

*The real problem is not whether machines think but whether men do.*

– B. F. Skinner Contingencies of Reinforcement, 1969

**University of Alberta**

**3D MODEL REPRESENTATION AND MANIPULATION BASED  
ON SKELETONIZATION**

by

**Liang Shi**

A thesis submitted to the Faculty of Graduate Studies and Research  
in partial fulfillment of the requirements for the degree of

**Master of Science**

Department of Computing Science

©Liang Shi

Fall 2011

Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

*To my dear mom  
For giving me all the support to become a happy geek.*

# Abstract

3D model is a promising type of multimedia content for entertainment, research and education purposes. This thesis addresses the representation and manipulation of 3D models based on skeletonization, which is a commonly used technique to extract a compact descriptor and effectively capture the topological and geometric structure of 3D models. My research focuses on the refinement of skeletonization and its applications in 3D model matching, retrieval, and decomposition. By introducing a framework based on Scale-Space-Filtering (SSF), I integrate both the global node significance and the local chain-coded structure for pose-aware model retrieval; then adopt the topological mapping scheme for skeleton-based model decomposition. Experiment and comparison with state-of-art work on benchmark databases demonstrate the accuracy and efficiency of this framework.

The first key contribution of this thesis is the improvement of skeletonization results. I introduce an adaptive skeletonization framework using SSF, and propose the notion of node significance and bending measurement to extract the structural features of 3D curve skeleton, hence deriving a more robust descriptor for model simplification and registration. By adopting SSF, I conduct dynamic skeleton pruning and smoothing based on the initial results of thinning. Results of improved skeletonization are shown compared to other thinning methods along with time performances of four recent approaches. This approach integrates the connectivity and the effectiveness in computation, thus it achieves the balance between representation ability and speed requirement.

Another key contribution is the development on topology matching and

chain coding techniques for measuring model similarity. To demonstrate the effectiveness in 3D model classification, I validate the chain code encoded skeletons using the Princeton Benchmark database, with particular emphasis on distinguishing different poses of similar models.

Finally in this thesis I present an effective approach for model decomposition with enhanced semantics. By extracting robust skeleton and mapping model surface nodes to the decomposed skeleton branches, this method identifies the topology and geometry information of the 3D model. Thus it results in more meaningful segmentation components. Experiments on 2194 model demonstrate the advantage of this framework, comparing to three state-of-art approaches. According to animal anatomy, the proposed method keeps superior fidelity on four-leg animal models.

Overall, my research in this thesis proposes a novel approach to integrate the global information as well as the local geometry structure of 3D models into the process of skeletonization. Applications on model retrieval and decomposition fully proved its effectiveness and accuracy.

# Acknowledgements

I would like to thank my supervisors, Dr. Anup Basu and Dr. Irene Cheng, for their guidance, encouragement, and all the support during my study. Their insight and passion on research highly motivated and inspired me to achieve this thesis work. I also would like to thank my committee: Dr. Michael Bowling, Dr. Pierre Boulanger and Dr. Vicky Zhao for their suggestions and help to improve the work.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	3D models Simplification and Skeletonization . . . . .	3
1.2	Model Matching and Retrieval . . . . .	4
1.3	3D Model Decomposition . . . . .	6
1.4	Summary . . . . .	8
1.5	Thesis Outline . . . . .	9
<b>2</b>	<b>Review of Related Work</b>	<b>10</b>
2.1	3D Model and Curve Skeleton Representations . . . . .	10
2.1.1	Definition of Skeleton . . . . .	11
2.1.2	Desired Properties of Skeleton . . . . .	12
2.2	Skeletonization of 3D Models . . . . .	14
2.2.1	Thinning Methods . . . . .	14
2.2.2	Field-based Methods . . . . .	15
2.2.3	Geometric Methods . . . . .	15
2.2.4	Summary of Skeletonization Methods . . . . .	15
2.3	Content-based Model Indexing and Retrieval . . . . .	17
2.3.1	View-based Methods . . . . .	18
2.3.2	Mesh-based Methods . . . . .	18
2.4	Skeleton based 3D Model Matching . . . . .	22
2.4.1	Conceptualized Retrieval Model . . . . .	23
2.4.2	Skeleton Matching . . . . .	23
2.4.3	Skeleton Graph for Topology Matching . . . . .	24
2.5	Model Decomposition . . . . .	25
2.5.1	Mesh-based Methods . . . . .	26
2.5.2	Clustering-based Methods . . . . .	26
2.6	Summary . . . . .	27
<b>3</b>	<b>Methodology and Algorithms</b>	<b>28</b>
3.1	System Overview of Refined Skeletonization . . . . .	28
3.2	Geometrically Adaptive Skeleton Refinement based on Node Significance and Bending Measurement in Scale Space . . . . .	31
3.2.1	Skeleton Extraction using VDSM Thinning . . . . .	31
3.2.2	SSF Representation and Decomposition of 3D Skeleton . . . . .	32
3.2.3	Skeleton Pruning by Node Significance . . . . .	32
3.2.4	Skeleton Smoothing integrating Bending Measurement with Adaptive Filtering . . . . .	35
3.3	Encoding skeleton with Chain code for 3D model matching and pose recognition . . . . .	38

3.3.1	Skeleton Graph Maching . . . . .	39
3.3.2	Chain Code Encoding . . . . .	40
3.4	3D Model Decomposition based on Skeletonization and Topo- logical Mapping . . . . .	43
3.4.1	Topological Mapping with Decomposed Skeleton . . . . .	43
3.5	Summary . . . . .	45
<b>4</b>	<b>Experiment and Results</b>	<b>47</b>
4.1	Results of Refined Skeletonization . . . . .	47
4.1.1	Adaptive SSF-Refined Skeletonization . . . . .	47
4.1.2	Comparative Study on Time Performance . . . . .	51
4.2	Model Matching and Retrieval with Chain Coding . . . . .	54
4.3	Results of 3D Model Decomposition . . . . .	57
<b>5</b>	<b>Conclusion</b>	<b>61</b>
5.1	Contributions . . . . .	62
5.2	Discussion . . . . .	62
5.3	Future Research . . . . .	63
	<b>Bibliography</b>	<b>64</b>

# List of Tables

2.1	Comparison of skeletonization with desired properties. . . . .	16
4.1	Computation times of skeletonization methods. . . . .	53
4.2	Distance of chain code on the Armadillo model. . . . .	57
4.3	Metric distance on anatomy of horse model. . . . .	60

# List of Figures

1.1	Examples of skeletons of 3D models. . . . .	2
1.2	Pose divergence of the same model. . . . .	6
1.3	Example of model decomposition. . . . .	7
1.4	Comparison of model decomposition results. . . . .	8
2.1	Results of Skeletonization in comparison. . . . .	17
2.2	2D silhouettes views of cars. . . . .	19
2.3	Examples from Princeton 3D Search Engine. . . . .	19
2.4	Conceptual model of skeleton-based 3D model retrieval [53].	22
2.5	The node-to-node matching of two 3D models. . . . .	25
3.1	Overview of the robust skeletonization framework. . . . .	29
3.2	A example of the original thinning result and its refinement. .	30
3.3	Results of VDSM thinning compared to refined skeleton. . . .	31
3.4	Closest model vertices to the skeleton. . . . .	33
3.5	Pruning results. . . . .	34
3.6	Results of skeleton with enlarging filter size. . . . .	35
3.7	An example of angles for significance computation. . . . .	36
3.8	Steps of skeleton refinement. . . . .	37
3.9	System overview of model retrieval. . . . .	38
3.10	Result of topology matching of two skeletons. . . . .	39
3.11	Chain elements for 3D curves. . . . .	40
3.12	Orthogonalized skeleton with chain code expression. . . . .	41
3.13	Anatomy preserving model decomposition. . . . .	44
4.1	Comparison between original and filtered results. . . . .	48
4.2	Comparison between thinning and the proposed method. . . .	49
4.3	Decomposed skeleton pruning results. . . . .	50
4.4	Skeletonization and skeleton decomposition results. . . . .	52
4.5	Running time comparison of five approaches. . . . .	53
4.6	Precision-recall curve of model retrieval. . . . .	55
4.7	Five poses of the Armadillo model. . . . .	56
4.8	Results of model decomposition. . . . .	58
4.9	Comparison of decomposition methods. . . . .	59
4.10	Mapping between animal anatomy decomposition result. . . .	60

# Chapter 1

## Introduction

Nowadays, 3D models are commonly used in the entertainment industry of 3D Animation, Movies, Visual Reality and Games. They are also intensively used in robotics, tracking (visual surveillance), computer-aided design, as well as surgical training and medical diagnosis for education. Due to the application demands for interactive 3D graphics, many advanced modeling, digitizing and visualizing techniques have been developed in order to facilitate the creation and rendering of 3D models. Especially recent development in laser scanning and webcam reconstruction made it more approachable to build geometrically precise 3D objects.

Because of the increasing number of 3D assets in databases, a more succinct representation is necessary for the purpose of automatic model retrieval, matching, editing, registration, recognition and manipulation. Therefore, 3D simplification techniques are studied with intense interests across disciplines of research groups in computer graphics, virtual reality, robotics, human computer interaction (HCI), gesture recognition and generation (e.g., hand gesture and gait recognition). Many works cast a highlight on the content-based model analysis. Common 3D model retrieval engine utilize the 2D projections from 3D models such as the model contour and sketch [51], and a number of model rendering and retrieval systems have inspired further interests in developing more effective and representative features to enhance real-time 3D applications with more semantic-aware interactions.

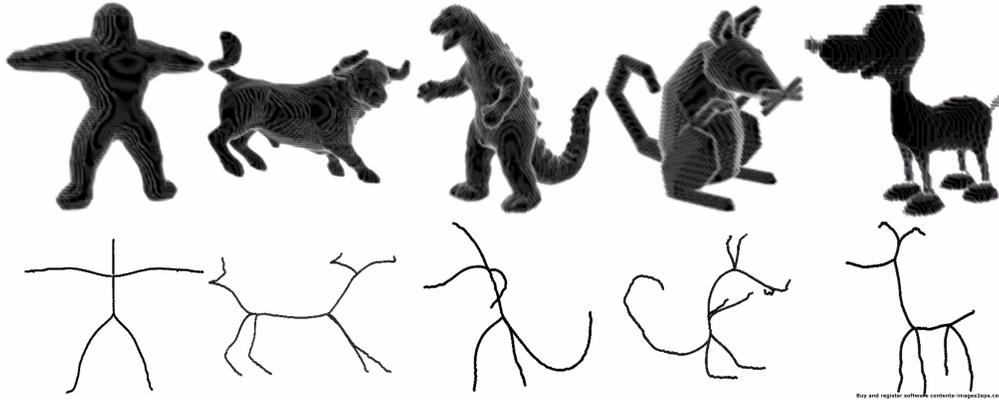


Figure 1.1: Examples of skeletons of 3D models.

Content analysis is challenging due to the ambiguity and multiple meaning of both 2D and 3D visual inputs. Compared to 2D objects in images, 3D models do not need to be segmented from the background, exhibit no projective deformation, and have no direct boundary parameterizations [52], and thus have many advantages in model retrieval. However, the representation of 3D models is much more complicated than its 2D counterpart, because of the additional dimension and more diversified formats varying from meshes to volumes and to point clouds. For efficient real time 3D model matching and retrieval, a simplified but compact and representative representation is therefore necessary.

One approach is to extract the “skeleton” of 3D models, which is a graph-like abstraction of 3D models at the center-line of the original mode, as shown in Figure 1.1. It contains the essential topology and geometric information, thus is a discriminatory and concise way to reduce the complexity in both dimension and scale. In addition, skeleton is easy to use in the way of part matching (finding components rather than global matching), visualization (showing the results), intuitiveness (editing by users to refine the search with perceptual understanding), articulation (dealing with similar objects of articulated poses) and indexing (having lower dimension in the search space) in the task of model retrieval. It has many applications such as model retrieval, 3D segmentation, direct manipulation and so on.

Because human visual system (HVS) is sensitive to changes on the shape

of an object [47], it is natural and logical to adopt the structural information into 3D model simplification, such as the 2D contour from a 3D model projection. However, because the projected contour can vary significantly from changes of view, this method is not stable to model transformation. On the contrary, using the skeleton or curve skeletons, which is invariant to isometric transformations, provides a more effective way for model simplification.

## 1.1 3D models Simplification and Skeletonization

Skeletonization, as a simplification approach, extracts the curve-skeletons from 3D models as a compact geometric and topological descriptor. During the recent decades, many efforts have been devoted to the extraction of curve skeletons, including thinning methods, field-based methods, geometric methods and etc. Thinning is more computationally efficient, and also protects the connectivity during computation; Field method usually does not depend on object orientation, and is less sensitive to noise; Geometric method performs relatively better in preserving the topology and hierarchical information with flexible model representations. For the task of model retrieval, since the connectivity of a skeleton and the speed in computation is of major concern, thinning is thus more commonly used. Nevertheless, the result of thinning is highly sensitive to both local and global noises.

Our solution to this problem is to improve the thinning-based skeletonization methods with a fast adaptive Scale-Space-Filtering (SSF) and the notion of node significance. This method provides the ability of noise elimination as well as skeleton registration. The skeletons generated preserve connectivity, thinness and smoothness in geometric and topological features. In order to demonstrate the effectiveness of this proposed refinement on thinning, I apply the skeleton results on two applications of model retrieval and decomposition.

## 1.2 Model Matching and Retrieval

As a result of the increasing number of 3D models, automatic model matching and retrieval has become necessary for searching the huge model repositories. Retrieval techniques are widely studied among the community of computer graphics, gesture recognition and human computer interaction. A number of model systems and search engines have been developed by leading research institutions: the Ephesus search engine at the National Research Council of Canada [43], the 3D model search engine at Princeton University [38], the 3D model retrieval system at the National Taiwan University [50], the Google 3D Warehouse, Junaio and 3DVIA search engines, etc. These developments have inspired further interests in enhancing real time 3D model retrieval techniques.

In 3D model retrieval, there are several factors to be considered:

- (1) Representation of the 3D model;
- (2) The measurement of dissimilarity;
- (3) Discrimination ability;
- (4) Ability to perform partial matching;
- (5) Efficiency;
- (6) Robustness;

Considering these factors, in order to simplify the representation of 3D model along with the measurement of dissimilarity, it's ideal to fully utilize the curve skeleton as the model descriptor in retrieval. The main challenge is how to generate skeleton that represents the model's semantic features, such as the topology and geometric fidelity; and another challenge is to balance between the trade-off of accuracy and efficiency. My solution is to adopt the fast linear-time thinning method for skeleton extraction, then enhance the results with SSF and node significance. This framework will produce noise-free skeleton, which also avails the registration between skeleton

junctions. The reason of selecting thinning to extract skeletons is listed as follows:

- (1) Thinning is commonly used as a simplified representation of the 3D model;
- (2) It can be encoded for the purpose of dissimilarity measurement;
- (3) Thinning can discriminate the key features from the original 3D model;
- (4) With the connected and unit-width skeleton, decomposed branches from the skeleton can be used on partial matching.
- (5) Fully parallel thinning is computed in linear time, thus it's fast and efficient;
- (6) Thinning results should be improved to reduce both global and local noises;

As we can see, the refined skeletonization based on thinning is ideal for model retrieval. Once the representative skeletons of 3D models are generated, the model matching problem is reduced to skeleton matching. In order to measure how similar two skeletons are, it is necessary to compute the distance between them. A commonly applied metric is skeleton graph matching [52], which measures the topology similarity between two skeleton graphs. However, this approach is limited when comparing similar models with different poses, as shown in Figure 1.2, where the same Armadillo model [51] with six poses are treated as identical in the topology matching. In many applications including 3D animation, games, computer-aided design, clinical detection and diagnosis, different poses of a model can lead to very different cause of actions. Therefore, it is necessary to take into consideration of both the distance from the topology matching and the geometrical similarity between two models. In this work, I perform topology matching and chain coding [10] based on the robust skeleton to uniquely discriminate the structures and curvatures of 3D skeletons. It results in more accurate retrieval results that can distinguish the pose dis-

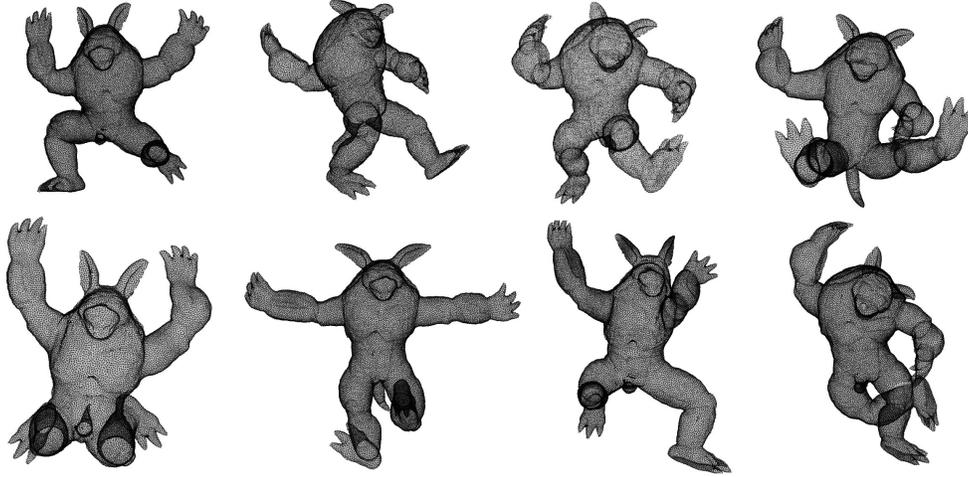


Figure 1.2: Pose divergence of the same model.

similarity, as shown in Figure 1.2.

