related to the object angle which has to do with the relationship between the bi-tangent points associated with a skeletal point. For small amounts of foreshortening due to modest rotations in depth, the change in object angle will be small.

3.3.4 The DAG Matcher

Since each flux graph is a directed acyclic graph (DAG), to match a query shape with other shapes, we need to develop a DAG matcher. The DAG matcher receives two DAGs as input and computes a value representing their similarity, as well as a list of corresponding nodes in the two DAGs. This analysis considers both topological structure ($\mathit{\Gamma}$) and geometric information ($\mathit{\Delta}$) associated with a flux graph's vertices. Each of these two measures return a value normalized in the interval [0 1]. The final similarity score is a weighted combination of these two

$$S(G_1, G2) = \omega \mathit{\Gamma}(G_1, G_2) + (1 - \omega)\mathit{\Delta}(G_1, G_2),$$

where $S(G_1, G2)$ represent the similarity between DAGs derived from two given shapes, and ω is a tunning weight in the interval [0 1]. At the end of the process a list of corresponding nodes and a similarity measure are made.

3.4 Experimental Results

In this section, we present our experimental results using flux graphs together with the shape representation and graph matching tasks. The representation task experiment includes results regarding equivalency to the main object discussed in Subsection 3.2.3, and the matching problem experiment goal is to recognize unseen 2-D query views of 3-D objects by matching the query view against all the available data modes (reviewed in Section 3.3). We compare results of these experiments with those obtained using shock graphs [27], [23].

3.4.1 The Dataset

The dataset used for experiments is the same dataset used for experiments done for Bone Graphs [24], [26] and Shock Graphs [25]. The data set has 13 3-D graphics models used for matching. Perspective projection of the 3-D object is computed onto the image plane where each model is centered in a uniformly tessellated view sphere. With 128 uniformly sampled views per object, the data set contains a total of 1664 2-D projected views.

3.4.2 Complexity Comparison with the Shock Graph

In this chapter, we suggested a method to abstract a representation based on the proposed simplified skeleton. One of the contributions this thesis has made, is this new category of skeletal graphs that are more simple and almost as powerful as existing popular approaches. To evaluate this claim, we carried out a complexity

Graph	Shock Graph					
	Nodes	Edges	$\Sigma \deg(v) d$	Depth	Skeletal points	TSV
Alien	27.63	29.47	49.16	4.72	453.2969	58.17
Bull	35.12	37.83	77.09	6.14	604	74.62
Camel	32.33	34.54	73.00	6.52	593.1562	67.56
Child	10.58	10.44	13.06	3.58	431.1953	20.47
Dinasour	26.50	27.85	53.45	5.55	433.6328	56.78
Dog	23.66	24.89	43.16	4.94	441.5938	47.92
Dolphin	20.84	21.34	42.84	5.25	362.8047	44.63
Eagle	14.55	14.55	21.68	4.13	411.1562	28.41
Guitar	25.45	27.14	44.70	4.86	320.4219	52.38
Horse	17.66	17.99	31.08	4.73	501.9844	38.49
Kangoro	28.59	31.02	52.85	5.24	524.3359	58.28
Knife	35.65	38.99	64.84	5.41	274.4531	70.56
Pig	27.48	28.54	58.65	5.63	537.8516	59.28

Table 3–1: The above table provides the average values of some complexity measures for shock graphs. Results are generated based on 1664 2-D view-based shapes of 13 3-D objects.

comparison experiment against Shock Graphs, one of the most known applied skeletal graphs. Our attempts to quantitatively evaluate the complexity of a flux graph have been related to the measures as follows: the count of graph vertices, the count of graph edges, the cumulative sum of number of nodes at each depth multiplied by the depth, the depth, total number of skeletal points on the graph, and average of TSV (topological signature vectors) values. Quantitative results of computing these measures based on the data set used for this thesis are provided in tables 3–2, and 3–1. The efficiency percentage that flux graph contributes with respect to the shock graph is presented in table 3–3. This measure indicates how much more efficient the measure is for flux graphs over shock graphs, expressed as a percentage.

