

UNIVERSITY OF THE AEGEAN

Department of Product and Systems Design Engineering



**Part based 3D representation for the retrieval
of 3D graphical models**

**3Δ Αναπαράσταση βασισμένη σε κατάτμηση για την
αναζήτηση και ανάκτηση 3Δ γραφικών μοντέλων**

PhD Dissertation

Alexander G. Agathos

ΣΥΡΟΣ

ΝΟΕΜΒΡΙΟΣ 2009

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

3Δ Αναπαράσταση βασισμένη σε κατάτμηση για την αναζήτηση και ανάκτηση 3Δ γραφικών μοντέλων

Αλέξανδρος Αγάθος

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Νικόλαος Σαπίδης

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:

Νικόλαος Σαπίδης, Καθηγητής, Πανεπιστήμιο Αιγαίου

Σταύρος Περαντώνης, Ερευνητής Α' ΕΚΕΦΕ Δημοκρίτος

Φίλιππος Αζαριάδης, Επίκουρος Καθηγητής, Πανεπιστήμιο Αιγαίου

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

- 1. Νικόλαος Σαπίδης** (επιβλέπων), Καθηγητής Τμήματος ΜΣΠΣ, Πανεπιστήμιο Αιγαίου.
- 2. Σταύρος Περαντώνης** (μέλος τριμελούς συμβουλευτικής επιτροπής), Ερευνητής Α', Ινστιτούτο Πληροφορικής & Τηλεπ/νίων, ΕΚΕΦΕ "Δημόκριτος"
- 3. Φίλιππος Αζαριάδης** (μέλος τριμελούς συμβουλευτικής επιτροπής), Επίκ. Καθηγητής Τμήματος ΜΣΠΣ, Πανεπιστήμιο Αιγαίου.
- 4. Βασίλειος Γάτος**, Ερευνητής Β', Ινστιτούτο Πληροφορικής & Τηλεπ/νίων, ΕΚΕΦΕ "Δημόκριτος".
- 5. Θωμάς Σπύρου**, Επίκ. Καθηγητής Τμήματος ΜΣΠΣ, Πανεπιστήμιο Αιγαίου.
- 6. Δημήτριος Λέκκας**, Λέκτορας Τμήματος ΜΣΠΣ, Πανεπιστήμιο Αιγαίου.
- 7. Σπύρος Βοσινάκης**, Λέκτορας Τμήματος ΜΣΠΣ, Πανεπιστήμιο Αιγαίου.

Περίληψη

Η αναπαράσταση 3D αντικειμένων έχει γίνει αναπόσπαστο μέρος των σύγχρονων εφαρμογών των Γραφικών σε Υπολογιστή, όπως εφαρμογές CAD, ανάπτυξη 3D παιχνιδιών σε υπολογιστή και δημιουργία ταινιών. Επίσης τα 3D δεδομένα έχουν γίνει πολύ κοινά στο πεδίο της Υπολογιστικής Όρασης, Υπολογιστικής Γεωμετρίας, Μοριακής Βιολογίας και Ιατρικής. Η ραγδαία εξέλιξη στην ανάπτυξη υλικού και λογισμικού στον τομέα των γραφικών, ιδιαίτερα στη διαθεσιμότητα χαμηλού κόστους 3D σαρωτών, έχει βοηθήσει ιδιαίτερα στην ανάκτηση, δημιουργία και διαχείριση 3D μοντέλων δίνοντας την ευκαιρία σε μία μεγάλη κοινωνία χρηστών να βιώσουν εφαρμογές που σχετίζονται με 3D μοντέλα. Όσο ο αριθμός των 3D μοντέλων συνεχίζει να αυξάνεται το πρόβλημα της δημιουργίας 3D μοντέλων μετατοπίζεται στο πρόβλημα αναζήτησης ήδη υπαρχόντων 3D μοντέλων. Συνεπώς, η ανάπτυξη αποδοτικών μεθόδων αναζήτησης είναι αναγκαία για την αποδοτική ανάκτηση 3D αντικειμένων από μεγάλες βάσεις δεδομένων.