### 1.3 3D Model Decomposition

3D model decomposition has its fundamental meaning in multiple applications, such as modeling, shape retrieval, model simplification, texture mapping and animation. Thus the demand for automatic model analysis and manipulation is fast-growing. The segmentation or decomposition of 3D models indicate the partition of the 3D mesh (point cloud models can easily be converted to mesh models). Some examples of segmented models are shown in Figure 1.3. This area has been profoundly studied by the computer vision and graphics community.

Geometry based segmentation methods emphasize partial consistency of the generated segments so that each of them is uniform in the sense of local curvature or distances. This type of approach is more about the characteristics associated with individual segments. The choice of model decomposition techniques is application specific, dependent on the requirements of some expected outcome. Current geometric techniques can be classified into six categories; namely, region growing, watershed based, clustering, spectral and field analysis, feature point based and graph based [5]. One

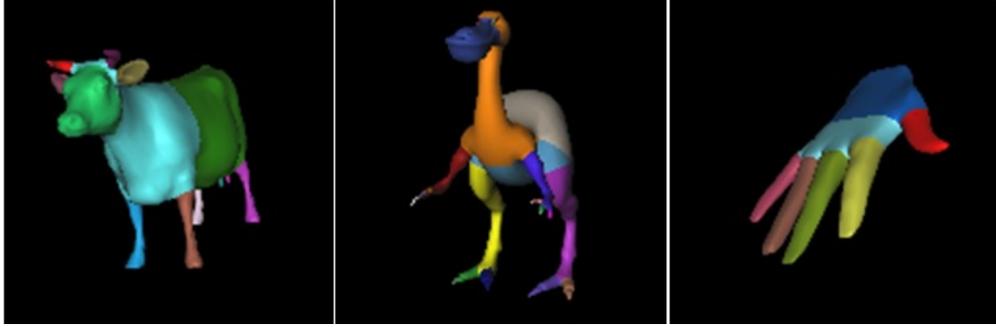


Figure 1.3: Example of model decomposition.

common weakness of these methods is that they only incorporate local features, such as curvatures and normal of the surface, lacking the global semantics associated with the 3D model. Since perceptual quality is very important in graphics applications, particularly for animations, my work focuses on performing decomposition with respect to the human intuition of segmenting a model and its topological layout. For example, in Figure 1.4, three examples of mesh-based segmentation are shown on the left; and one topology-based segmentation result from propose method is shown in comparison. It can be seen that the horse more is likely to be divided into the body, head, tail and legs to match the animal anatomy segmentation, instead of into excessive detailed patches on the mesh surface.

Semantic oriented methods have attracted more attention in recent years due to their appropriateness in mapping the decomposed mesh into perceptually meaningful partial components, for retrieval and animation. Although many research works have looked into generating natural shape decomposition from a human perceptual point of view, there is no obvious metric to evaluate the accuracy and quality of a decomposed result. Also, the granularity of the decomposition is hard to decide for many hierarchical methods. In this regard, relying on semantically rich surface features is necessary to decompose a model into meaningful parts, so that these parts can be perceived as integral components of the 3D model.

In order to make the decomposition operation more effective and accurate, I apply skeletonization of 3D models to guide decomposition, followed

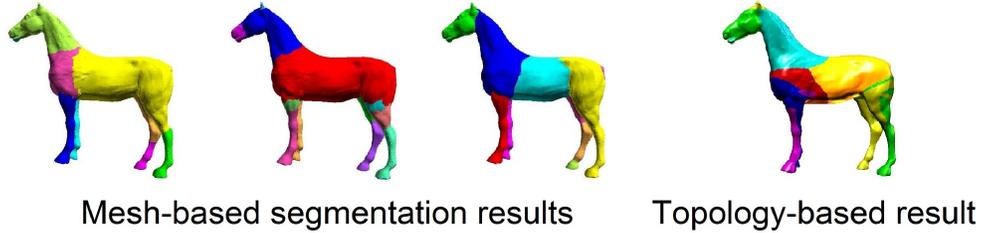


Figure 1.4: Comparison of model decomposition results.

by mapping surface points onto skeleton branches. This approach has the advantage of incorporating information from both global shape and local features, helping enhance perceptual quality. Results have enhanced correspondance between animal anatomy, which is required in animation and gaming applications.

## 1.4 Summary

Inspired by the problems of current research works, this thesis will focus on addressing the following issues:

- Propose an adaptive curve skeleton refinement method to reduce the noises generated from thinning in a systematic framework.
- Utilize the adaptively scheme of both global topology as well as local geometry features for effective model representation.
- Adopt the effective topology matching and skeleton coding method for the evaluation of model dissimilarity.
- Enable the discrimination of similar models with different poses.
- Provide an efficient mapping scheme from labeled sample nodes to the model surface points to achieve skeleton-based model decomposition.

## 1.5 Thesis Outline

The rest of the thesis is organized as follows: Chapter 2 reviews related work in skeletonization, content-based model retrieval and model decomposition. A detailed discussion of the proposed methodology and algorithms is presented in Chapter 3. Chapter 4 explains the setting of my experiments and compares the results with other start-of-art approaches. Finally, Chapter 5 gives the conclusion and discusses future work.

# Chapter 2

## Review of Related Work

For the task of 3D model representation and manipulation, there are many research works with diversified application background. In this chapter, a brief review is provided on the following topics: model and skeleton representation; skeletonization; model retrieval and model decomposition.

### 2.1 3D Model and Curve Skeleton Representations

To simplify the 3D objects into abstract representation, there are two types of most-commonly used methods: point cloud and mesh-surface. The point cloud is a set of unstructured points representing the model's external surface; and the mesh-surface or a polygon mesh or unstructured grid is a collection of vertices, edges and faces that defines the shape of a polyhedral object.

The properties and extraction methods of skeleton are highly related to the model representation. An important issue is the type of model representations that a shape retrieval system accepts. The most common format of 3D model is mesh-based, which is defined to support visual appearance [52]. Various formats are introduced to represent the models' surface as well as the volume. Such as the implicit surfaces, CSG (constructive solid geometry), BSP (binary space partitioning) trees, octrees, B-rep (boundary representation), free form surfaces, and etc.

A skeleton of a 3D model is a compact abstraction of the topological and

geometric information, which avails effective coarse-to-fine model information. Since the definition of skeleton is intuitive in most former works, I first compile a list of desired properties of skeleton. Based on this criterion I explore three categories of extraction methods by analyzing their suitability for the task of model representation.

### 2.1.1 Definition of Skeleton

One of the difficulties in the process is that the skeleton of a 3D model is ill-defined, which results in many heuristic methods with fine-tuned parameters and thresholds for specific application. In 3D space, skeleton is the medial axis of the shape, a center-spine or stick-figure like 1D representation within the 3D model. It is a subset of skeletal surface (medial surface), which is compact and captures the essential geometric and topological information of the underlying objects.

The skeleton of a region is defined by the medial axis transformation (MAT) [31]. The MAT of a region  $R$  with border  $B$  is defined as follows: for each point  $p$  in  $R$ , I find its closest neighbor in  $B$ . If  $p$  has more than one such closest neighbor, then  $p$  belongs to the medial axis (or skeleton) of  $R$ . The closeness depends on the metric used. To be more specific, the skeleton can be defined in terms of the distance of a point  $x$  to a set  $B$ , where

$$d(p, B) = \inf_{z \in B} d(p, z). \quad (2.1)$$

The function  $d$  can be any metric, such as the Euclidean metric, and  $\inf_{z \in B} d(p, z)$  indicates the shortest distance of point  $p$  to the points in set  $B$ . If more than one  $z$  exists, then  $p$  is defined to be on the skeleton of the object.

However, the definition of skeleton in differed applications is not rigorous. There is also a trade-off between the descriptiveness and sensitivity to noise of the skeleton [13]. Therefore, this intuitive definition of skeleton is not adequate to evaluate the “goodness” of skeletons. Instead, I should consider some explicit properties that are important for model representation.

## 2.1.2 Desired Properties of Skeleton

Because of the application-specified characteristics of skeleton, I elaborate the properties that are essential in representing and manipulating the 3D mode, pruned from the list in [13].

### **Topology preservation**

Though topology information is not complete to distinguish different 3D models, it is an effective and compact way to pre-classify models into several categories without precise comparison at the first stage. Therefore the topology preservation is the basic requirement in the process of skeletonization. It should represent the topological and geometric structure of the original 3D model, thus keep the consistency in model matching.

### **Centeredness**

Centeredness indicated the skeleton resided within the 3D model, and the distances from one nodes on the skeleton to the orthogonalized surface nodes are desired to be comparable. This feature is helpful in animation and topological mapping.

### **Smoothness**

Smoothness means the decomposed skeleton should have as less jitter effect as possible. For example, in visual navigation for medical examination, the smoothness of the skeleton can guarantee the smooth and fluent camera movement. Hence improve the viewing user experience.

### **Connectivity**

Connectivity ensures to transverse the skeleton from one endpoint with junction handling. It is important for constructing a hierarchy of curve-segments.

## **Invariance**

Invariance means the distance between points are preserved in the process of isometric transformations, in which the characteristics of skeleton should be preserved after translations, rotations and reflections of the 3D model. In model retrieval, it is common to have similar objects in different orientations that should be matched without specific registration and normalization, i.e., the shape descriptor must be invariant to object orientation.

## **Robustness**

The skeleton should have weak sensitivity to noise on the boundary of the object, that is, the skeleton extracted from a noise-free object and the one from the same objects with noise should be identical or similar.

## **Reconstructability**

This feature shows whether the 3D model is possible to be rebuilt by the skeleton extracted. Obviously we can use the maximal inscribed balls for this purpose, which has the center at the skeleton. But this operation also requires the centeredness of the skeleton to be true. This property reflects the differences between two models with similar topology.

## **Efficiency**

Efficiency of the curve-skeletonization algorithm is essential in applications that require real-time processing of skeletons for fast model computations and search.

## **Hierarchy**

Hierarchical approach is useful to reflect the natural level-of-details on the complex components of an object, which contains a subset of different scales of skeletons for multi-resolution matching. This feature is important for coarse-to-fine model retrieval.

## **Junction Handling**

The skeleton should be able to distinguish different junctions of the original object, reflecting its part or component structure. This implies that the logical components of the object should have a one-to-one correspondence with the logical components of the skeleton.

## **Variability**

Variability indicates the ability to handle different object representations, including voxelized objects, polygonal objects and unorganized point sets of the input example model.

## **2.2 Skeletonization of 3D Models**

The extraction of compact and expressive skeletons for 3D models is critical for the task of model representation and manipulation. I briefly describe three major categories of skeleton extraction methods: thinning methods, field-based methods and geometric methods.

### **2.2.1 Thinning Methods**

To produce a curve skeleton, thinning methods iteratively remove points of a 3D image from the boundary until no more points can be removed. These boundary points can be identified by inspecting the 26-neighbourhood in 3D space with a set of templates [13]. Since any single connected object without holes or tunnels will be reduced to a single point by sequential removal of boundary points, the approach adopts a constraint called “surface end points” to avoid over simplification of the skeleton. The thinning method was introduced by Morgenthaler in 1981 [39] and eventually developed into several sub-methods, including directional or border sequential thinning [23], subfield sequential thinning [36] and fully parallel thinning [33]. One of the classic thinning is proposed by Ma and Sonka [35] on 3D binary images in 1996, it is widely cited and applied in medical areas

as a benchmark. Thinning is effective in computation and reliable in keeping the topological information. Thinning is effective in computation and reliable in keeping the topological information, but it is sensitive to surface noise and the output skeleton often has jitter effects.

### **2.2.2 Field-based Methods**

The field-based method can be divided into two groups: the distance field and the general field method. The distance field method is based on the definition of medial axis transformation (MAT) [31]. It uses a distance function to compute a scalar or a vector field, which is then used to extract the curve skeleton by combining thinning or connection of the local maxima and ridge-points [24]. Instead of the distance function, other types of function can be used to generate a field and extract a curve skeleton, e.g., generalized potential function [14], visible repulsive force function [60], radial basis functions [37] and so on.

### **2.2.3 Geometric Methods**

While both thinning and field methods operate on voxel-based objects, geometric methods aim at objects represented by polygonal meshes or scattered point sets in continuous space. There are four main types of geometric methods: Voronoi diagram [9], Cores and M-reps [45], shock graph [30] and Reeb graph [6]. However, these methods are more computationally expensive because the field value at each point is influenced by more boundary points. They are also sensitive to noise and junction articulation [13].

### **2.2.4 Summary of Skeletonization Methods**

A comparison of the skeletonization results is introduced in a review of Cornea et al. [15], shown in Figure 2.1. It can be seen that distance field and geometric methods do not generate unit-width skeletons thus needing post-processing of pruning and connecting to create satisfying skeletons; thinning can generate connected well-centered and unit-width skeletons,

Table 2.1: Comparison of thinning, field-based and geometric extraction methods with desired skeleton properties.  $\checkmark$  means suitable,  $\times$  means unsuitable and - indicates possible.

<b>Skeletonization Methods</b>	<b>Thinning</b>	<b>Field-based</b>	<b>Geometric</b>
Topology	$\checkmark$	-	$\checkmark$
Centeredness	$\checkmark$	-	$\checkmark$
Smoothness	$\times$	$\checkmark$	$\times$
Connectivity	$\checkmark$	-	-
Invariance	-	$\checkmark$	-
Robustness	$\times$	$\checkmark$	$\times$
Reconstructability	$\times$	-	$\times$
Efficiency	$\checkmark$	$\checkmark$	-
Hierarchy	$\times$	-	$\checkmark$
Junction	-	$\times$	$\times$
Variability	$\times$	$\checkmark$	$\checkmark$

but has jitter effects with extra sub-branches; and potential field methods can provide smoothest skeletons among the results. On the other hand, though the potential field method generates better results, it is more time consuming than other approaches. For example for models with 30000 voxels, the computational time for potential field is above 50 seconds, while for thinning the time is less than 5 seconds. Recently, other linear approaches of skeletonization, such as the Euclidean skeletons from Roerdink et al. [25], demonstrate its computation efficiency. However, most of the experiments are conducted on simple 2D images and medical models, while the curve skeletons are not unit-width. Consequently, among the approaches that generate curve-skeleton directly, thinning is more effective and preferable for most applications.

And the suitability of these methods considering ideal skeleton features is summarized in Table 2.1 with respect to the desired properties for retrieval. It can be seen that there is no optimal method for all the properties. Thinning is more effective to compute, and also protects the connectivity during computation; Field method usually does not depend on object orientation, and is less sensitive to noise with general field; Geometric method performs relatively better in preserving the topology and hierarchy infor-

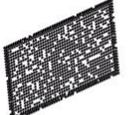
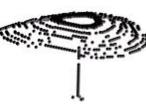
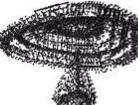
	Test Objects	Distance Field	Thinning	Geometric	Potential Field
Thin Block (34x9x54)					
Monster (54x87x75)					
Mushroom (80x87x59)					
Colon (204x132x260)					
Chess Piece (40x39x87)					
Chess Piece with 10% noise on the surface (39x38x86)					

Figure 2.1: Comparison of related skeletonization approaches from [15].

mation with flexible model representations.

Among these skeletonization methods, thinning is most computational efficient and can generate connected unit-width skeletons [58]. Therefore, I integrate the skeletonization refinement technique with the thinning method. In order to balance the trade-off between accuracy and speed, I will integrate the refinement from SSF and node significance techniques to enhance the performance of thinning.

## 2.3 Content-based Model Indexing and Retrieval

As the amount of various 3D models grows, there is an increasing need for effective model indexing and retrieval. Though there is text description attached to each model, it is not sufficient to include the visual features

from 3D models. Not to mention the accuracy of most descriptions. In order to identify what the model actually is, we need to utilize the content information, such as color, texture and shape. Since color and texture are easy to use but limited for models like proteins and texture-less models, I will discuss the adaptation of shape, which includes both 2D views, 3D mesh and model skeleton.

### **2.3.1 View-based Methods**

View means the 2D projection from 3D models. It assumes that the observations of similar objects will look similar from the same viewpoint. For example, the shape of a typical car from in the same view (front and side) looks alike, as shown in Figure 2.2. It contains the boundary of a shape from one specific view point. Compatible with using edges and contour in 2D images for object recognition, using several views of models can downsize the problem from 3D to 2D. Thus it becomes a matching problem between multiple sets of images.

The Princeton 3D model Search Engine [38] utilized two views from user's painting of the object silhouettes along with one text keyword to search 3D models. For example, in Figure 2.3 , the user has drawn outline contours specifying a shape, and the system has returned a set of matching objects.