Graph	Flux Graph						
	Nodes	Edges	$\Sigma \deg(v)d$	Depth	Skeletal points	TSV	Coverage
Alien	15.55	14.70	27.48	4.19	378.7891	35.40	0.99
Bull	15.62	14.62	27.14	4.41	409.4297	35.63	0.98
Camel	11.47	10.48	17.22	3.88	459.4531	24.79	0.98
Child	6.58	5.58	5.35	2.42	340.4453	11.15	0.99
Dinosaur	12.95	11.96	20.88	4.03	327.0312	27.80	0.99
Dog	15.20	14.48	19.94	3.85	328.8672	29.89	0.98
Dolphin	5.52	4.55	4.40	2.42	294.5938	8.83	0.99
Eagle	14.92	13.99	25.05	4.22	339.4297	31.33	0.99
Guitar	12.62	11.62	19.41	3.95	272.3047	27.28	0.99
Horse	12.73	11.87	17.91	3.88	370.6484	27.58	0.99
Kangaroo	11.83	10.86	16.58	3.55	358.0625	24.00	0.99
Knife	14.60	13.77	27.05	4.38	229.8594	33.94	0.99
Pig	13.86	12.86	21.85	3.93	362.1328	31.18	0.99

Table 3–2: The above table provides the average values of some complexity measures for flux graphs introduced in this thesis. The final column on the right is the coverage percentage of the original object, i.e., the ratio of the area reconstructed by the flux graph to the area of the original shape. These results are generated based on 1664 2-D view-based outlines of 13 3-D objects [27], [23].

	Nodes	Edges	$\Sigma \deg(v)d$	Depth	Skeletal point	TSV	Coverage
Efficiency	%49.87	%56.08	%59.99	%26.38	%24.08	%48.52	%99

Table 3–3: Average efficiency percentage of flux graphs over to shock graphs. These measure are obtained by taking the ratios of the corresponding entries in Tables 3–1 and Table 3–2, subtracting those ratios from 1, and then averaging over all the objects in the database.

3.4.3 Matching 2-D Views of 3-D Models

In this subsection, the flux graph representations is evaluated against the shock graph in a set of view-based object recognition experiments. This comparison is followed by the same matching framework of [26]. The recognition task is followed by: (a) Each view is one by one removed from the database (1664 2-D view-based shapes), and compared to all other remaining views (b) if the class of closest the matching view is the same as that of the query, then we have a correct recognition. In the next sets of trials in each step, %25 of the total views are removed randomly from the database. The same experiment is carried out with further subsampled databases. Figure 3–6 plots the recognition estimation success rates for both shock graphs and flux graphs, averaged over all views of all objects in the database. See [27] for a more detailed explanation of the experimental set up.

Figure 3–7 provides some qualitative results for the discussed matching experiment. These typical examples show that the flux graph representation offers the advantages of efficiency in terms of fewer nodes, edges, depth levels and skeletal points than shock graphs and other skeletal graphs, while still allowing for intuitive part to part correspondences that appear to respect the hierarchy of parts in the object.

In terms of quantitative results, shock graphs outperform flux graphs slightly but this is because the matcher used in the experiment is tuned to shock graphs and their detailed features (see Figure 3–6). In this experiment, the DAG matcher used is not entirely appropriate for matching flux graphs because the geometry similarity