Σε αυτήν την Διδακτορική Διατριβή παρουσιάζεται μια αναπαράσταση βασισμένη σε γράφημα του 3D αντικειμένου με χρήση ενός καινοτόμου αλγορίθμου κατάτμησης και της ιδέας του Γραφήματος Σχεσιακών Χαρακτηριστικών (Attributed Relational Graph). Η προτεινόμενη αναπαράσταση βασισμένη σε γράφημα θέτει τη βάση για μία αποτελεσματική μεθοδολογία ανάκτησης 3D αρθρωτών αντικειμένων (3D articulated objects).

Η συνεισφορά αυτού του Διδακτορικού είναι δίπτυχη:

Η πρώτη συνεισφορά είναι η κατασκευή ενός αλγορίθμου κατάτμησης του 3D αντικειμένου βασισμένη στην προϋπόθεση ότι το 3D αντικείμενο αποτελείται από ένα κυρίως σώμα (main body) και τα επιμέρους προεξέχοντα τμήματα (protrusible parts). Ο προτεινόμενος αλγόριθμος κατάτμησης στοχεύει στην κατάτμηση του αντικειμένου σε αυτά τα τμήματα. Για να επιτευχθεί αυτός ο στόχος, πρώτα ανιχνεύονται τα προεξέχοντα σημεία (salient points) του αντικειμένου. Αυτά τα σημεία είναι προεξέχοντα χαρακτηριστικά του 3D αντικειμένου που οδηγούν τον αλγόριθμο κατάτμησης. Στην συνέχεια, τα προεξέχοντα αυτά σημεία συγκεντρώνονται σε ομάδες που αναπαριστούν τα κύρια προεξέχοντα τμήματα του αντικειμένου. Στη συνέχεια, το κυρίως σώμα του αντικειμένου προσεγγίζεται χρησιμοποιώντας μονοπάτια ελαχίστου κόστους (minimum cost paths) μεταξύ των προεξέχοντων σημείων του αντικειμένου. Η κύρια ιδέα του προτεινόμενου αλγορίθμου προσέγγισης του κυρίως σώματος είναι η επέκταση ενός

συνόλου σημείων με αύξουσα διάταξη ως προς τη συνάρτηση προεξοχής (protrusion function) μέχρι η επέκταση να καλύψει ένα συγκεκριμένο ποσοστό των σημείων των ελαχίστων μονοπατιών. Η προσέγγιση του κυρίως σώματος είτε καλύπτει μέρος των προεξεχόντων τμημάτων είτε είναι πολύ κοντά στην περιοχή του συνόρου μεταξύ των προεξεχόντων τμημάτων και του κυρίως σώματος. Στη συνέχεια ανιχνεύεται το σύνορο κατάτμησης (partitioning boundary), το οποίο είναι το σύνορο μεταξύ του προεξέχοντος τμήματος και του κυρίως σώματος του αντικειμένου. Θεωρείται ότι στην περιοχή που εσωκλείεται από το επιθυμητό σύνορο μεταξύ του προεξέχοντος τμήματος και του κυρίως σώματος. Συμβαίνει μία απότομη μεταβολή στον όγκο του 3D αντικειμένου, συνεπώς στόχος είναι η ανίχνευση αυτής της μεταβολής. Για να επιτευχθεί αυτό, κατασκευάζονται κλειστές οριακές καμπύλες (closed boundaries) που ορίζονται από μία συνάρτηση απόστασης (distance function) η οποία σχετίζεται με τον αντιπρόσωπο της ομάδας που αναπαριστά το προεξέχον τμήμα. Η απότομη μεταβολή του όγκου ανιχνεύεται εξετάζοντας τις κλειστές οριακές καμπύλες και τίθεται σαν προσέγγιση του συνόρου κατάτμησης η κλειστή οριακή καμπύλη όπου παρατηρείται η μέγιστη μεταβολή περιμέτρου. Η προτεινόμενη προσέγγιση συνόρου κατάτμησης είναι πολύ αποτελεσματική διότι τα πειραματικά αποτελέσματα δείχνουν ότι είναι πολύ κοντά στο πραγματικό σύνορο κατάτμησης. Τέλος, η προσέγγιση του συνόρου κατάτμησης γίνεται ακριβέστερη έτσι ώστε να περνάει από τις καμπυλότητες του αντικειμένου με χρήση ενός αλγορίθμου ελάχιστης περικοπής.