The limit of this method is lack of flexibility which misses the 3D geometric and structural information, which requires a certain amount of different views to cover all the details of the 3D model. At the same time, the extraction and registration of 2D views are other concerns without user interaction.

### **2.3.2 Mesh-based Methods**

Given a shape, we need to find a proper measurement that discriminates this shape from others. Mesh-based methods analyze the general geometric and structural properties of the model as a whole. It considers the volume,

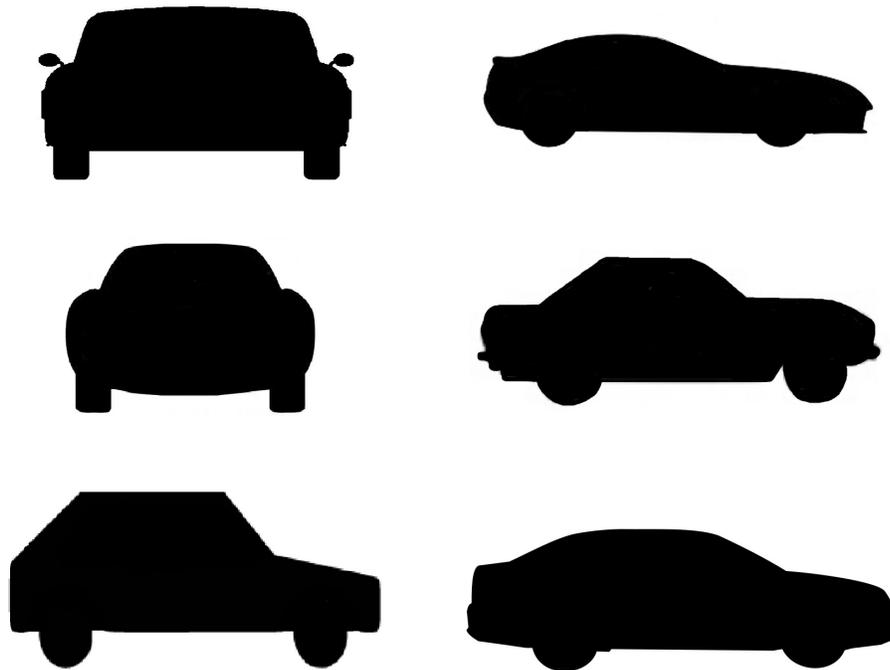


Figure 2.2: 2D silhouettes of cars from front views and side views.

Text & 2D Sketch

Search

Keyword:

Side View

Undo Clear

Front View

Undo Clear

Top View

Undo Clear

Princeton Shape Retrieval and Analysis Group

### 3D Model Search Engine

[Text & 2D Sketch](#)
[Text & 3D Sketch](#)
[File Compare](#)
[Research](#)
[Contact Us](#)
[Links](#)
[FAQ](#)
[Main](#)

Search results in database [espona], 1000 models (click on a thumbnail for more information on that model)

[Next page \(17 - 32\)](#)
search type: [2D sketch only], results: 100

1. mera (esp) <a href="#">Find similar shape</a>	2. nas1 (esp) <a href="#">Find similar shape</a>	3. nas2 (esp) <a href="#">Find similar shape</a>	4. cith (esp) <a href="#">Find similar shape</a>
5. mini (esp) <a href="#">Find similar shape</a>	6. fiat5 (esp) <a href="#">Find similar shape</a>	7. grand (esp) <a href="#">Find similar shape</a>	8. bmw502 (esp) <a href="#">Find similar shape</a>
9. quadfin (esp) <a href="#">Find similar shape</a>	10. vwgo1f (esp) <a href="#">Find similar shape</a>	11. karmann (esp) <a href="#">Find similar shape</a>	12. e900009 (esp) <a href="#">Find similar shape</a>
13. relojes (esp) <a href="#">Find similar shape</a>	14. verdi (esp) <a href="#">Find similar shape</a>	15. bie bu (esp) <a href="#">Find similar shape</a>	16. townhou3 (esp) <a href="#">Find similar shape</a>

Figure 2.3: An example of view-based model retrieval from Princeton 3D Search Engine [38].

aspect ratio, surface area, curvature or other kinds of numerical descriptions from the 3D model.

A key issue in model or mesh-based methods is how to deal with transformations. Usually these techniques put the models in a canonical coordinate system in the first step in order to keep it invariant to similarity transformation, including translation, rotation and scaling. One general way to deal with transformation is principal component analysis (PCA), which computes the principle axes of the model to match the corresponding canonical coordinate system.

Several categorize of approaches are briefly discussed as follows.

### **Feature Distributions and Spatial Maps**

This type of methods extracts the distributions of shape features to measure the global geometric properties. In Osada et al. [41], features include the angle between three random surface points, distance between one fixed point and another random point, or two random points, the square root of the area of the triangle formed by three random points, the cube root of the volume of the tetrahedron formed by four random points, and etc. These features are invariant to rotation and translation, but experiment shows that it performs well only on symmetric models [40].

Spatial Maps also computes the distribution of models, but it divide the model into sections and calculate the distribution distinctively. Ankerst et al. [2] introduced a methods involving building and computing the distance of shape histograms based on the discrete representations of the model. The shape histograms can be separated into three types: the shell model, the sector model and the spider-web model, in which the shell model is invariant to rotation.

### **Integral Transforms and Special Functions**

This method uses the coefficients of integral transforms and other kernel function, such as Hough transform [61], Fourier transform [57], Wavelet

transform [1], Radon transform [17] and Laplace transform [21]. These transforms are applied using their discrete form to create a feature vector as the shape descriptor.

### **Information Theory Approach**

Page et al. [42] proposed an approach to integrate information theory by measuring the shape complexity of 3D surfaces. It computes the entropy of the curvature to build the shape information.

### **Volumetric Difference**

Under the assumption that all the models occupy the volume differently, volumetric difference first normalize the pose of models, then present the model in a well-designed structure, such as the OBBTree [27]. The distance between two structures stands for the differences between two models. Some user interactions are also helpful in this method.

### **Local Shape Descriptors**

Instead of considering only one point as the feature point, this category add the relationship among its neighborhoods on the surface. To evaluate the local properties, curvature and partial matching is combined to join the local descriptors to a global one [28]. This type of methods do not need pre-processing of pose normalization, thus are faster compare to other approaches.

### **Weighted Point Sets**

Based on the point cloud of 3D models, this method sample a set of points on the model surface, and then weight them to measure the difference between two models. Tangelder and Veltkamp [54] propose several ways of creating a weighted point set, using Gaussian curvature, facet normal variation and the geometric structure. The similarity is measured by adopt-

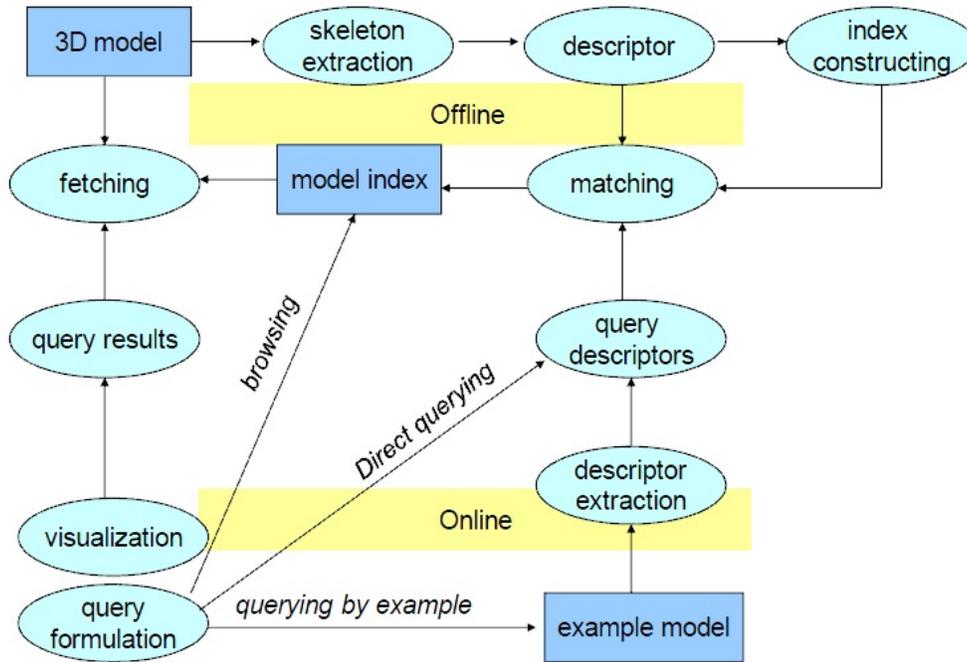


Figure 2.4: Conceptual model of skeleton-based 3D model retrieval [53].

ing the earth mover’s distance, and experiment demonstrated better performance compared to Osada’s method [41].

## 2.4 Skeleton based 3D Model Matching

Model matching generally consists of the problem of effective finding promising candidates from the database given a query. Since model retrieval deals with the matching of an input object model with a huge database of models for extracting the most similar on through a large number of comparison operations, an effective descriptor for each model is quite important, for the model matching is simplified into the similarity measurement of descriptors. I use the term of dissimilarity to indicate the notion of distance: small distance means the models are similar to each other.

### 2.4.1 Conceptualized Retrieval Model

The process of skeleton-based 3D model retrieval is conceptualized by Figure 2.4, adapted from the shape retrieval survey by Johan et al. [53]. The query could be browsing, direct query (descriptor query) or sample model query. Each is transferred to corresponding index and skeleton descriptor for further matching. Since the processes of index search and direct querying are trivial compared to the example-based querying, I mainly discuss the descriptor extraction and corresponding similarity measurements in model matching, which are done identically both offline and online. To be more specific, there are 3 major problems: (1) Skeleton description; (2) Skeleton extraction and simplification; (3) Skeleton matching.

### 2.4.2 Skeleton Matching

Given the skeleton of a 3D model, the model matching problem is reduced to skeleton matching. In order to measure how similar two skeletons are, it is necessary to effectively compute distances between pairs of descriptors using a degree of their resemblance. Commonly, there are two types of descriptors: skeleton graph [52] [7] and chain code [10] [58]. The former is a classic approach combined with graph matching and is widely adopted, while the latter is a recent promising work as a more discriminatory descriptor for post difference between 3D models.

To adopt a skeleton as the descriptor in model matching, many researches focus on formatting the skeleton as a topology graph. By making use of the well-studied graph matching algorithms, finding the similarity between two 3D models is reduced to finding the similarity between their skeleton graphs. Shokoufandeh et al. [18] combined spectral and geometric neighborhood information to match multi-scale blob and ridge decompositions in a coarse-to-fine manner. This method is based on a metric-tree representation of labeled graphs with the metric embedded into normed vector spaces. But this approach is limited by the fact that two graphs to be

matched were typically embedded into vector spaces with different dimensions. Siddiqi et al. [44] used spectral graph characterization to match shock graphs, and Pelillo et al. converted hierarchical matching into a maximum clique problem. They emphasized on the hierarchical organization of the graphs by constructing the association graph using graph-theoretic maximal sub-tree isomorphism. This method converts the problem into an indefinite quadratic problem using the Motzkin- Straus theorem. Belongie et al. [4] presented a scheme to measure the similarity between two shapes by solving the correspondence between points on the shapes and using the correspondence to estimate an aligning transform.

### 2.4.3 Skeleton Graph for Topology Matching

Since graph matching is a well-developed topic in the mathematical discipline of graph theory, methods based on skeleton graph are quite popular in the domain of model matching [16] [48]. The skeletal graph is usually computed directly from the 3D object containing the mean, radius, degrees of freedom about the joint, degree of importance of a particular joint or node in terms of the graph and local shape descriptors, which are held at each node in the graph [13]. An example of node-to-node match between two models is shown in Figure 2.5.

To decide whether two nodes match with each other, we need to consider two factors: (1) Topological similarity of the sub-trees rooted at the nodes; (2) Local shape information at the node. The output results of matching are:

- The number of matched nodes
- The sizes of clusters of matched nodes
- Detailed specification of which nodes are matched

As shown in Figure 2.5, the corresponding colored nodes are matched based on these factors. Typical method [52] combined a greedy form of

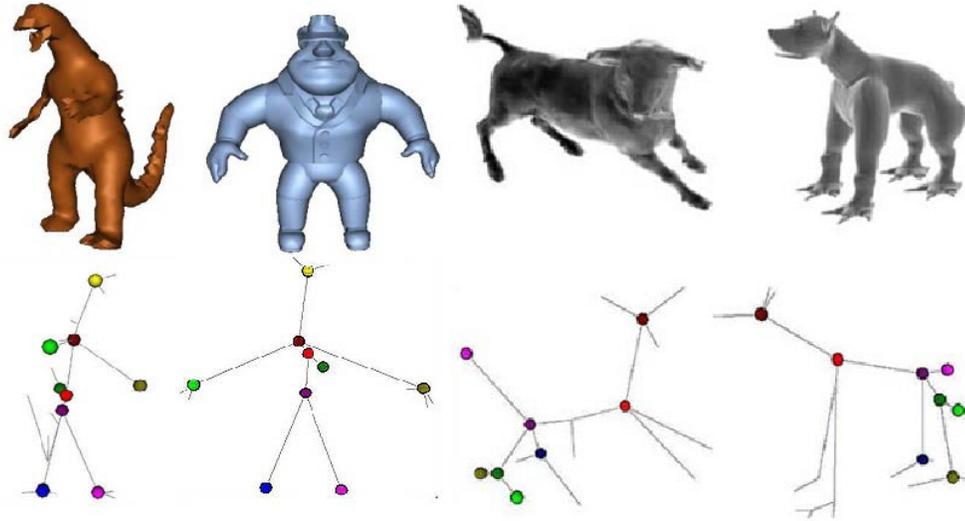


Figure 2.5: The node-to-node match in two 3D models of a human and an animal [13].

the bipartite formulation with a recursive depth-first search to preserve the hierarchical relationships in the graph. Each non-terminal node is assigned with an eigenvector of the subgraph adjacency matrix rooted at that node for indexing. Afterward, the problem is formulated as largest isomorphic subgraph problem, and the matching task becomes finding the maximum cardinality and minimum weight matching in a bipartite graph [14].

This type of method is robust to perturbational noise because it has small effect on the eigenvalues [53]. It also supports partial matching from the percentage of nodes matched in a certain region on a node-by-node basis. The drawback is that it only computes the dissimilarity from differed topological and geometric layout, but ignores the pose of the model, i.e., the orientation of each component. As a result, skeleton graph is not suitable to distinguish similar models with different curvatures.

## 2.5 Model Decomposition

3D model decomposition is to segment the surface of mesh models into subdivisions, according to either geometric or perceptual criterion [3]. It has wide application potentials in education, entertainment (game and ani-

mation), CAD and model partial retrieval. Current decomposition methods can be classified into five groups of region growing, watershed based, clustering, spectral and field analysis, feature point based and graph based [5].

### **2.5.1 Mesh-based Methods**

Mesh-based methods generate a segmentation boundary on the surface mesh of the 3D model. There are two key factors in this type of methods, one is the features it uses and the other is the criteria to judge the boundary.

In Region growing [5], first some distinct seed elements are generated, such as points, triangles and regions. Then they are expanded to grow until the set of terminating rules are met. It uses the features of Variable-order approximating polynomials, normal of the point clouds, principal curvatures, Dihedral angle of the adjacent triangles and Super-quadratics. The criteria for this method is based on distance of the points from the polynomial surface, comparative normal orientation, Gaussian curvature by user defined threshold and convexity validation based on the dihedral angles. This method can be either surface-based or part-based.

In contrast, Watershed based methods [42] adopt the deviation from flatness and curvatures, including Gaussian, mean, root and absolute, to determine the segmentation boundary based on Watershed function  $f$ . The function is defined as points belong to the catchment basins that the function  $f$  creates; minimum curvatures or normal curvatures with threshold; edges belong to the catchment basins that the function  $f$  create.

### **2.5.2 Clustering-based Methods**

In this category, model decomposition is achieved by an iterative clustering, which can use various clustering scheme. For example, Attene et al. [3] proposed a hierarchical method to classify the mesh triangles into clusters. It provides a high-level structure that can be interpreted semantically. However, it only performs well with narrow range of mesh surface such as spheres and cylinders.

There are also other types of method including the Reeb graph, Spectral analysis, Explicit Boundary Extraction and Markov Random Fields based methods. Besl and Jain [5] perform region growing in range images. They initially label the data points using the mean and Gaussian curvature. Using this labeling, they construct seed regions from which region growing will occur. The method of region growing fit variables to the growing region and neighboring data points are added according to their compatibility with the approximating polynomial. Here the termination criterion is satisfied when the region cannot grow any further. Their algorithm has been later extended by Vieira and Shimada [55] to 3D meshes.

The major problem of these methods is that they only incorporate the local features such as curvatures and normal of the surface, lacking the global semantics in correspondence with the topology fidelity.