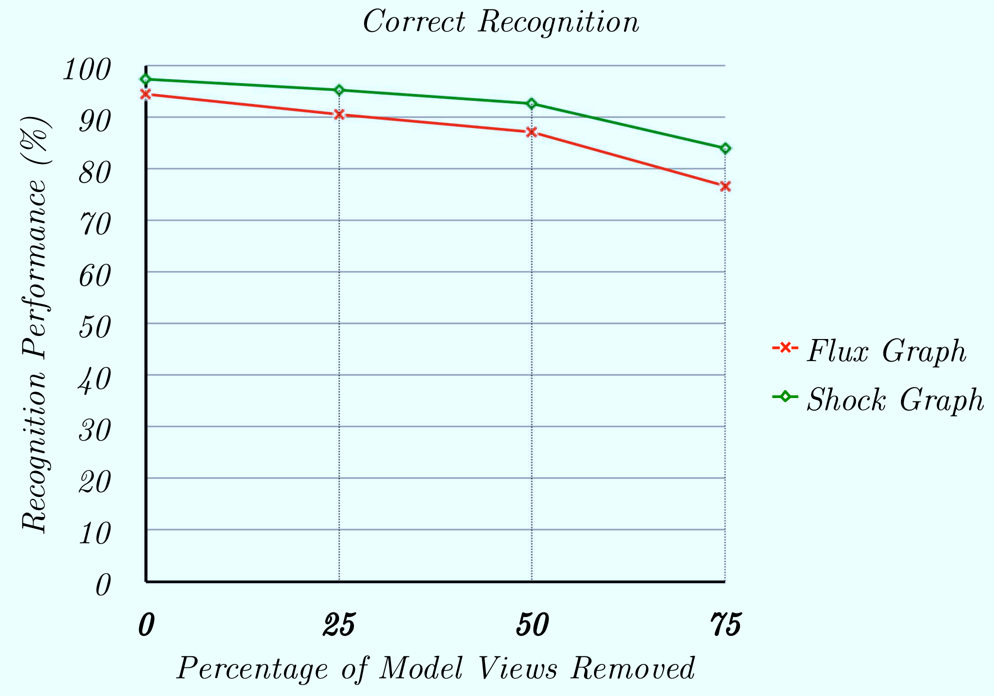
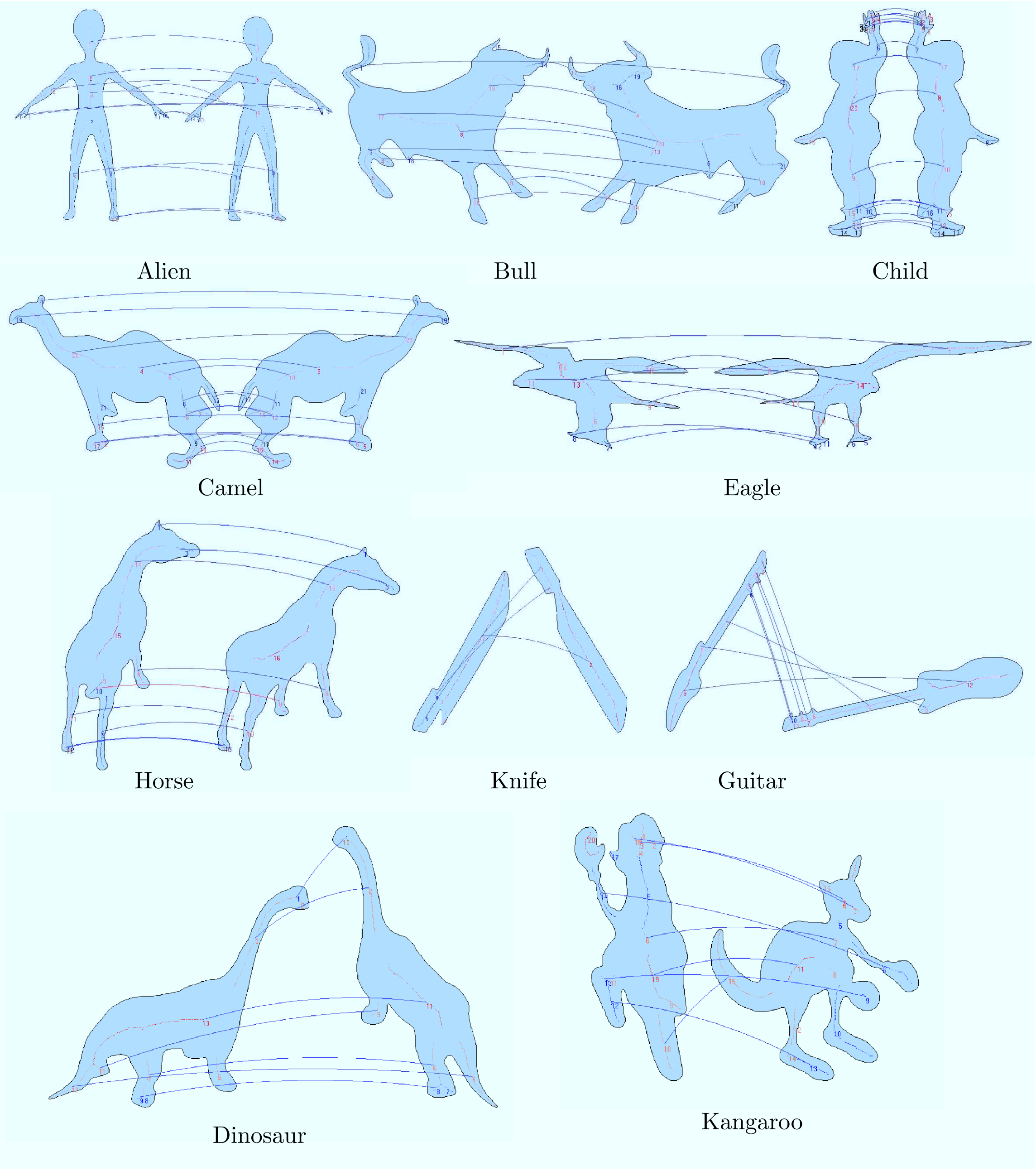