Η δεύτερη συνεισφορά του Διδακτορικού είναι μία μεθοδολογία ανάκτησης 3D αντικειμένων που βασίζεται στην αναπαράσταση βασισμένη σε γράφημα ενός αρθρωτού 3D αντικειμένου το οποίο δημιουργείται από τον προτεινόμενο αλγόριθμο κατάτμησης. Συγκεκριμένα, τα τμήματα που εξάγονται από τον αλγόριθμο κατάτμησης τίθενται σαν κορυφές του γραφήματος και οι ακμές του συνδέουν όλα τα προεξέχοντα τμήματα με το κυρίως σώμα του αρθρωτού αντικειμένου. Οι κορυφές και οι ακμές του γραφήματος σχετίζονται με μοναδιαία (unary) και δυαδικά (binary) χαρακτηριστικά που αναπαριστούν τα γεωμετρικά χαρακτηριστικά των τμημάτων όπως επίσης και τη σχέση που έχουν μεταξύ τους. Η αναπαράσταση βασισμένη σε γράφημα μπορεί να θεωρηθεί σαν ένα Σχεσιακό Γράφημα Χαρακτηριστικών (Attributed Relational Graph). Το σχεσιακό γράφημα χαρακτηριστικών του αντικειμένου χρησιμοποιείται στη συνέχεια για αποτελεσματική ανάκτηση 3D αρθρωτών αντικειμένων. Συγκεκριμένα, το Σχεσιακό Γράφημα Χαρακτηριστικών του εξεταζόμενου αντικειμένου (query object) ταιριάζεται με τα Σχεσιακά Γράφημα Χαρακτηριστικών των αντικειμένων σε μια βάση που περιέχει 3D αρθρωτά αντικείμενα. Το αποτέλεσμα της διαδικασίας ανάκτησης είναι μία

ακολουθία από αντικείμενα από τη βάση παρόμοια με το εξεταζόμενο αντικείμενο. Το χρησιμοποιούμενο κριτήριο ομοιότητας βασίζεται σε μία μετρική ομοιότητας που ονομάζεται Earth Mover's Distance.

Η βελτιωμένη απόδοση του προτεινόμενου 3D σχήματος κατάτμησης καθώς επίσης και η προτεινόμενη μεθοδολογία ανάκτησης 3D αρθρωτών αντικειμένων έχει επιδειχθεί μέσω εκτενούς αξιολόγησης σε καθιερωμένη βάση 3D αντικειμένων σε αντιπαράθεση με άλλες σημαντικές σύγχρονες μεθοδολογίες.

Abstract

3D object representations have become an integral part of modern computer graphics applications, such as computer-aided design, game development and film production. At the same time, 3D data have become very common in domains such as computer vision, computational geometry, molecular biology and medicine. The rapid evolution in graphics hardware and software development, in particular the availability of low cost 3D scanners, has greatly facilitated 3D model acquisition, creation and manipulation, giving the opportunity to experience applications using 3D models to a large user community. As the number of 3D models is continuously growing the problem of creating new 3D models has shifted to the problem of searching for existing 3D models. Thereupon, the development of efficient search mechanisms is required for the effective retrieval of 3D objects from large repositories, both of a single class and across classes.

In this dissertation, a graph-based representation of a 3D object is introduced using a novel segmentation algorithm and the attributed relational graph concept. The proposed graph-based representation sets the base for an effective 3D object retrieval methodology of articulated objects.

The contribution of this dissertation is two-fold.