## 2.6 Summary

In this section, some former research on 3D model skeletonization, Model retrieval and decomposition are addressed. The work of this thesis is inspired by the current problems that model representation and manipulation are either locally on surface mesh or globally on feature distribution. There should be a more intrinsic way to combine these two types of information. To solve this problem, there are several guidelines for my research work:

- Consider both the local features such as surface shape as well as the global geometric structures.
- Build a systematic approach to integrate the task of model retrieval and segmentation.
- Fully utilizes the advantage of skeletonization representation to simplify the measurement of model features.
- Provide a flexible and adaptable scheme for different types of models.

# Chapter 3

## Methodology and Algorithms

In this chapter, firstly, an overview of the refined skeletonization approach is provided in Section 3.1. Section 3.2 proposes the details of generating skeleton using thinning, the robust skeleton representation of the 3D models with Scale-Space-Filtering, and the incorporation of node significance. Then I integrated the chain-coding for effective model matching and retrieval in Section 3.3. Another application of model decomposition is discussed in Section 3.4. It maps the points on the mesh model to each sub-branch, combining the factors of surface curvature and normal to modify the boundaries of the segments. The result divides the model according to the semantic structures and is more feasible for model manipulation in animation.

### 3.1 System Overview of Refined Skeletonization

The overview of skeletonization refinements is shown in Figure 3.1. The input is a sample 3D model, and the output is a refined curve skeleton. To be more specific, there are three major processes: (1). Extraction of skeleton from thinning methods; (2). Skeleton pruning; (3). Skeleton smoothing. The focus is on the last two steps to generate robust skeletons based on the initial skeletonization results of thinning.

The definition of good skeletonization is actually ill-defined, mostly by subjective judgment from end users. However, there are several criterions

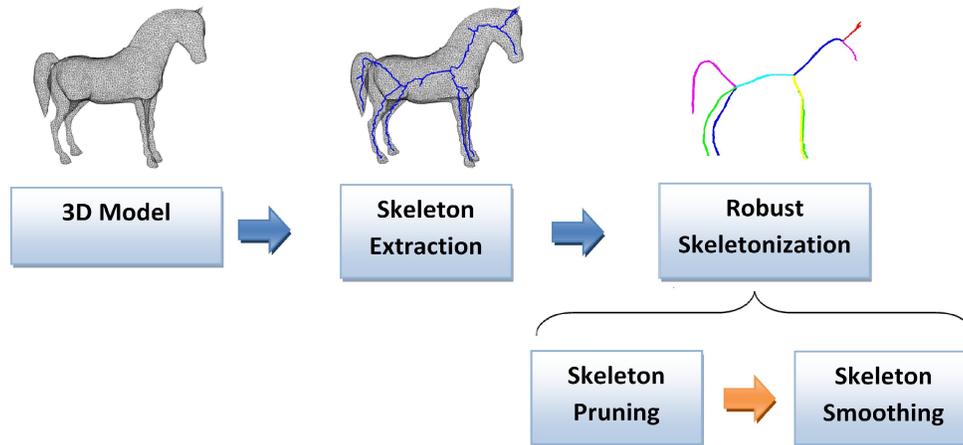


Figure 3.1: Overview of the robust skeletonization framework.

that a robust skeleton representation should meet, such as:

- The representation should preserve the important geometric information of the model at different levels of detail. Moreover, it should be clear from the representation which features belong to coarser levels of detail and which features belong to finer levels.
- The generation must be computationally efficient.
- The representation should be invariant under rotation, uniform scaling, and translation; otherwise, reliable registration will not be possible.
- The amount of change in the representation should correspond to the amount of change made to the curve. In other words, a small change to model parts should create a small amount of change in the skeleton representation.

Thinning is effective in protecting the topological invariance and computational efficiency. I aim to keep these features, especially the connectivity and geometric fidelity to enhance the skeleton output.

Considering thinning-based results, there are two types of noise involved: one is extra sub-branches, and the other is the local noise on each branch.

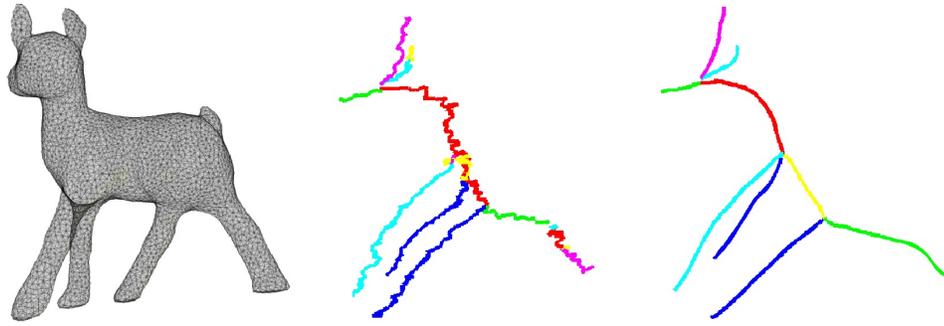


Figure 3.2: A example of the original thinning result (middle) and its refinement.

Undesired sub-branches are often present in curve skeletons, causing inaccurate topological matching. Local variations on the 3D model surface also cause unwanted jitter effect on skeleton branches, illustrated in the middle column of Figure 3.2. These undesired structures have adverse effects on applications of virtual navigation, direct manipulation, model indexing and matching. In order to minimize the global and local noise, I apply an adaptive refinement technique motivated by Scale-Space-Filtering (SSF). By combining SSF with skeleton pruning and smoothing, we can extract a robust skeleton providing precise structural information on a 3D model. The integration of SSF provides the flexibility to use a dynamic window size adaptable to a model's scale. I first conduct skeleton pruning to eliminate branch noise based on the junction and end point significance. This process removes extra sub branches and produces a cleaner graph for topology matching. Next I apply the Gaussian filter on the pruned skeleton to smooth each branch. The results of skeleton refinement including both pruning and smoothing can be seen in Figure 3.2. Since it incorporates local geometric information in the heuristic node-significance without iteration, the algorithm is simple and fast, which provides results comparable to other methods with less computational expense. More details on the refinement technique are given in Section 3.2.

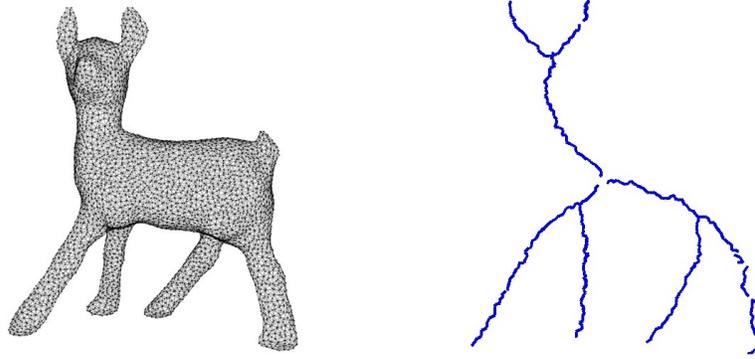


Figure 3.3: Comparison between VDSM thinning and refinement of the skeletonization results.

## 3.2 Geometrically Adaptive Skeleton Refinement based on Node Significance and Bending Measurement in Scale Space

### 3.2.1 Skeleton Extraction using VDSM Thinning

Due to the advantages of thinning in topology originality and time efficiency, I will extract skeleton using thinning. Though thinning is aimed at preserving the connectivity and thinness of the skeleton, it's found that the classic methods from Ma and Sonka [35] do not always generate connected output. A proof is given by introducing P-simple points [34]. To solve this problem, several efforts have been proposed to modify the original thinning algorithm [34] [58]. And I adopt one of the up-to-date fully parallel method to extract curve skeletons utilizing the VDSM (Valence Driven Spatial Median) technique [58]. Because it generates both connected and unit-width curve skeleton without crowd regions, so this method is ideal for the smoothing and pruning processes.

This VDSM approach works on both 3D binary images and 3D meshes by the voxelization pre-processing step. It is computational efficient without the need of using control parameters. Some results after VDSM thinning are shown in the middle column of Figure 3.3.

### 3.2.2 SSF Representation and Decomposition of 3D Skeleton

A 3D skeleton, as a composition of 3D curves, does not behave like a single-valued function in general. A parameterization of the curve should be found which makes it possible to compute the curvature of the curve at various levels of detail. To simplify the filtering process, I first decompose the 3D skeleton into separate branches by disconnecting them at the junctions (a point in a skeleton where it has more than two neighbors). The number of neighbors of a skeleton node is called its degree. As shown in Figure 3.10, the junction point has degree  $\geq 2$ , joining several branches of the skeleton bones; while an end point has degree 1, being at the end of each sub branch. After decomposition of the skeleton, each sub problem only involves one curve to be processed, which may include a branch (a part of the skeleton terminated by a junction or an endpoint) or a circle. In other words, each decomposed branch from the skeleton to be filtered is treated as a signal in 3D space. For this three-dimensional signal  $f : \mathfrak{R}^3 \rightarrow \mathfrak{R}$ , its scale-space representation  $L : \mathfrak{R}^3 \times \mathfrak{R}_+ \rightarrow R$  is defined in [32] as:

$$L_k(\boldsymbol{\epsilon}) = \int_{\xi \in \mathfrak{R}^3} f_k(\boldsymbol{\epsilon} - s) g_k(s) ds \quad (3.1)$$

where  $k$  is the index of branches and  $g$  denotes the Gaussian kernel:

$$g_k(\boldsymbol{\epsilon}; t_k) = \frac{1}{(2\pi t_k)^{3/2}} e^{-(\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2)/2t_k}. \quad (3.2)$$

The variance  $t_k$  of this kernel is referred to as the 'scale parameter'  $t$  for the branch indexed by  $k$ .

### 3.2.3 Skeleton Pruning by Node Significance

Since the applications of model matching and manipulation based on the skeleton graph rely on the topological fidelity, the pruning of the skeleton is essential for effective skeleton registration and matching. The basic idea of pruning is to distinguish the branches as 'major bones' or the 'sub-branch

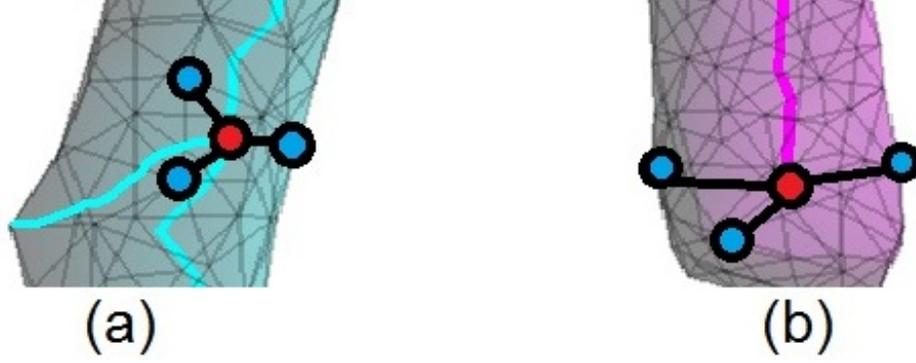


Figure 3.4: Closest model vertices to the skeleton node. (a), (b) are two portions of the leg from the sheep model, in which the cyan and purple lines are the extracted skeleton branches. And the red nodes are on the skeleton branches, while the blue ones are vertices on the surface of the 3D model.

noise’. Usually the major bones stands for the topological structure as a sketch of the model, while the sub-branch noise is triggered by some surface angle or block. In the field of image processing, pruning has been a widely studied topic. However, the pruning of 3D skeleton has not been addressed effectively. Considering thinning results, I notice that the extra branches are always less significant in terms of length and topological position. Thus, I introduce the idea of node significance to discriminate undesired extra branches. The significance of each node is computed by traversing the skeleton nodes, which is defined as follows:

$$S(v) = \begin{cases} 0 & \text{only if } degree(v) = 2 \\ \sum_{i \in k_v} l_i + \sum_{j \in C_v} d_j & \text{else} \end{cases} \quad (3.3)$$

where  $S(v)$  means the significance of skeleton node  $v$ ;  $k_v$  denotes all the branches connected to node  $v$  and  $l_i$  is the length of the branch;  $C_v$  is the set of the ten closest model surface vertices to the skeleton node measured by Euclidean distance, and  $d_j$  is the corresponding distance. As shown in Figure 3.4, the red nodes are junctions or end points on the skeleton, and the blue nodes are the vertices on the surface of the 3D model with closest distances to the skeleton nodes. All those skeleton nodes with degree two are intermediate points in a branch to keep the connectivity of the skeleton,

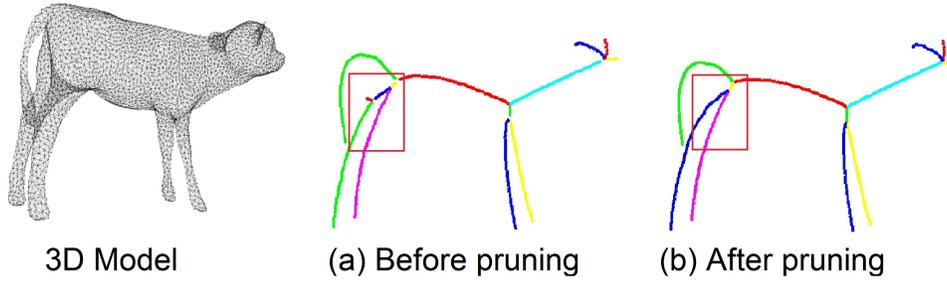


Figure 3.5: Example of pruning result. (a) and (b) show the results of skeleton before pruning and after pruning. The red rectangle shows that the extra sub branch is removed after pruning.

and they are not critical for the branching structure of the skeleton. So I will set their significance to zero and only evaluate those junction points or end points. This definition is also used to distinguish between junction nodes from the topology and the geometric characteristics of the 3D model.

In many cases, the sub-branches appear near the articulations of models due to higher curvature of the model surface, as shown in Figure 3.4(a). In order to remove these noises, I filter each end point with the significance of this node and its neighboring nodes. Suppose the current end point is  $v_m$  and  $\text{degree}(v_m) = 1$ . If  $S(v) > 0$  and  $S(v) < \frac{\text{degree}(v_m) \sum_{i \in n} \frac{S(v_n)}{\text{degree}(v_n)}}{t_k \times n}$ , in which  $\frac{S(v_n)}{\text{degree}(v_n)}$  stands for the averaged significance of the whole model nodes and  $n$  is the total number of the junction and end nodes, I will delete this point along with its sub-branch.  $t_k$  is the parameter in SSF so that the significance is normalized to be adaptive to the model scale. See more descriptions in Section 3.2.4. Repeat this process on each end point and traverse the skeleton structure to delete less-important branches. To be noted, only the nodes with degree of 1 will be removed, as the end node of a sub-branch, while all the other nodes will be kept. As a result, the pruned skeleton are free of extra branches, and the total number of junction nodes decreases due to reduction of sub-branches. In order to join the other branches together, we need to decompose the skeleton again. For example, in Figure 3.5, the extra branch on the leg of the sheep model is removed, while the two branches

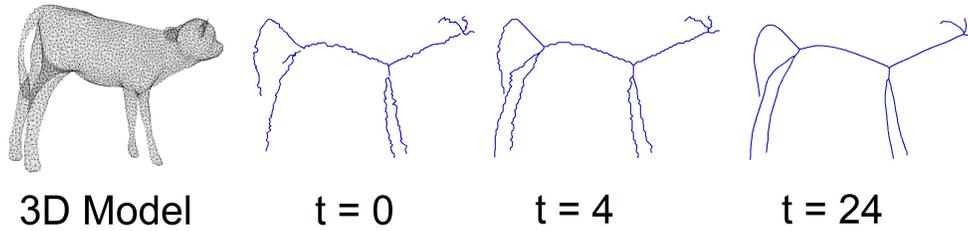


Figure 3.6: Results of 3D skeleton simplification on smoothing by SSF with enlarging size of Gaussian filter.

(blue and green) separated in (a) are joined in to one sub branch. More experimental results demonstrate the effectiveness of my approach to delete noisy branches near the articulations of models, as shown in Figure 4.3.

### 3.2.4 Skeleton Smoothing integrating Bending Measurement with Adaptive Filtering

Applying SSF in the 3D skeleton refinement process has the advantage of generating a spectrum of levels-of-detail and the generation is controlled only by one-parameter. The conditions (scale-space axioms) that specify the uniqueness are essentially linearity and spatial shift invariance, combined with formalizing the notion so that new structures should not be created in the transformation from fine to coarse scales. The goal is to smooth a skeleton (from a fine to a coarser scale) removing local noises and undesirable sub-branches.