Figure 3–6

measure is not properly suited for matching nodes in flux graphs, and also the greedy matching algorithm does not exploit all hierarchical information.

In addition, presently there is a greater amount of commonality between flux graphs than shock graphs due to the removal of ligature-like regions. This makes matching flux graphs more tolerant (less sensitive) to small changes in viewpoint. This builds a case for their use in view-sphere partitioning but it may be that they are not yet suited for matching. More work needs to be done to investigate this issue.

In future work, one may conceive of developing a new node similarity measure based on other parameters that could be extracted from nodes of a flux graph. Furthermore, for the matching experiment, there is always the possibility of using various choices of graph matching algorithms based on the characterization of flux graphs.



Alien Bull Child

Camel Eagle

Horse Knife Guitar

Dinosaur Kangaroo

Figure 3–7: Some examples of node correspondences found by the DAG matcher.

CHAPTER 4
Conclusion

In this thesis, we extend the flux-based skeleton to achieve full boundary reconstruction in a manner that is efficient since the distance function computation is optimized. In addition, we present a a novel simplified shape representation that can be used to reconstruct almost all of the original object from a shape's medial axis.

We consider the flux-based medial axis approach developed by Dimitrov *et al.* [17], and we show that the method can be improved to perform skeletonization process a number of times faster than the original method. This contribution paves the way for tasks that apply the method on large databases of examples (e.g. the representation and matching experiments done in Chapter 3). A fast medial axis transform also benefits online algorithms by reducing the cost of computing a medial representation.

Another contribution of this thesis is its boundary reconstruction for flux-based skeletons. A comprehensive recovery of an object's boundary supports the integrity of a shape representation. In addition, a complete representation suggests a way of directly relating medial quantities to boundary features, because the medial features are easier to handle, to store and to compare with other represented objects than shapes boundaries directly.

Furthermore, the uniqueness of an inscribed disk is introduced as a new measure of saliency that evaluates the importance of a skeletal point using only medial information. This contributes to the task of ligature analysis where the goal is to remove portions of the medial axis that are unstable in the presence of slight boundary perturbations.

Using the uniqueness of the inscribed disk as a measure combined with the limiting average outward flux, a novel approach for flux graph representation is proposed, where skeletons of interest are simplified and abstracted as graphs that are much simpler than popular skeletal graphs in the literature.

In contrast with methods that carry out ligature analysis based on the limited number of configurations of the placement of ligature and non-ligature parts, our investigation has the advantage that the notion of saliency is defined for each skeletal point separately.

The representation has been evaluated by a matching framework designed for shock graphs ([23], [24]) to recognize 2D views of 3D objects and we have shown that flux graphs yield results that are almost as good as shock graphs.