The first contribution is a 3D mesh segmentation algorithm based on the premise that a 3D object consists of its main body part and its constituent protrusible parts. The proposed segmentation algorithm aims to segment the object into these parts. To achieve this goal, first the salient points of the object are detected. These points are prominent features of the 3D object that guide the rest of the segmentation. Afterwards, the salient points are gathered into groups representing the main protrusible parts of the object. In the sequel, the main body (core) of the object is approximated using the minimum cost paths between the salient points of the object. The key idea of the proposed core approximation is to expand a set of vertices in ascending order of protrusion function value until the expanded set touches a certain percentage of all elements of the minimum cost paths. The core approximation either covers portions of the protrusible part areas or is very close to the neighboring areas where the real boundary between the core component and the protrusible part is situated. Afterwards, the partitioning boundary is detected, that is the boundary between a protrusible part and the main body of the mesh. It is considered that in the area enclosed by the desired

boundary between the protrusible part and the main body, an abrupt change in the volume of the 3D object should occur, thus, the goal is to detect this change. To accomplish this, closed boundaries are constructed which are defined by a distance function associated to a representative of the group which represents the protrusible part. The abrupt change of volume is detected by examining the closed boundaries perimeter and setting the closed boundary where the largest change of perimeter occurs as the partitioning boundary approximation. The proposed partitioning boundary approximation is very effective since it is shown in the experimental results that it is very close to the real partitioning boundary. Lastly, the partitioning boundary is refined so that it passes through the concavities of the object using a minimum cut algorithm.

The second contribution of this dissertation is a 3D object retrieval methodology that relies upon a graph-based representation of an articulated 3D object produced by the proposed segmentation scheme. In particular, the parts extracted by the segmentation algorithm are set as the nodes of the graph structure while its edges connect all the protrusible parts with the part representing the main body of the articulated object. The nodes and the edges of the graph structure relate to unary and binary attributes which represent the geometrical characteristics of the parts as well as the relationships with each other. This graph-based representation can be viewed as an Attributed Relational Graph (ARG). The ARG of the object is later used for effective 3D articulated object retrieval. Specifically, the query's object ARG is matched with the ARGs of the objects in a database containing 3D articulated objects. The outcome of the retrieval process is a sequence of objects from the database similar to the query object based upon a similarity criterion that relies upon the Earth Mover's Distance similarity measure.

The improved performance of the proposed 3D mesh segmentation scheme as well as the proposed object retrieval methodology for 3D articulated objects has been demonstrated by an extensive evaluation in standard 3D object databases against the major state-of-the-art methodologies.

Table of contents

1. Introduction	10
1.1 Scope of Dissertation	10
1.2 Innovations of Dissertation	12
1.3 Dissertation layout.....	13
2. State of the art in 3D mesh segmentation.....	14
2.1 Introduction	14
2.2 Basic Definitions	15
2.3 State of the art in surface segmentation	16
2.3.1 Region Growing	16
2.3.2 Watersheds	18
2.3.3 Reeb Graph Method	20
2.3.4 Model based Methods	23
2.3.5 Skeleton based Methods.....	24
2.3.6 Clustering Methods	26
2.3.7 Spectral Analysis.....	30
2.3.8 Explicit Boundary Extraction Methods.....	31
2.3.9 Volumetric Methods.....	33
2.3.10 Critical Points.....	35
3. Proposed 3D mesh segmentation algorithm	39
3.1 Proposed Surface Segmentation Methodology	39
3.1.1 Salient points extraction	42
3.1.2 Salient points grouping	43
3.1.3 Core approximation.....	45
3.1.4 Partitioning Boundary Detection.....	48
3.1.5 Partitioning Boundary refinement.....	54
3.2 Experimental Results.....	56

3.3 Conclusions	69
4. State of the art in retrieval of 3D articulated objects	71
4.1 Introduction	71
4.1.1 The 3D object retrieval problem	71
4.1.2 Graph-based Retrieval of articulated objects	72
4.2 State of the art in 3D object retrieval	73
4.2.1 Methods with global shape representations	73
4.2.3 Methods using a graph-based representation	79
5. Proposed 3D retrieval algorithm	85
5.1 Earth Movers Distance Similarity Measure	85
5.2 Proposed retrieval methodology for articulated objects.....	87
5.2.1 Construction of the ARG of the 3D object.....	88
5.2.2 A methodology for Matching two ARGs.....	88
5.3 Experimental Results.....	93
5.4 Conclusions	102
6. Conclusions	103
Acknowledgement	105
Publications derived from this PhD Phesis.....	105
References	106

Chapter 1

Introduction

1.1 Scope of Dissertation

In recent years there has been an increasing demand for 3D object recognition and retrieval.