In the refinement process, I first apply filtering on the coordinates of each decomposed part separately in 3D space. So each branch will be smoothed by convolution with Gaussian kernels of increasing window size. I traverse each decomposed part three times in the  $x$ ,  $y$  and  $z$  directions, using the previous filtered values. The major concern is how to automatically determine the filter parameters required by the Gaussian kernel. Note that each branch of the skeleton is associated with a different level of noise and curvature distribution. It is not trivial to determine the proper window size for SSF. As shown in Figure 3.6, different window sizes represented by  $t$  have strong

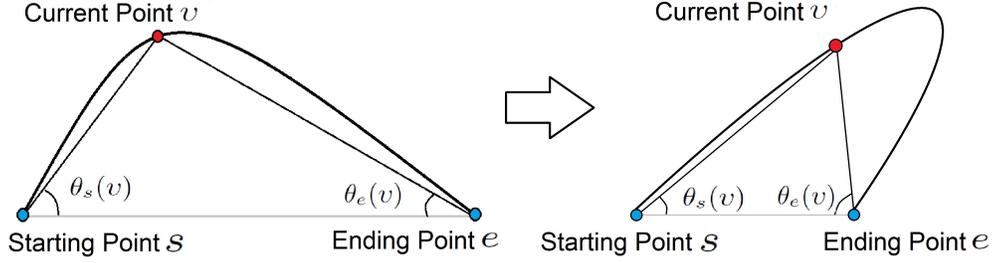


Figure 3.7: An example of angles for significance computation. The black thicker line shows one branch and the current node is got by traversing the branch from starting point to the ending point. The larger the angles  $\theta_s(v)$  and  $\theta_e(v)$ , the larger the curvature of this branch.

influence on the smoothing effect. Parameter  $t = 0$  indicates the original skeleton with a lot of local and global noises. Also note that the outcome is ideal to represent the topology of the sheep model when  $t = 24$ . If the scale parameter keeps increasing, the curve will be suppressed so the skeleton will change its centerness and might locate out of the model. To address this issue, I introduce an adaptive strategy in the selection of filter window size by computing the bending measurement  $F(b_k)$  of the current skeleton branch. First I pre-process the skeleton by applying SSF with a small window size  $t = 4$  to get rid of the local isolated nodes. Then when traversing each branch, the bending of the current branch is defined as follows:

$$F(b_k) = \frac{\int_{v \in b_k} (\sin \theta_s(v) + \sin \theta_e(v)) dv}{2 \times n(k)} \quad (3.4)$$

in which  $\theta_s$  and  $\theta_e$  are the angles from the current point  $v$  to the starting and the end point. A 2D example is shown in Figure 3.7.  $k$  is the index of the current branch.  $n(k)$  is the total number of points of the branch. The window size  $t$  of the branch  $k$  is computed as  $t_k = 1/F(b_k) \times n(k)$ , so  $t_k = (\int (\sin \theta_s(v) + \sin \theta_e(v)) dv) / 2$ . In this way, the filter is dynamically adapted to the model geometry and the branch scale, which ensures a better smoothing result. At the same time, the linear computing without iteration helps to achieve fast computation.

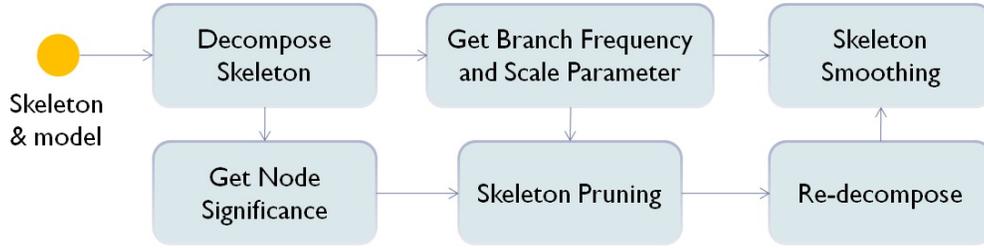


Figure 3.8: The computing procedure of skeletonization refinement.

This method can operate in different scales by varying one parameter  $t_k$ , and thus is very effective in controlling the smoothing effect. To summarize, the adaptive SSF refinement process is composed of several steps, as shown in Figure 3.8:

- Step 1.** Decompose the skeleton by dividing the branches at the junction nodes;
- Step 2.** Measure the significance  $S_k(v)$  of each node; Determine the corresponding SSF window size  $t_k$ ;
- Step 3.** Filter the branches with the significance level of its junction points. If the  $S_k(v)$  of the end point is less than the threshold, delete the node and the sub branch;
- Step 4.** Re-decompose the skeleton based on pruned skeleton result;
- Step 5.** Analyze the filter region based on curvature representation  $F(b_k)$  of each branch  $k$ ; Adjust the scale parameter  $t_k$ ;
- Step 6.** Apply Gaussian filter to smooth the skeleton with dynamic window size;
- Step 7.** Return the pruned and smoothed skeleton structure;

There are two major differences between my algorithm and the traditional SSF approach: Firstly, this system integrates both smoothing and pruning under the SSF representation by introducing the node significance; Secondly, it is adaptive to varied model size and level of noise, thus more flexible to achieve automatic skeleton improvement.

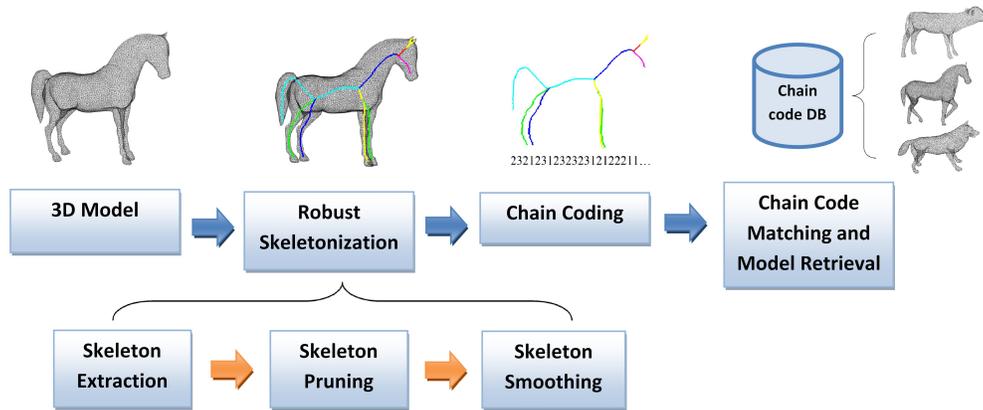


Figure 3.9: System overview of model retrieval.

### 3.3 Encoding skeleton with Chain code for 3D model matching and pose recognition

Our skeleton-based 3D model retrieval approach contains several steps as shown in Figure 3.9. The input is a sample 3D model, and the output is a set of similar models from the model database. The similarity between two models is evaluated using the chain code metric. To be more specific, there are three major processes: (1) Generation of a robust skeleton, which includes skeleton extraction, skeleton pruning and smoothing; (2) Skeleton encoding using chain code; (3) Chain code matching and model retrieval.

Given the skeleton of a 3D model, the model matching problem is reduced to skeleton matching. In order to measure how similar two skeletons are, it is necessary to effectively compute distances between pairs of descriptors using a degree of their resemblance. In this section, I integrate the descriptor of skeleton graph [52] [7] and chain code [10] [58] for effective model matching, along with their corresponding similarity measurements. Thus it achieves the branch matching from topological node-to-node matching, then compute the dissimilarity scores between corresponding matched branches.

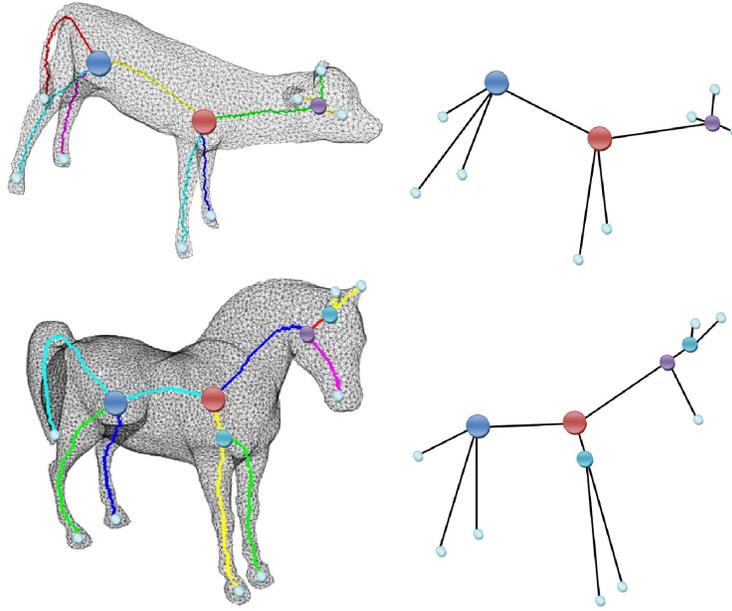


Figure 3.10: Result of topology matching of two skeletons. The size of each node indicates the significance.

### 3.3.1 Skeleton Graph Matching

Skeleton graph formulates the extracted skeletons in an undirected graph for topology matching. Since graph matching is a well-developed topic in the research of graph theory, methods based on skeleton graph are quite popular in the domain of model matching [16] [48]. The skeleton graph is usually computed directly from the 3D object containing the mean, radius, degree of freedom about the joint, degree of importance of a particular joint or node in terms of the graph and local shape descriptors, which are held at each node in the graph [13].

To decide whether two nodes match with each other, we need to consider two factors: (1) Topological similarity of the sub-trees rooted at the nodes; (2) Local shape information at the node. The output of matching includes the number of matched nodes, the sizes of clusters of matched nodes, and a detailed specification of which nodes are matched.

I adopted the method [52] with a greedy form of the bipartite formula-



Figure 3.11: Chain elements for 3D curves.

tion and a recursive depth-first search to preserve the hierarchical relationships in the graph. Each non-terminal node is assigned with an eigenvector of the subgraph adjacency matrix rooted at that node for indexing. Afterward, the problem is formulated as the largest isomorphic subgraph problem, and the matching task becomes finding the maximum cardinality and minimum weight matching in a bipartite graph [14]. I incorporate the skeleton graph matching method with the significance of each junction node to achieve the topological matching. Results can be seen in Figure 3.10. The corresponding color and size on the junction nodes indicate the matching results. The end points with the nearest junction point are matched by bottom up dynamic programming using the sub-tree edit distance [8].

This method is robust to perturbational noise because it has small effect on the eigenvalues [53]. It also supports partial matching from the percentage of nodes matched in a certain region on a node-by-node basis.

### 3.3.2 Chain Code Encoding

To obtain the ability for more detailed classification between different poses of the models, I make use of the chain coding to describe the geometric structure of each skeleton. The original chain code method was introduced by Freeman in 1974 [20], and recently adopted for representing 3D tree objects such as skeleton [10].

In a 3D skeleton, there are five possible descriptor values of the chain code, defined as follows in [10] [58]:

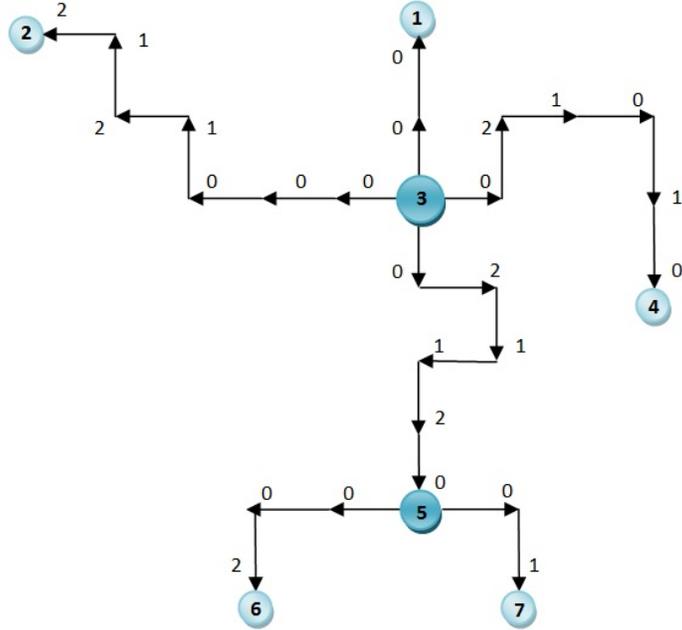


Figure 3.12: Orthogonalized skeleton with chain code expression. The number is the index of each node. The size and color indicates the significance of skeleton nodes. The bigger and darker, the more is its significance.

$$Chain\_Element(b, c, d) = \begin{cases} 0, & \text{if } d = c \\ 1, & \text{if } d = b \times c \\ 2, & \text{if } d = b \\ 3, & \text{if } d = -(b \times c) \\ 4, & \text{if } d = -b \end{cases} \quad (3.5)$$

where  $\times$  denotes the cross product. The chain code components are shown in Figure 3.11 [58].

Since there are only a fixed number of orthogonal chain codes, the SSF refined curve skeleton needs to be formalized by an *orthogonalization* [10] process. Because more than one possible descriptor can be chosen depending on the turning direction relative to the current position, the rule is to choose the highest descriptor value among the possible steps. The output skeleton is in a zig-zag representation, as shown in Figure 3.12. Since the choice of direction and the start point of coding can result in different ver-

sions of code expression, I decompose the skeleton after topology matching and encode each branch separately from the junction to the end point in order to generate a unique representation. Each orthogonalized skeleton is denoted by a chain code expression starting from its junction node on the skeleton.

For example, in Figure 3.12, junction node 3 and 5 have higher significance compared to end points 1,2,4,6,7, and thus the chain coding goes from the junction to the end. Meanwhile the significance of junction 3 is greater than junction 5, so the skeleton branch between them will be encoded from 3 to 5. For each branch in the skeleton, the resulting chain code is enclosed sequentially in order to distinguish the entire tree hierarchy from the root to its leaves. Then they are rearranged by the topological matching before the chain code matching.

To measure the similarity between two chain codes  $S$  and  $P$  with lengths  $l_s$  and  $l_p$  respectively, where  $l_s \leq l_p$ , I adopt the stretching and twisting operations described in [58]. The procedure compares the chain codes from the two sequences one descriptor at a time, and moves onto the next one if they are the same. If not, it performs one of the stretching or twisting operations defined in [58] as follows:

- $\int_{s \rightarrow p}$  - Stretch  $S$  by inserting the corresponding descriptor from  $P$ . The stretching operation terminates when  $l_s = l_p$ .
- $\vartheta_{s \rightarrow p}$  - Twist(bend) the descriptor in  $S$  to match the descriptor in  $P$ .

The final dissimilarity score of two skeletons is measured by

$$D(S, P) = \frac{\#\vartheta + \#\int}{\#S + \#\vartheta + \#\int} \quad (3.6)$$

in which  $\#S$  represents the number of same descriptor pairs, while  $\#\vartheta$  and  $\#\int$  represent the numbers of stretching and twisting operations. The overall distance is calculated by summing  $D(S, P)$  on each branch of the entire skeleton.

The primary benefit of the chain code approach is higher discrimination ability, considering both the topology as well as geometry similarity. It can distinguish various poses of the same model with a dissimilarity score between 0.06 and 0.19, and the score between different models is over 0.38 in the experimental results, whereas the graph based methods produce no dissimilarity for the same model with different poses.

One limitation of chain coding is from the orthogonal code, which will generate artificial stairs effect in a straight diagonal line. To solve this problem we can introduce a more flexible chain coding by increasing the code to eight directions.

### **3.4 3D Model Decomposition based on Skeletonization and Topological Mapping**

Having the refined skeleton, we can use it to guide the decomposition operation by mapping surface points onto skeleton branches. This approach has the advantage of incorporating information from both global shape and local features, helping enhance perceptual quality with more semantic meanings.

#### **3.4.1 Topological Mapping with Decomposed Skeleton**

With the decomposed 3D skeleton, we now have the topology reference for model decomposition. The problem is downsized into model points labeling according to the label of skeleton branches. First, the major concern of the mapping operation is to keep the fidelity of the decomposed skeleton that contains the topological layout of the model. The label of each sub-branch should be cast on to the model by generating connections between the skeleton and the mesh model. Since I adopt thinning for skeleton extraction, all skeleton branches are embedded in the model. Consequently, we can adopt simple mapping methods by using the distance and normal between skeleton and surface nodes. Also the curvature should be included

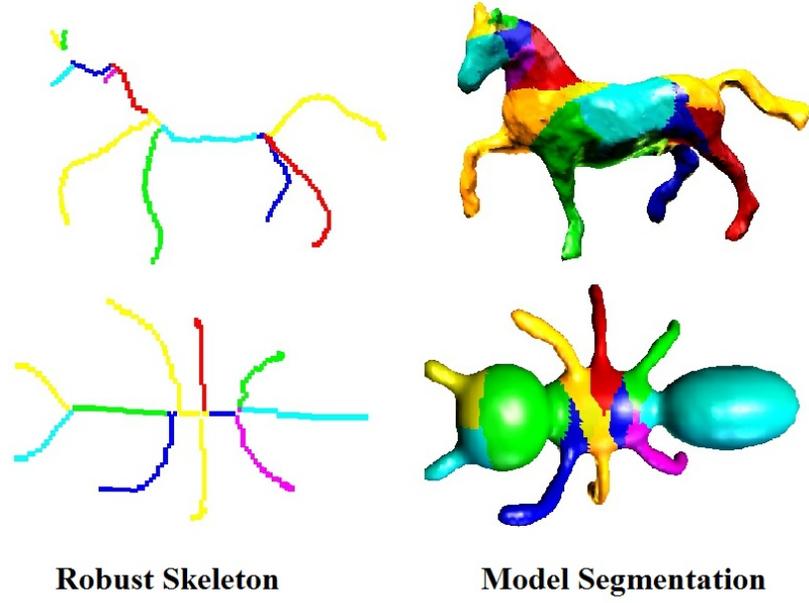


Figure 3.13: Anatomy preserving model decomposition.

to generate more precise boundary of the segments.