4.1 Future Research Directions

To advance the representation proposed, one can incorporate some notions of types for nodes. These types can be used later in implementation of node similarity measures. Moreover, in order to efficiently utilize our representation for view-based shape matching and object recognition, it is necessary to address the problems of: (a) designing a proper geometry similarity measure, (b) applying a new matching algorithm that is in accordance with the analysis of shapes with flux graphs, (c) examining the use of our representation in more comprehensive problems of view-based pose estimation and object recognition.

The geometry similarity measure used in our framework to evaluate flux graphs is the same as that used to evaluate Shock Graphs in [25]. The similarity measure used is tuned in favor of Shock Graphs, which a new method of geometrical comparison between nodes of flux graphs.

Like the geometry similarity, the matching algorithm used in our evaluation, the greedy algorithm, is not the best choice. The greedy algorithm restricts the ability of the matcher to find the best matching parts from examples. The topological structure of a given shape can easily change and also may result in elimination or creation of some fragments in the presence of small variations or within-class deformation. Therefore, one potential improvement is to use a many-to-many correspondence matching technique for a given pair of medial axis graphs.

The other direction that this representation can be used for is to examine changes in pose estimation of 2D view based outlines of a 3D object. In summary, we believe that the established ability of the flux graphs can be further extended in problems of ligature detection, matching, and pose estimation with larger databases of examples, as well with a matching framework better suited for this kind of representation.

References

[1] Milton ABRAMOWITZ and Irene A. STEGUN. *Handbook of mathematical functions - with formulas, graphs, and mathematical tables.* Dover publications, New York, 1970.

[2] Nina Amenta and Marshall Bern. Surface reconstruction by voronoi filtering. In *Proceedings of the fourteenth annual symposium on Computational geometry*, SCG '98, pages 39–48, New York, NY, USA, 1998. ACM.

[3] J. August, A. Tannenbaum, and S.W. Zucker. On the evolution of the skeleton. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 315 –322 vol.1, 1999.

[4] Jonas August, Kaleem Siddiqi, and Steven W. Zucker. Ligature instabilities in the perceptual organization of shape. *Comput. Vis. Image Underst.*, 76(3):231–243, December 1999.

[5] Xiang Bai and Longin Jan Latecki. Path similarity skeleton graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1282–1292, July 2008.

[6] Xiang Bai, Xingwei Yang, Deguang Yu, and Longin Jan Latecki. Skeleton-based shape classification using path similarity. *IJPRAI*, 22(4):733–746, 2008.

[7] Harry Blum. A transformation for extracting new descriptors of shape. In Weiant Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.

[8] Harry Blum. Biological shape and visual science (part i). *Journal of Theoretical Biology*, 38(2):205 – 287, 1973.

[9] Harry Blum. Discussion paper: A geometry for biology. *Annals of the New York Academy of Sciences*, 231(1):19–30, 1974.

[10] Harry Blum and Roger N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10(3):167 – 180, 1978. The Proceedings of the IEEE Computer Society Conference.

[11] Jonathan W. Brandt and V. Ralph Algazi. Continuous skeleton computation by voronoi diagram. *CVGIP: Image Underst.*, 55(3):329–338, May 1991.

[12] Frédéric Chazal and André Lieutier. The "-medial axis". *Graphical Models*, 67(4):304 – 331, 2005.

[13] N.D. Cornea, M.F. Demirci, D. Silver, Shokoufandeh, S.J. Dickinson, and P.B. Kantor. 3d object retrieval using many-to-many matching of curve skeletons. In *Shape Modeling and Applications, 2005 International Conference*, pages 366 – 371, June 2005.

[14] M. Demirci, Ali Shokoufandeh, Yakov Keselman, Lars Bretzner, and Sven Dickinson. Object recognition as many-to-many feature matching. *International Journal of Computer Vision*, 69:203–222, 2006. 10.1007/s11263-006-6993-y.

[15] M.F. Demirci. Graph-based shape abstraction. In *Signal Processing and Communications Applications Conference, 2009. SIU 2009. IEEE 17th*, pages 13 –16, April 2009.

[16] M.F. Demirci, A. Shokoufandeh, and S.J. Dickinson. Skeletal shape abstraction from examples. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):944 –952, May 2009.