Recognition of a 3D object is achieved by understanding its structure. According to the Recognition by Components (RBC) theory of Biederman [Bie87] human perception understands the structure of the 3D object by breaking it into parts and assigning to them basic volumetric primitives [Bie87]. Specifically when an image of an object is painted on the retina, RBC assumes that a representation of the image is segmented-or parsed-into regions of deep concavity, particularly at cusps where there are discontinuities in curvature. Each segmented region is then approximated by one of a possible set of simple components called geons (for “geometricalions”) that can be modeled by generalized cones.

This process has also been mimicked in Computer Vision in order to understand and model the structure of the objects so that they can be recognized by a system. The basic process that is used in object recognition is segmentation, i.e. the process that breaks the object into meaningful volumetric parts.

In the literature there exist a plethora of surface segmentation algorithms that partition the mesh in meaningful components. All of these algorithms use different criteria and methodologies to achieve this goal.

In this dissertation a 3D mesh segmentation algorithm has been developed and will be presented analytically in the subsequent Chapter 2. This algorithm is based on the premise that a 3D object consists of its main body and its constituent protrusible parts.

The scope of the segmentation algorithm is to separate (segment) the object into the prementioned parts i.e the main body and its protrusible parts. This segmentation process is initiated with the detection of the so called *salient* points. These points are

the prominent features of the 3D object and guide the rest of the segmentation procedure.

The next step is to bring together into distinctive groups the salient points which belong to the particular protrusible part connected to the main body.

In the sequel the main body which can be also named as the “core” of the 3D object will be approximated using the minimum cost path that connect the salient points of the object. This core approximation either covers portions of the areas of the protrusible parts or approaches in the best possible way the neighboring areas in which the real boundaries separate the core and the associated protrusible parts.

In succession the *partitioning boundary*, i.e. the boundary between the main body and the particular protrusible part of the mesh, is detected. The basis of this detection is the obvious expected abrupt change of the volume which happens in the area which is enclosed by the partitioning boundary under investigation. The goal of this process is to detect this abrupt change of volume. Finally the detected partitioning boundary is further refined in order to pass through the concavities of the object using a *minimum cut* algorithm.

3D object retrieval is the problem where given a 3D query object and a database consisting of 3D objects similar in shape objects as the query has to be extracted from the database. This problem has received over the last years significant attention and various methodologies have been proposed to deal with it. Most of these approaches are based on global representations of the 3D object, i.e. 2D/3D features are extracted from the object and are used to globally describe it in a feature space. In this case the similarity between the query object and the objects of the database is accomplished by comparing their respected features. The problem with these approaches is that with a global descriptor similar articulated objects in shape under different poses can be regarded as non similar in the retrieval process.

In this thesis a different methodology for retrieval is going to be followed that is based on a graph-based representation and can successfully retrieve articulated objects. The basis of the whole retrieval process is the construction of a graph based representation of an articulated 3D object. This graph is obtained using the segmentation algorithm mentioned already. Specifically the parts that the segmentation algorithm extracts will be set as the nodes of the graph structure while its edges will connect all the protrusible parts with the one representing the main body of the articulated object. Finally unary and binary attributes are assigned representing the parts geometrical characteristics as well as the relationships existing between the parts.

This graph based representation is named *Attributed Relational Graph (ARG)* and is going to be used for efficient 3D articulated object retrieval. Specifically the ARG of the query articulated 3D object will be compared with the ARGs of the 3D articulated objects being stored in the database. The result of this comparison will be a sequel of objects taken from the database which are similar to the query object. The retrieval will be obtained using a graph – matching algorithm concept which is based on the *Earth Movers Distance* Similarity Measure.

1.2 Innovations of Dissertation

This dissertation is innovative both in the proposed segmentation and retrieval algorithms. For a survey in mesh segmentation algorithms see [APP+07].