There are three factors that I will consider in this process between each skeleton vertices  $v_r(i)$  (with category reference  $r$ ) and model surface points  $p(j)$ :

(1) Point distance  $d_p(p(j), v_r(i))$ : the  $L2$  distance between model surface nodes and skeleton vertices:

(2) Normal vector  $n_i$  from skeleton point  $v_r(i)$  to surface points  $p(j)$ , in comparison with the local surface normal vector  $n_j$ . The distance is defined as:

$$d_n(p(j), v_r(i)) = \frac{\|n_i\| \times \|n_j\|}{n_i \cdot n_j} \quad (3.7)$$

(3) Local curvature should be consistent to keep the segments uniform.

In order to boost the efficiency of the computation, I randomly select 20% sample points into a set  $P_j$ , which are from all the surface nodes to be

mapped. For each  $p(j) \in Pj$ , I compute its distances against all the skeleton vertices  $v_r(i)$ :

$$d_r(p(j), v_r(i)) = d_n(p(j), v_r(i)) \times d_p(p(j), v_r(i)) \quad (3.8)$$

Thus  $p(j)$  is mapped to category  $r$  if the distance between  $p(j)$  and  $v_r(i)$  is minimum:

$$\operatorname{argmin}_i d_r(p(j), v_r(i)) := \{p(j) \Rightarrow \text{Set } r | p(j) \in Pj\} \quad (3.9)$$

More specifically, the standard to map one point on the model surface is that it should be close to the skeleton branch nodes, with skeleton-to-surface normal vector pointing the similar direction with the surface normal vectors, also the neighboring nodes should be with the same label according to curvature similarity.

After labeling the selected sample points, I adopted the idea of watershed method [56] in image segmentation to “flood” the label from samples. Defined on surface nodes, the labeled sample nodes are regarded as water source in each regional minimum. The boundary is build when different sources meet. The un-labeled nodes are categorized according to those neighbor nodes which are already labeled. If a collision appears (two contradictory labels), I choose the one with closer surface normal. And results are modified by refining the boundary with local mean-shift curvature consistency.

### 3.5 Summary

In this research, I refine the thinning skeleton results by integrating Scale-Space-Filtering and node significance. After the smoothing and pruning processing steps, the resulting curve skeleton is more robust to noise. Thus the improved skeletons eliminate both global structural noise of extra branches as well as the high frequency local jitter noises.

Next, I adopt the topology matching and chain coding techniques for measuring model similarity. This algorithm can generate more accurate topological and geometrical representation compared to classical thinning algorithms, with particular emphasize on distinguishing different poses of similar models.

Finally, I present an effective model decomposition technique with consideration of the model topology. By extracting unit-width curve skeletons, which are robust to noise, and mapping skeleton branches to model surface nodes, this method accurately identifies the topology and geometry information of a 3D model, resulting in more semantically rich segmented components.

# Chapter 4

## Experiment and Results

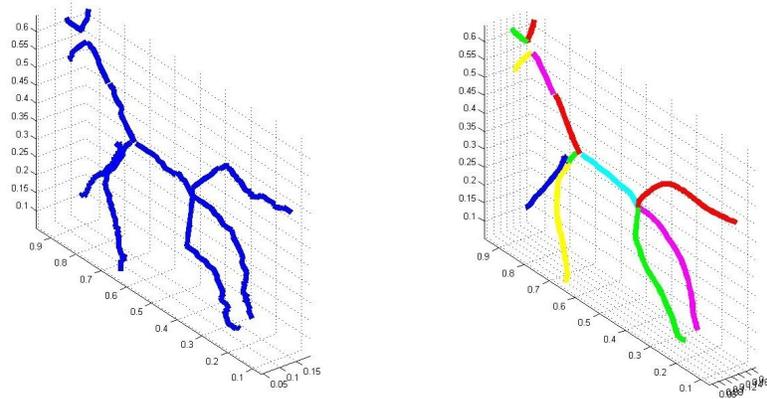
### 4.1 Results of Refined Skeletonization

In the experiments, I used 1814 3D models from the Princeton Shape Benchmark [51] and 380 models from the data set of 3D model segmentation Benchmark [12]. The model format is Object File Format (.off) with polygonal geometry. Skeletons generated from fully parallel thinning [58] and from the proposed method were compared.

#### 4.1.1 Adaptive SSF-Refined Skeletonization

Using the algorithm discussed in Section 3.2, the skeletons of 2194 models are generated to compare with the thinning results. As shown in Figure 4.1 and Figure 4.2, the skeletons from the proposed method are more stable despite the size, level-of-detail and the model pose changes. It can be seen that the skeleton generated from my algorithm are more robust to noise both locally as well as globally. The sub-branches at junctions and points with large articulation are removed (Figure 4.3). Overall, my approach produces a better representation of both the topology and the geometric structure of the models.

Our method improves the result making it noise-free and comparable to recent approaches [15] [19] [46] [59], as shown in Figure 4.4. The computed curve skeletons possess the following features, defined in [13], which are widely analyzed and agreed upon:



**Original Skeleton Results**      **Scale-Space Filtered Results**

Figure 4.1: Comparison between original and filtered results.

- (1) Homotopy: Ensured by thinning for the reason it only removes the voxels that do not alter topology;
- (2) Topology preservation: Since the computation is based on each sub-branch, the topology of skeleton is not sensitive to object orientation;
- (3) Thinness (1D): The VDSM thinning guarantees the thinness of the results;
- (4) Centeredness: After smoothing, the skeleton is better centered within the model;
- (5) Junction handling: The approach is able to distinguish different junctions of the original object, reflecting its part or component structure. This implies that the logical components of the object should have a one-to-one correspondence with the logical components of the skeleton;
- (6) Connectivity: The connectivity is maintained by the characteristics of thinning;
- (7) Robustness: The results are more robust to noise in 3D models. It can provide match-able topological structures between component-wise differentiated objects;
- (8) Smoothness: Adaptively controlled by the filter size and is thus flexible to ensure smoothness on each branch; and,

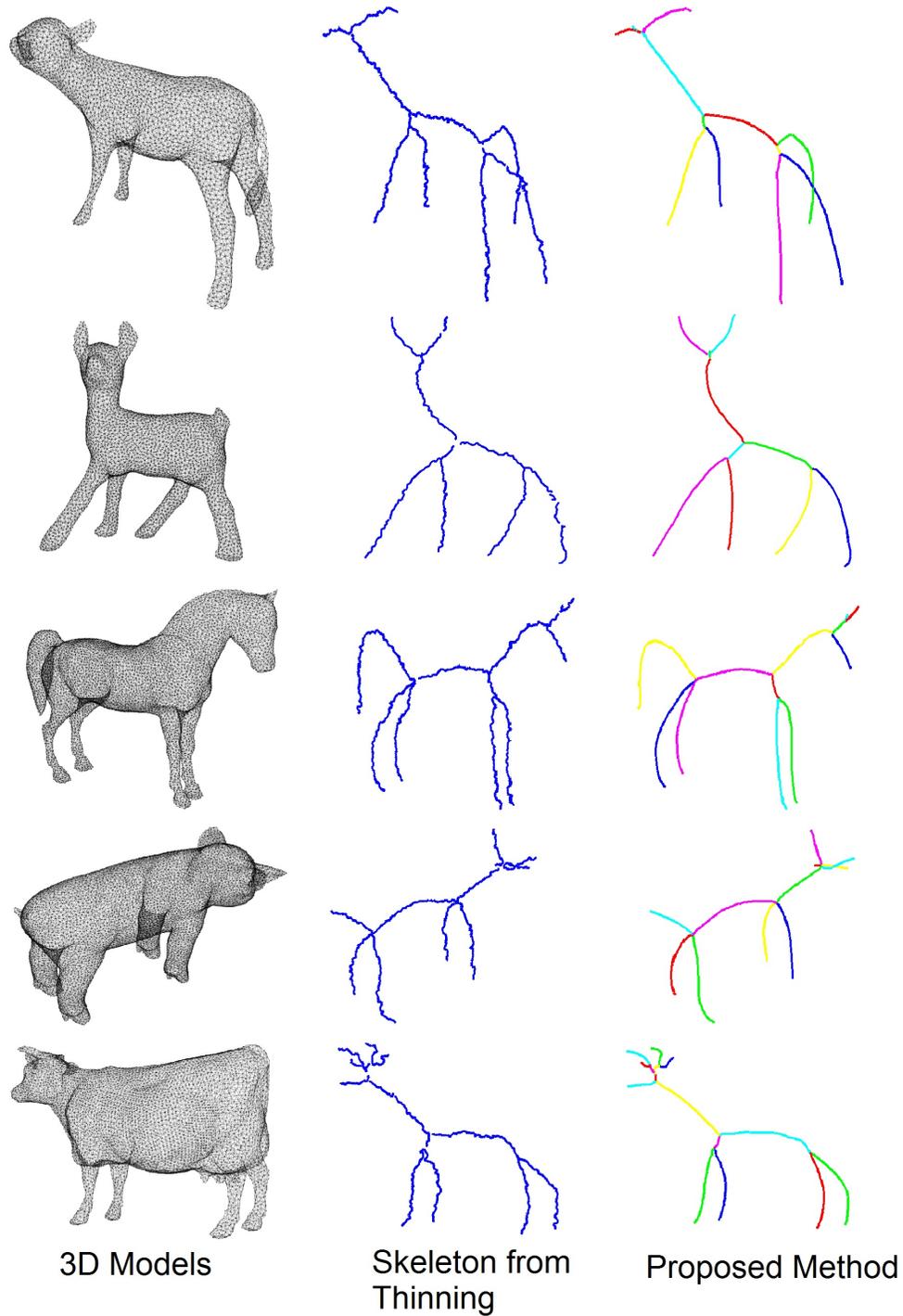


Figure 4.2: Comparison of the proposed method (right) with the thinning method (middle). It can be seen that the proposed method eliminates the jitter effects generated from thinning, producing a smoother skeleton. Also, the sub-branch noises are removed, which can improve the accuracy in topological matching.

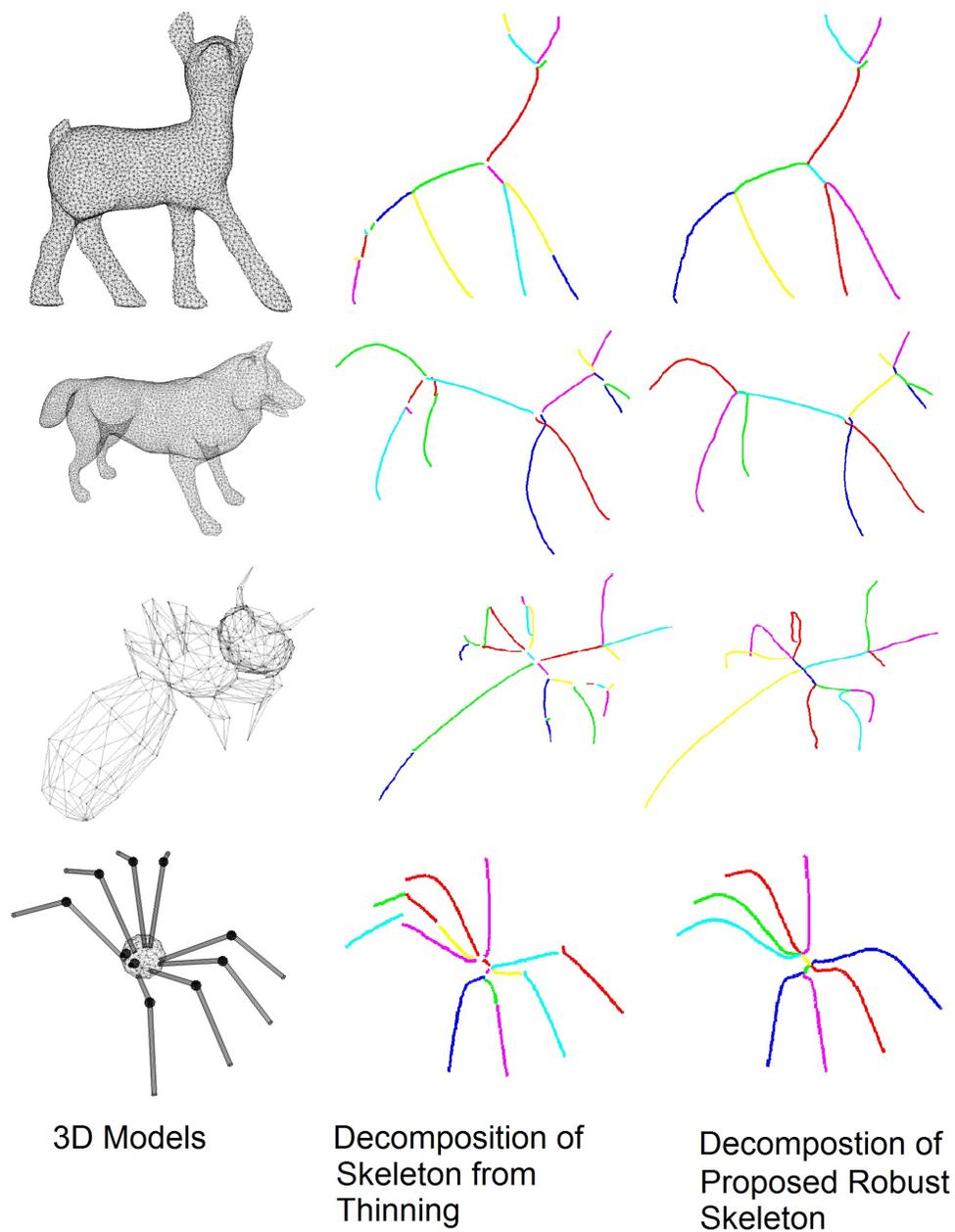


Figure 4.3: Decomposed skeleton pruning results. Left is the 3D mesh model. Middle is the original skeleton from thinning [58]. Right is the filtered skeleton with my proposed pruning method. Different color indicates decomposed skeleton branches. The extra branches at the legs of each model are removed and the total number of sub branches is decreased.

(9) Efficiency: Refined thinning is more efficient than other approaches, which is essential for applications requiring fast model computations and matching.

These properties of this framework are in tune with the criterion for ideal skeletonization in Section 2. Skeletons from the proposed method are more stable despite size, level-of-detail and pose changes in 3D models. It can be seen that the skeletons generated by the proposed algorithm are more robust to noise both locally as well as globally. The sub-branches at junctions and points with large articulation are removed (Figure 4.3) and jitter noise is eliminated generating smooth and well-centered output. Overall, my approach produces a better representation of both topology and geometric structures of the models. More results can be seen in Figure 4.4.

#### 4.1.2 Comparative Study on Time Performance

To demonstrate the efficiency of the proposed algorithm, I compare the running time to four methods, including two classic schemes of potential field [15] and medial geodesic function [19], along with other two recent approaches of Reniers’s hierarchical extraction [46] and Wang’s Iterative Least Squares Optimization method [59]. The topology representation and smoothness of skeleton results are comparable between this refined thinning and these three approaches. Nevertheless, the running time of this method is significantly shorter than its counterparts. Table 4.1 shows the comparison of time performance (in log scale) for 10 typical models.

It is clear from the time expenses that the proposed framework is more efficient than others, shown with Log(time) scale in Figure 4.5. In Table 4.1, for ten typical models of plane, ant, hand, armadillo, dog, cow, horse, octopus, rabbit and bird, the number of voxels (No.V) vary from 11140 to 85100, the proposed method takes less than five seconds for all the models; while others need much more time, even hours in some cases. For instance, on the bird model, Dey’s medial geodesic function [19] uses more than three hours(11523 s), Cornea’s [15] and Reniers’s [46] method use about three

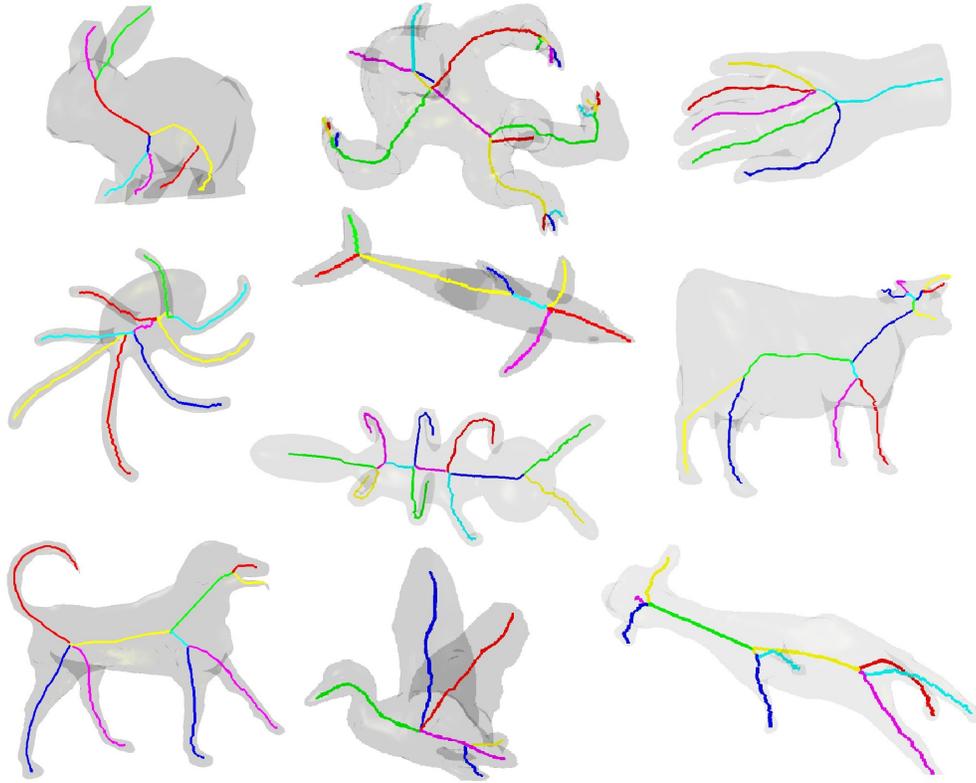


Figure 4.4: Skeletonization and skeleton decomposition results. The color indicates topological segmentation of each skeleton. It can be seen that my results are free of both global noise (extra sub-branches) as well as local noise (jitter effects on each branch). Therefore it conserves more fidelity and represent the topology and geometry structure of the 3D model superiorly.