[17] 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.

[18] P. Dimitrov, C. Phillips, and K. Siddiqi. Robust and efficient skeletal graphs. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 417–423. IEEE, 2000.

[19] Pavel Dimitrov. Flux invariants for shape. Master's thesis, McGill University,, Montreal, July 2003.

[20] Peter Jonker and Adam Vossepoel. On skeletonization algorithms for 2, 3 .. n dimensional images. In *Shape, Structure and Pattern Recognition*, pages 71–80, 1995.

[21] Y. Keselman, A. Shokoufandeh, M.F. Demirci, and S. Dickinson. Many-to-many graph matching via metric embedding. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–850 – I–857 vol.1, June 2003.

[22] David G. Kirkpatrick. Efficient computation of continuous skeletons. In *Foundations of Computer Science, 1979., 20th Annual Symposium on*, pages 18 –27, oct. 1979.

[23] D. Macrini, S. Dickinson, D. Fleet, and K. Siddiqi. Object categorization using bone graphs. *Computer Vision and Image Understanding*, 115(8):1187–1206, 2011.

[24] D. Macrini, K. Siddiqi, and S. Dickinson. From skeletons to bone graphs: Medial abstraction for object recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, June 2008.

[25] D.A. Macrini. *Indexing and matching for view-based 3-d object recognition using shock graphs.* PhD thesis, Citeseer, 2003.

[26] Diego Macrini, Sven Dickinson, David Fleet, and Kaleem Siddiqi. Bone graphs: Medial shape parsing and abstraction. *Computer Vision and Image Understanding*, 115(7):1044 – 1061, June 2011.

[27] Diego A. Macrini. *Bone Graphs: Medial Abstraction for Shape Parsing and Object Recognition.* PhD thesis, University of Toronto, 2010.

[28] J.N. Mather. Distance from a submanifold in euclidean space. In *Proceedings of Symposia in Pure Mathematics*, volume 40, pages 199–216, 1983.

[29] 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, June 1992.

[30] R.L. Ogniewicz. Skeleton-space: a multiscale shape description combining region and boundary information. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 746 –751, June 1994.

[31] Robert L. Ogniewicz. *Discrete Voronoi skeletons.* Hartung-Gorre Verlag Konstanz, 1993.

[32] S. M. Pizer, W. R. Oliver, and S. H. Bloomberg. Hierarchical shape description via the multiresolution symmetric axis transform. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(4):505–511, April 1987.

[33] Jean Serra. *Image Analysis and Mathematical Morphology.* Academic Press, Inc., Orlando, FL, USA, 1983.

[34] K. Siddiqi, S. Bouix, A. Tannenbaum, and S.W. Zucker. Hamilton-jacobi skeletons. *International Journal of Computer Vision*, 48(3):215–231, 2002.

[35] K. Siddiqi and S. Pizer. *Medial representations: mathematics, algorithms and applications*, volume 37. Springer, 2008.

[36] Kaleem Siddiqi, Ali Shokoufandeh, Sven J. Dickinson, and Steven W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35:13–32, 1999. 10.1023/A:1008102926703.

[37] Gábor Székely. Shape characterization by local symmetries. *Medical Imaging*, pages 30–31, 1996.

[38] Yao Xu, Bo Wang, Wenyu Liu, and Xiang Bai. Skeleton graph matching based on critical points using path similarity. In Hongbin Zha, Rin-ichiro Taniguchi, and Stephen Maybank, editors, *Computer Vision – ACCV 2009*, volume 5996 of *Lecture Notes in Computer Science*, pages 456–465. Springer Berlin Heidelberg, 2010.

[39] Xingwei Yang, Xiang Bai, Deguang Yu, and Longin Jan Latecki. Shape classification based on skeleton path similarity. In *Proceedings of the 6th international conference on Energy minimization methods in computer vision and pattern recognition*, EMMCVPR'07, pages 375–386, Berlin, Heidelberg, 2007. Springer-Verlag.

[40] Dai Zhengguo. Thinning image. MATLAB CENTRAL, April 2010.