Concerning the proposed segmentation algorithm, although it is aligned with a general framework introduced by Lin *et. al.* [LLL07], new approaches have been introduced for the implementation of distinct stages of the framework leading to improved efficiency and robustness as has been verified by an extensive experimental evaluation.

The two most significant contributions of the segmentation algorithm is the approximation of the core of the object and the detection of the partitioning boundaries. Both of these approaches differ significantly from the approach followed by Lin *et. al.* [LLL07].

The core approximation is based on a new algorithm that expands a set of vertices starting from the center of the object and the expansion continues until the expanded set reaches the protrusible parts of the object and in some cases cover a portion of them. In Lin *et. al.* a simple thresholding is being used that will be shown to be inadequate in section 3.1.4. The core approximation is a very crucial part in the whole segmentation framework since the partitioning boundary detection is heavily based on its correct approximation. In the experimental results of section 3.2 it will be shown that the approximation of the core is very robust.

A novel way to trace the partitioning boundaries is also introduced. The novelty is based on the use of closed boundaries to detect the sudden change of volume which happens at the area which encloses the partitioning boundary. In the proposed algorithm this area will be defined with the aid of the core approximation and this area

will be swept by these closed boundaries. The major advantage of this approach over Lin's *et. al.* [LLL07] approach is that the defined closed boundaries are not dependant on the resolution of the mesh, are less susceptible to noise and sweep a small area of the mesh where the true partitioning boundary lies. The segmentation algorithm was published in [APPS].

Concerning the retrieval part of the dissertation the attributed relational graph concept has been used in order to describe an object. The innovative part of the dissertation is the use of the EMD similarity measure based on newly defined ground distances in order to match the ARGs of two articulated objects.

In the experimental results (section 5.3) the superiority of our retrieval methodology will be shown against the MPEG7 retrieval methodology which is based also on a graph representation of the object and the retrieval methodology of Papadakis *et. al.* [PPT+08] which is based on a global representation of the object.

The results of the proposed retrieval algorithm were presented in [APP+09, APP+].

1.3 Dissertation layout

The layout of this dissertation is as follows :

In chapter 2 the state of the art of surface segmentation will be presented where a wide spectrum of the segmentation algorithms will be categorized according to the methodology it is used in order to achieve segmentation. In chapter 3 the proposed segmentation algorithm will be presented and all of its distinct stages will be fully described. Experimental results will also be presented based on a consistent framework presented in the literature. In chapter 4 the state of the art in retrieval of articulated objects will be presented. In chapter 5 the proposed retrieval methodology will be introduced and also experimental evaluation will be made which shows the highly efficient performance of the proposed retrieval methodology in retrieving articulated object. Lastly in Section 6 conclusions will be drawn.

Chapter 2

State of the art in 3D mesh segmentation

2.1 Introduction

In recent years there has been an increasing demand in Computer Graphics, Vision, and Multimedia, for efficient manipulation of 3D objects due to their plethora and easy to be acquired methods (e.g. from laser range scanners). This manipulation may involve a wide range of computer graphics techniques like retrieval, rendering, texture mapping, transmission, geometric morphing, collision detection etc.

Usually 3D models are represented by low level structures such as a triangular mesh of irregular connectivity which makes them hard to be manipulated.

Surface segmentation, a process which segments the mesh into subparts, is very useful for alleviating this lack of high level representation and provides a base for surface understanding. It is capable of breaking a specific task into subtasks which can be more easily manipulated making it a versatile process.

In the literature a plethora of algorithms for surface segmentation has been presented. Some of these belong to methodologies and/or techniques as for example “The Region Growing Technique”, “The Watersheds Approach”, “The Reeb Graph Methods”, “The Skeleton based methods” etc etc. All these are going to be presented shortly in the next section. The associated algorithms under a specific objective use various criteria for grouping mesh elements together. Additionally, these criteria may use various local or global shape features extracted from the mesh and are often valid under some constraints. This means that segmentation is a heavily specific application. However there seems to be common constituents in the objectives of many segmentation algorithms. In the works of Shamir and Agathos, [Sha04, APP+07] based on the segmentation objective, two types of segmentation algorithms are distinguished. Specifically:

- Patch type segmentation, which objective is to segment the mesh into surface patches. This type of segmentation is useful for parametrization, texture mapping, remeshing, simplification, etc.
- Part type segmentation, which objective is to segment the mesh into meaningful to human perception components. This type of segmentation is based on the theory of cognitive science which describe how humans recognize objects [HR84, Bie87, HR97, SH01].