Table 4.1: Computation times Log(seconds) of skeletonization on a 2.6 GHz Intel Due core PC with 4GHz RAM.

Models	No.V	Running time(s)				
		Dey06	Cornea07	Renier08	Wang08	Proposed
Plane	11140	363	17.5	4.9	2.8	0.8
Ant	15087	531	19.9	7.8	3.6	1.3
Hand	16725	547	23.4	9.2	4.4	1.7
Armadillo	25491	625	39	9.3	7.4	3.2
Dog	32000	892	44	11.2	7.9	2.8
Cow	42384	2221	56	28	8.4	3.5
Horse	52058	2580	125.9	32	10.7	4.3
Octopus	76218	7824	263	54	12.7	3.6
Rabbit	82154	11100	199.53	174	15.2	2.4
Bird	85100	11523	214.7	183	17.3	3.2

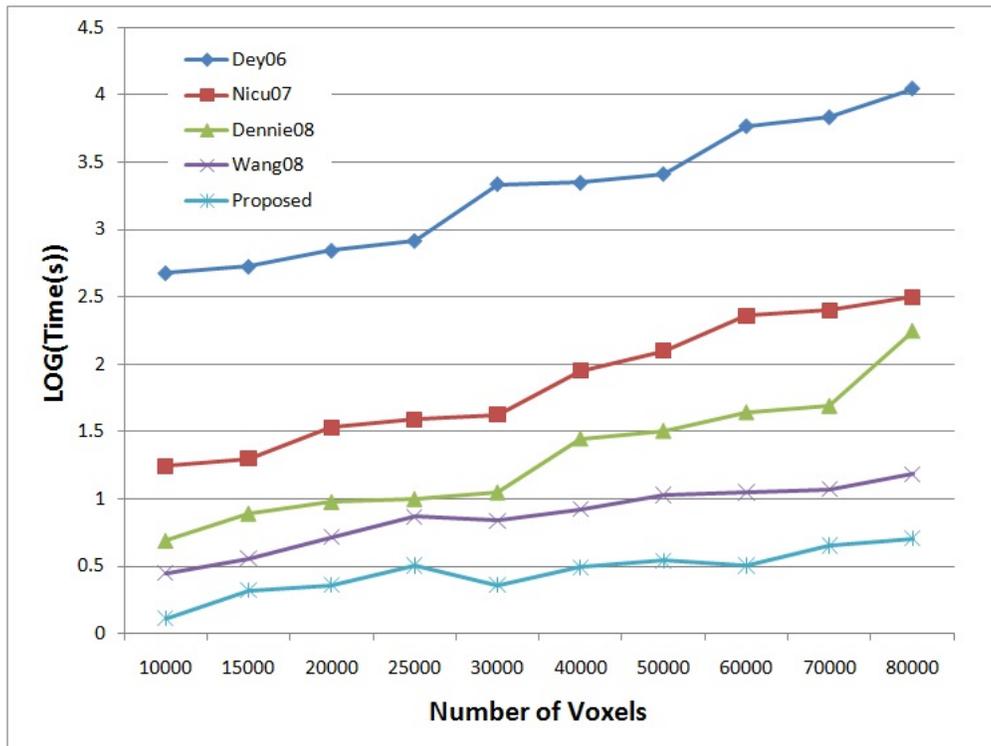


Figure 4.5: Running time comparison of five approaches.

minutes (214.7 s and 183 s), Wang's [59] approach needs 17.3 seconds, while my algorithm needs only 3.2 seconds, significantly improving the time efficiency. From the curves in Figure 4.5, it can be seen that the time expenses for thinning does not increase proportionally; but, other approaches are largely influenced by model dimension and complexity. This feature of thinning is required to maintain continuity and consistency in 3D applications.

In interactive deformation of 3D objects, skeletons are essential in the sense of leading the kinematic movement of each subdivision. The topologically segmented skeletons provide a promising potential to manipulate objects by controlling each branch of the deformable model, distributing the movements to other divisions from the junctions. It preserves the geometric features better than original thinning, and is more effective for computations leading to real-time animations.

As illustrated in Figure 3.10, the user can select branches of the skeleton to control the model. Since topological matching is achievable by skeleton decomposition, the user can simply create template movement for similar skeletons, thus generating similar motion for a diverse set of animal or object models.

## **4.2 Model Matching and Retrieval with Chain Coding**

After the SSF Refinement of the skeletons, I encoded them with chain codes for model matching. In this experiment, I used 10 types of models from the data set of 3D decomposition Benchmark [12], including Human, Airplane, Ant, Octopus, Fish, Table, Teddy, Hand, Armadillo and Fourleg, with 200 models in total.

Due to the smoothed and pruned effect on the refined skeletons, the average length of the chain codes was decreased by 23% compared to the original skeletons from thinning. I measured the similarity between each model by computing the pair-wise distance of each two models. Generally,

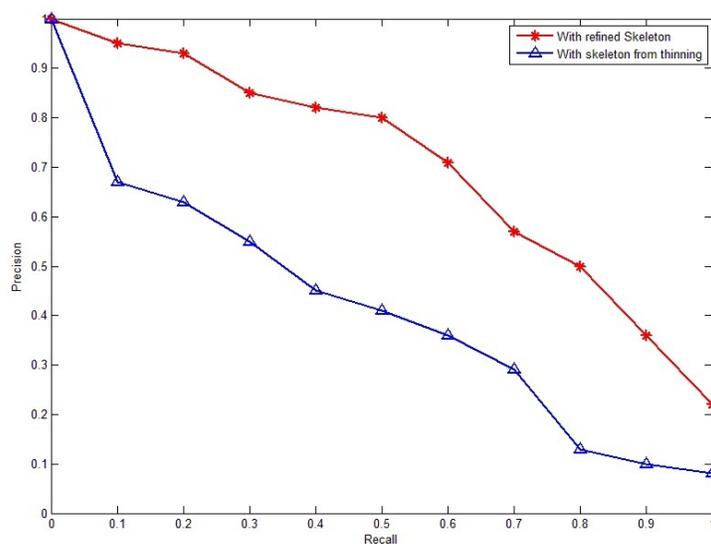


Figure 4.6: Average precision-recall curve of the 10 types models. The axes stand for normalized units of precision and recall. Blue curve represents the result from original skeleton by thinning, and the red curve shows the result of refined skeleton.

given one input sample model, the performance of the retrieval engine is measured by the precision-recall (PR) curve. If the returned model belongs to the same type of the input model, it is regarded as a correct return. Based on the pair-wise distance, the average PR curves of my approach and the thinning method are shown in Figure 4.6. Both axes stand for the normalized unit of precision (among the retrieved models, how many are correct) and recall (among all the model which should be matched, how many are found). The accuracy of this retrieval system is better due to more precise topology and less local jitter noise of the skeletons. Notice that the similar models within the same type are not identical with variations in position and pose. The ranking of returned models are dependent on the similarity in poses. In other words, the models with identical or comparable poses are with higher ranking, which is useful in pose recognition.

In order to demonstrate the discrimination of model poses, I selected five models from the Armadillo type, including various poses of hug, jump, hit,

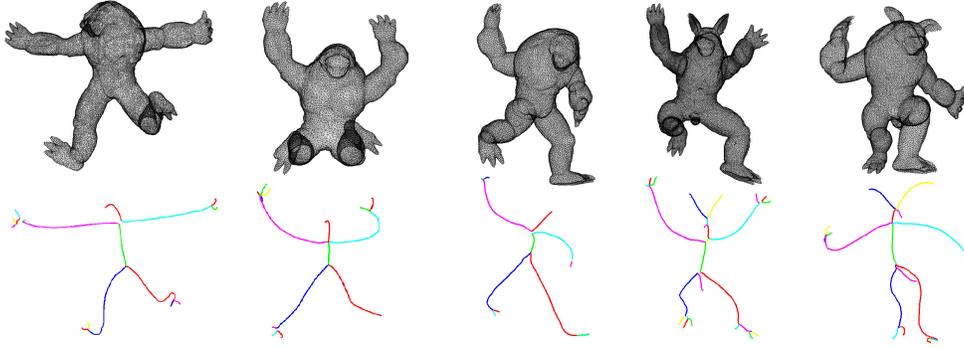


Figure 4.7: Five poses of the Armadillo model in two versions. The first three do not have ears and the tail. The poses of each model are (from left to right): Hug, jump, hit, hop and dance. The distances between their chain codes are shown in Table 4.2.

hop and dance (See Figure 4.7). I first employed the topology matching of the skeleton from Sundar et al. [26]. Results are marked with corresponding colors in the skeletons. I compute the distances between the models by generating their chain codes of the SSF-refined skeleton branches. The distances between chain codes are shown in Table 4.2, from which we can see that the topological matching is the dominating factor in the computation of chain code distance. Thus different models without similar topology will have the distance larger than 0.38; while the models with similar topology have distance scores under 0.20. At the same time, the distance scores greater than 0.06 and less than 0.19 suggest similar models with different poses. Therefore, the topology matching is dominating; in compensation, it can also identify the pose discrepancy. To notice in Table 4.2, the score between hop dance and other three models are larger than 0.19. The reason for this is these five models are not exactly identical. For the last two there is an pair of ears, thus the graph matching not matched for these two branches, which raised the dissimilarity scores.

Table 4.2: Distance of chain code on two types of the Armadillo model with 5 poses

	Hug	Jump	Hit	Hop	Dance
Hug	0.0	0.15	0.17	0.43	0.45
Jump		0.0	0.10	0.38	0.42
Hit			0.0	0.42	0.32
Hop				0.0	0.20
Dance					0.0

### 4.3 Results of 3D Model Decomposition

The results of model decomposition are generated from the topology matching between skeleton and the 3D model, considering the model curvature as well as surface normal vector. Results can be seen in Figure 4.8.

I compare with 3 algorithms: Random Walks (RW) [29], Shape Diameter Function (DF) [49] and Randomized Cuts (RC) [22]. 380 meshes across 19 object categories are from the Princeton benchmark for 3D mesh decomposition [12], including Human, Cup, Glasses, Airplane, Ant, Chair, Octopus, Table, Teddy, Hand, Plier, Fish, Bird, Armadillo, Bust, Mech, Bearing, Vase and Fourleg. Comparison results are shown in Figure 4.9. From the results we can see that my topological mapping methods reflect the semantic structure clearer than other approaches.

In order to generate natural movement of the models, the kinematic animations are highly dependent on the topological structure or the 3D models. For instance the four-leg animals, as a typical model in most animations, are more feasible to be segmented based on animal anatomy. As shown in Figure 4.4 and Figure 4.10, I achieved a more accurate match between the animal anatomy on a horse [11] and the decomposition results. I evaluate the results of “horse” model in Figure 4.10 by using a metric function distance comparing to other three algorithms. Firstly, I manually register the segmentation cuts (the closed line separating two model segmentations) to the ground truth of animal anatomy by pairing the nearest cuts. Then error of automatic segmentation is defined as the Euclidian distance between

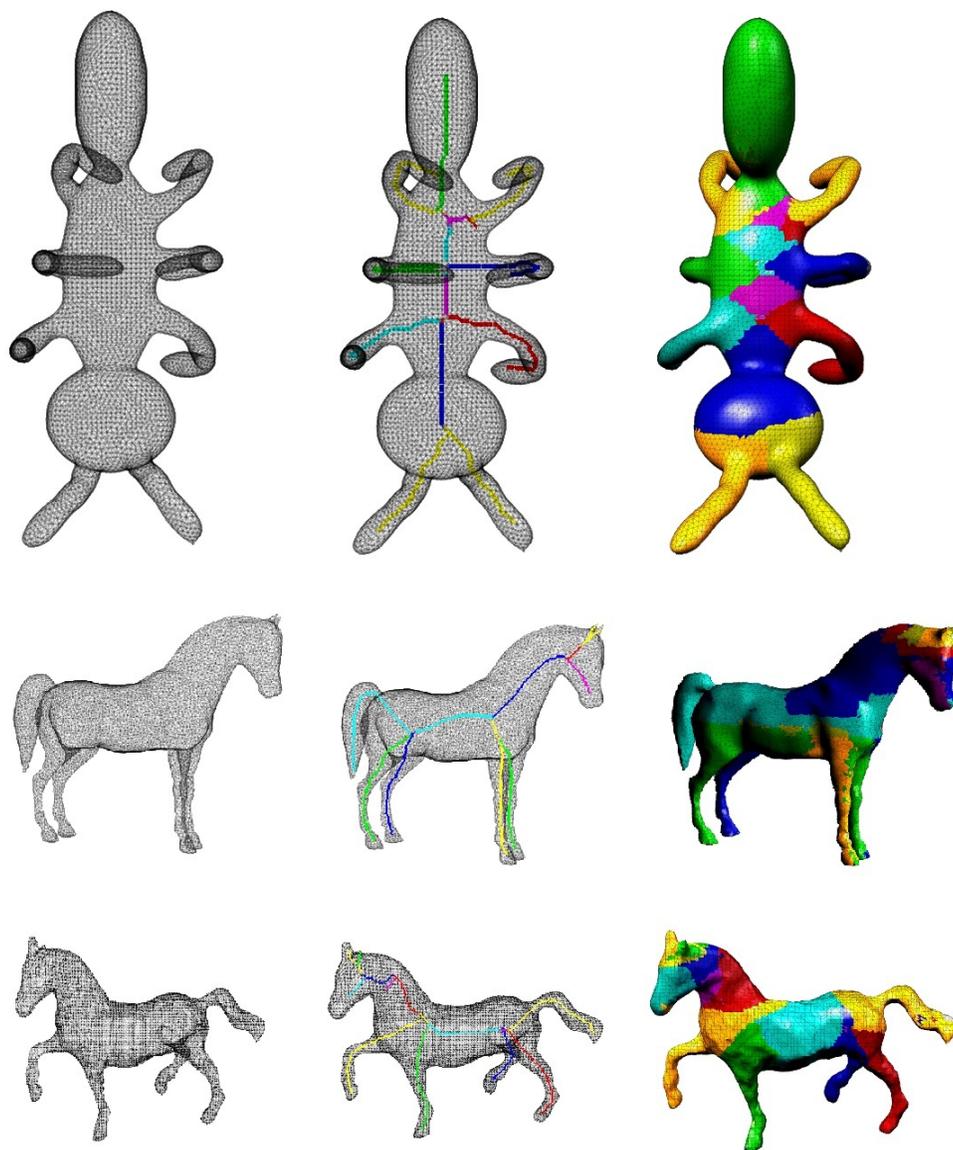


Figure 4.8: Results of model decomposition. The left column is the original model; the middle column is the model with its extracted skeleton; and the right column is the decomposition result.

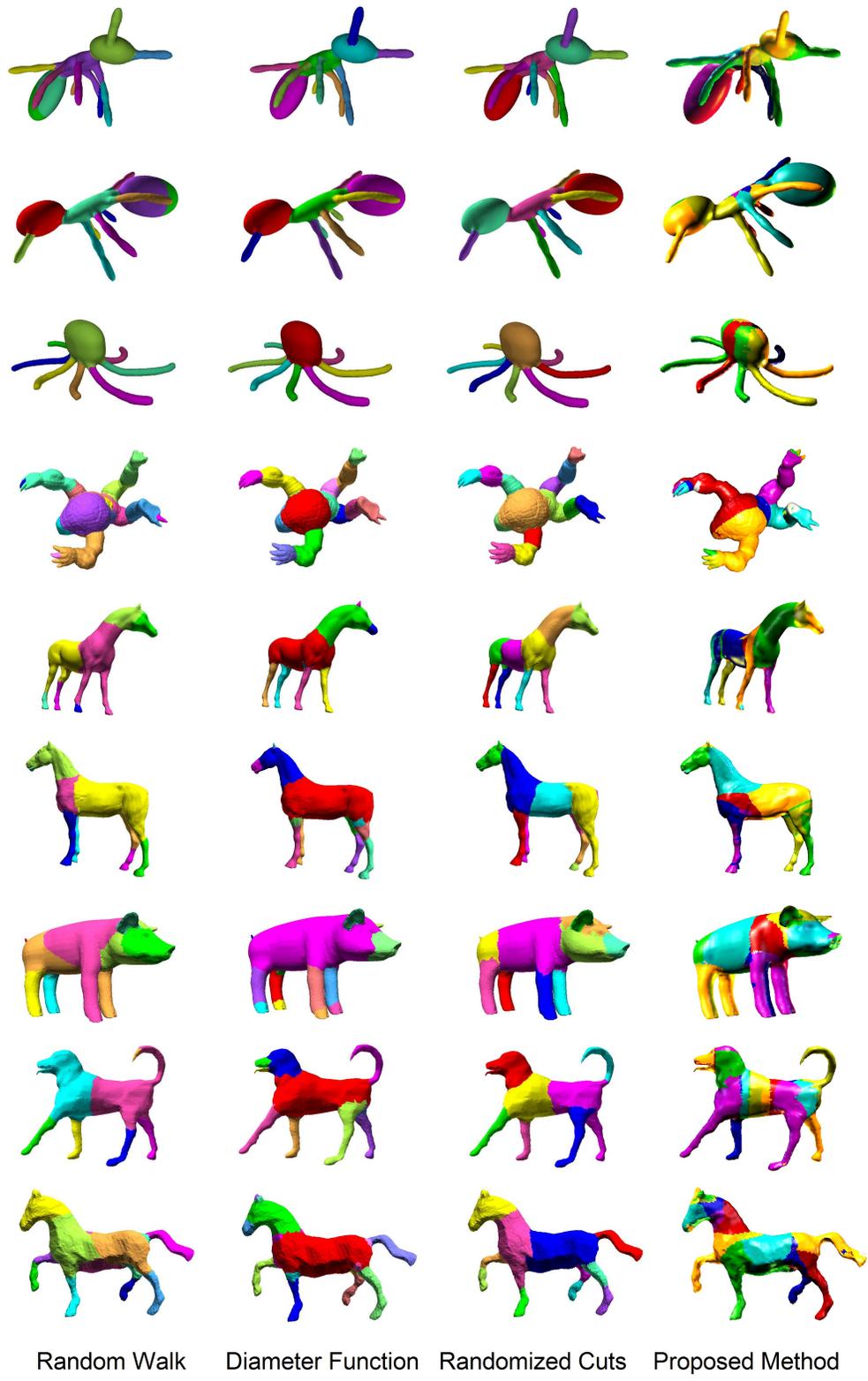


Figure 4.9: Comparison decomposition results with Random walk, shape diameter function and randomized cuts.

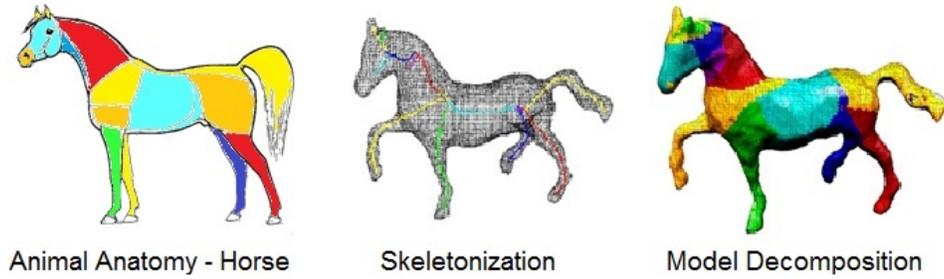


Figure 4.10: Mapping between animal anatomy decomposition result.

Table 4.3: Metric distance on anatomy of horse model.

Methods	RW	DF	RC	Proposed
No. of Segments	11	16	11	11
Metric Distance	1	0.79	0.84	0.45

position of nodes within the registered pair of cuts, plus the length of non-paired cuts (additional cuts without a counterpart in the ground truth). At last I normalize the distance within 1 and 0.

Results are shown in Table 4.3: the lower the metric distance, the closer it's to the anatomy criterion of segmentation. The topological mapping from the decomposed skeleton provides more semantic information of the model which meets better the requirements in animation and model manipulation.

# Chapter 5

## Conclusion

In this thesis, I introduced an effective skeletonization refinement technique based on Scale-Space-Filtering and Node Significance to remove undesirable branches and noises from 3D curve skeletons, and thus it improves the performance of 3D models matching and retrieval. I applied an adaptive scale parameter in the pruning and smoothing processes. By transforming the filtered skeleton into the chain code representation, this technique presents a better descriptor of the topology and geometric structure for 3D models. Also it provides a balance between computational time and accuracy.

This method is demonstrated by the experiments, using over 2000 models, to be effective in distinguishing similar models with different poses, with the capability to classify models using both skeleton graphs and curvature deviations in sub-branches. While the skeleton graph approach is a more stable tool for the abstraction of complex models with arbitrary topology, the chain code metric embraces more discrimination ability in pose recognition. Compared to the skeletons directly generated from thinning, my results prove to be more robust to noise and can perform better in chain-coding for 3D model matching and retrieval. Finally, with the noise-free skeleton results, I utilized topology mapping to improve the performance of 3D models decomposition. From mapping the sample surface nodes to decomposed skeleton branches, I achieved efficient model segmentation by applying watershed flooding algorithm. Results represent the topological

fidelity with the capability to match the global and local topology information with skeletonial presentation.

## 5.1 Contributions

The contribution of this thesis can be summarized as follows:

- Propose an adaptive skeletonization refinement method using Scale-Space representation, which effectively reduces noises in preparation for the subsequent model mapping and labeling process.
- Introduce the significance of junction node for pruning, which effectively reduces noises in preparation for the subsequent chain coding step.
- Enable the discrimination of similar models with different poses in model retrieval in addition to the topological matching, which is useful for applications requiring pose detection.
- Utilize the topology and geometry features for effective mesh model decomposition, adding more semantics fidelity.
- Provide an efficient mapping scheme from labeled sample nodes to the model surface points with watershed flooding algorithm.

## 5.2 Discussion

This method is limited in the case of models without obvious topological layout. For example, the flat convex and concave polygons model without any branches. In this case, the curve skeleton also does not fit well in the task of model retrieval and decomposition. The intrinsic advantage of skeletonization is to distinguish the models by its outline and shape information, thus if this key assumption is missing, this framework will also be limited on these models.

### 5.3 Future Research

In future work, I will extend this technique to the discrimination and measurement of more complex 3D structures, e.g. human anatomical structures such as veins, which require high precision. Also, due to the focus on semantic and pose discrimination, I will look into the application of animation from model decomposition with its skeleton counterparts. The potential advantage will be a natural mapping between the model surface and the manipulating junction and axes.

# Bibliography

- [1] Description of shape information for 2-d and 3-d objects. *Signal Processing: Image Communication*, 16(1-2):103–122, 2000.
- [2] Mihael Ankerst, Gabi Kastentmller, Hans-Peter Kriegel, and Thomas Seidl. 3D shape histograms for similarity search and classification in spatial databases. In *SSD '99: Proceedings of the 6th International Symposium on Advances in Spatial Databases*, pages 207–226, 1999.
- [3] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal. Mesh segmentation - a comparative study. In *SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006*, page 7, Washington, DC, USA, 2006. IEEE Computer Society.
- [4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.
- [5] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:167–192, 1988.
- [6] S. Biasotti, B. Falcidieno, and M. Spagnuolo. Extended reeb graphs for surface understanding and description. *International Conference on Discrete Geometry for Computer Imagery*, page 185, 2000.
- [7] S Biasotti, S Marini, M Mortara, and G Patane. An overview of properties and efficacy of topological skeletons in shape modelling. *Shape modeling international*, pages 245–254, 2003.
- [8] Philip Bille. A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337(1-3):217–239, 2005.
- [9] J.W. Brandt and V.R. Alazi. Continuous skeleton computation by voronoi diagram. *Computer Vision, Graphics and Image Processing: Image Understanding*, 55:329–338, 1992.
- [10] Bribiesca and Ernesto. A method for representing 3D tree objects using chain coding. *Journal of Visual Communication and Image Representation*, 19(3):184–198, 2008.

- [11] Klaus-Dieter Budras, Wolfgang O. Sack, Sabine Rck, Anita Wnsche, and Ekkehard Henschel. *Anatomy of the Horse: An Illustrated Text, 3rd ed.* Iowa State University Press, Ames, Iowa, 2001.
- [12] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3):1, August 2009.
- [13] N.D. Cornea. Curve-skeletons: Properties, computation and applications. *Ph.D. thesis Rutgers University*, 2007.
- [14] N.D. Cornea, D. Silver, X. Yuan, and R. Balasubramanian. Computing hierarchical curve-skeletons of 3D objects. *The Visual Computer*, 21(11):945–955, 2005.
- [15] Nicu D. Cornea, Deborah Silver, and Patrick Min. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):530–548, 2007.
- [16] C. M. Cyr and B. B. Kimia. A similarity-based aspect-graph approach to 3D object recognition. *Int. Journal of Computer Vision*, pages 57(1):5–22, April 2004.
- [17] Petros Daras, Dimitrios Zarpalas, Dimitrios Tzovaras, and Michael G. Strintzis. Shape matching using the 3d radon transform. *3D Data Processing Visualization and Transmission, International Symposium on*, 0:953–960, 2004.
- [18] M. Fatih Demirci, Ali Shokoufandeh, Sven Dickinson, Yakov Keselman, and Lars Bretzner. Many-to-many feature matching using spherical coding of directed graphs. *European Conference on Computer Vision, Prague, Czech Republic*, pages 332–335, 2004.
- [19] Tamal K. Dey and Jian Sun. Defining and computing curve-skeletons with medial geodesic function. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 143–152. Eurographics Association, 2006.
- [20] H. Freeman. Computer processing of line drawing images. *ACM Computing Surveys*, pages 57–97, 1974.
- [21] Thomas Funkhouser, Michael Kazhdan, Patrick Min, and Philip Shilane. Shape-based retrieval and analysis of 3d models. *Communication ACM*, 48:58–64, June 2005.
- [22] A. Golovinskiy and T. Funkhouser. Randomized cuts for 3D mesh analysis. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA)*, 27(5), December 2008.
- [23] W. Gong and G. Bertrand. A simple parallel 3D thinning algorithm. *IEEE Int. Conf. on Pattern Recognition*, pages 188–190, 1990.

- [24] M. Sabry Hassouna and A.A. Farag. Robust centerline extraction framework using level sets. *Computer Vision and Pattern Recognition*, pages 458–465, 2005.
- [25] Wim H. Hesselink and Jos B.T.M. Roerdink. Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:2204–2217, 2008.
- [26] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Toshiyasu L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 203–212, 2001.
- [27] Keitaro Kaku, Yoshihiro Okada, and Koichi Nijjima. Similarity measure based on obbtree for 3d model search. In *Proceedings of the International Conference on Computer Graphics, Imaging and Visualization*, pages 46–51, Washington, DC, USA, 2004. IEEE Computer Society.
- [28] Marcel Körtgen, G. J. Park, Marcin Novotni, and Reinhard Klein. 3d shape matching with 3d shape contexts. In *The 7th Central European Seminar on Computer Graphics*, April 2003.
- [29] Y. K. Lai, S. M. Hu, R. R. Martin, and P. L. Rosin. Fast mesh segmentation using random walks. In *SPM '08: Proceedings of the 2008 ACM symposium on Solid and physical modeling*, pages 183–191, New York, NY, USA, 2008. ACM.
- [30] F.F. Leymarie. 3D shape representation via shock flows. *Ph.D. thesis Brown University*, May 2003.
- [31] Frederic F. Leymarie and Benjamin B. Kimia. From the infinitely large to the infinitely small - applications of medial symmetry representations of shape. *Computational Imaging and Vision*, 37:327–351, 2008.
- [32] Tony Lindeberg. Scale-space. In: *Encyclopedia of Computer Science and Engineering (Benjamin Wah, ed)*, 5:2495–2504, January 2009.
- [33] C. Lohou and G. Bertrand. New fully parallel thinning algorithms for 2d and 3D binary images based on p-simple points. *SPIE Vision Geometry IX*, 2002.
- [34] Christophe Lohou. Detection of the non-topology preservation of ma and sonka’s algorithm, by the use of p-simple points. *Computer Vision and Image Understanding*, 114(3):384–399, 2010.
- [35] C.M. Ma and M. Sonka. A fully parallel 3D thinning algorithm and its applications. *Computer Vision and Image Understanding*, 64(3):420–433, 1996.
- [36] C.M. Ma, S.Y. Wan, and J.D. Lee. Three-dimensional topology preserving reduction on the 4-subfields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 24(12):1594, 2002.

- [37] W.C. Ma, F.C. Wu, and M. Ouhyoung. Skeleton extraction of 3D objects with radial basis functions. In *Proceedings of Shape Modeling International*, pages 207–215, 2003.
- [38] P Min, JA Halderman, M Kazhdan, and TA Funkhouser. Early experiences with a 3D model search engine. *Web 3D symposium*, pages 7–18, 2003.
- [39] D.G. Morgenthaler. Three-dimensional simple points: serial erosion, parallel thinning and skeletonization. *TR- 1005, Computer Vision Lab., Univ. of Maryland*, 1981.
- [40] Ryutarou Ohbuchi, Takahiro Minamitani, and Tsuyoshi Takei. Shape-similarity search of 3d models by using enhanced shape functions. In *TPCG '03 Proceedings of the Theory and Practice of Computer Graphics*, pages 97–104, 2003.
- [41] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21:807–832, 2002.
- [42] D. L. Page, A. F. Koschan, J. K. Paik, and M. A. Abid. Shape analysis algorithm based on information theory. In *Proceedings of the International Conference on Image Processing*, I:229–232, 2003.
- [43] E Paquet, A Murching, T Naveen, A Tabatabai, and M Rioux. Description of shape information for 2-d and 3-d objects. *Signal Processing: Image Communication*, 16:103–122, 2000.
- [44] Marcello Pelillo, Kaleem Siddiqi, and Steven W. Zucker. Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, November 1999.
- [45] S.M. Pizer, D. Eberly, and D.S. Fritsch. Zoom-invariant vision of figural shape: the mathematics of cores. *Computer Vision and Image Understanding*, 69:55, 1998.
- [46] Dennie Reniers, Jarke van Wijk, and Alexandru Telea. Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):355–368, 2008.
- [47] Diane C. Rogers-Ramachandran and V. S. Ramachandran. Psychophysical evidence for boundary and surface systems in human vision. *Vision Research*, 38(1):71–77, January 1998.
- [48] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Int. Conf. on Computer Vision*, pages 755–762, 2001.
- [49] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput.*, 24(4):249–259, 2008.

- [50] Yu-Te Shen, Ding-Yun Chen, Xiao-Pei Tian, and Ming Ouhyoung. 3D model search engine based on lightfield descriptors. *Eurographics, Granada, Spain*, September 2003.
- [51] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The princeton shape benchmark. *Shape Modeling International, Genova, Italy*, pages 167–178, June 2004.
- [52] H Sundar, D Silver, N Gagvani, and S Dickenson. Skeleton based shape matching and retrieval. *Shape modeling international*, pages 130–139, 2004.
- [53] Johan W. Tangelder and Remco Veltkamp. A survey of content based 3D shape retrieval methods. *Springer Science and Business Media, LLC*, December 2007.
- [54] Johan W. H. Tangelder and Remco C. Veltkamp. Polyhedral model retrieval using weighted point sets. *International Journal of Image and Graphics*, 3:209–229, 2002.
- [55] Miguel Vieira and Kenji Shimada. K.: Surface mesh segmentation and smooth surface extraction through region growing, computer-aided geometric design. *Computer Aided Geometric Design*, 22:771–792, 2005.
- [56] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.
- [57] D. V. Vranic and D. Saupe. 3d model retrieval. In: *Proceedings of the Spring Conference on Computer Graphics and its Applications (SCCG2000)* (editor B. Falcidieno), Budmerice, Slovakia, pages 89–93, May 2000.
- [58] T. Wang, I. Cheng, V. Lopez, E. Bribiesca, and A. Basu. Valence normalized spatial median for skeletonization and matching. *IEEE Workshop on Search in 3D and Video (S3DV)*, pages 55–62, 2009.
- [59] Y. S. Wang and T. Y. Lee. Curve-skeleton extraction using iterative least squares optimization. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):926–936, 2008.
- [60] F.C. Wu, W.C. Ma, P.C. Liou, R.H. Laing, and M. Ouhyoung. Skeleton extraction of 3D objects with visible repulsive force. *Eurographics Symposium on Geometry Processing*, pages 1–8, 2003.
- [61] T. Zaharia and F. Preteux. Hough transform-based 3d mesh retrieval. In *Proceedings of SPIE 4476 on Vision Geometry X, San Diego, USA*, 4476:175–185, August 2001.