The chapter organization is as follows:

- Section 2.2: Introduction of basic definitions.
- Section 2.3: Presentation of the state of the art in surface segmentation.

2.2 Basic Definitions

A 3D mesh, M , is a discrete approximation of the surface of a 3D object consisting of a set of Vertices, Edges and Faces. The edges and faces of the mesh are always incident to the vertices of the mesh. In most of the cases the faces are triangles.

Let S be the set of either the vertices, edges or faces of the mesh.

Segmentation of M is called the partitioning of S into k -disjoint sets, i.e:

$$\bigcup_{i=1}^k S_i = S, S_i \subset S, S_i \cap S_j = \emptyset, i, j = 1 \dots k, i \neq j \quad (2.2.1)$$

The *dual graph* of a mesh is the graph for which nodes are the triangular faces and each node is connected with another if their representative triangles share a common edge. The triangles, which represent its nodes, will be referred as the *dual triangles*, see figure 2.2.1.

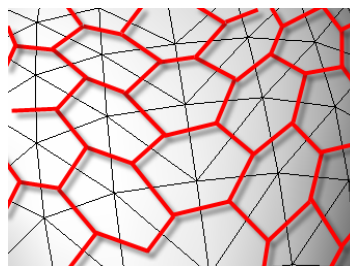


Figure 2.2.1. The dual graph a mesh denoted by red lines

2.3 State of the art in surface segmentation

In the literature there are several algorithms created for surface segmentation. These algorithms can be grouped into categories according to the method used for segmentation.

For the sake of clarity, from now on, the parts that the object is segmented into will be called *components or segments*. The boundaries where components (segments) meet will be called *partitioning boundaries*.

2.3.1 Region Growing

With this technique segmentation regions are generated with the expansion of seed elements (points, triangles, regions) that belong in a part. This expansion takes place only if certain criteria hold. These criteria depend on the objectives and constraints set by the segmentation algorithms.

For example, point Σ in Figure 2.3.1 expands to its neighbors and adds them to the part it belongs to if the criteria set hold. Subsequently, this process is repeated by the expansion of these neighbors and the process ends until no other points can be added.

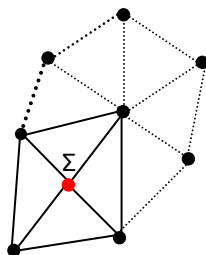


Figure 2.3.1. Point Σ “grows” adding its neighbors to the part it belongs.

Zhang *et. al.* [ZPKA02] label the points according to their Gaussian curvature. Following the *transversality regularity*¹ [HR84], by thresholding, they label points of high negative Gaussian curvature as boundary points.

By selecting randomly seed points among the non boundary points they do region growing. The growing process does not add points which belong to the boundary

¹ The transversality regularity states that when two arbitrarily shaped surfaces are made to interpenetrate they always meet in a contour of concave discontinuity of their tangent planes

or those that are already added to the region. Every non boundary point that is not examined by the neighboring scheme creates a new part. The region growing process ends when all non boundary points have been assigned to a part. Points which belong on the boundary are added to their closest region. Finally regions consisting of small number of points are merged to neighboring larger regions and the segmentation process ends.

One of the disadvantages of their method is that the output is heavily dependent on the choice of the user defined threshold which determines whether a point will belong to the boundary. Also not all boundaries can be found by using only the Gaussian Curvature [HR84]. In addition if the points labeled as boundaries do not form a closed loop then the algorithm will fail in finding the proper parts of the 3D object.

Zuckerberger *et. al.* [ZTS02] extract the parts of the object with a flooding on the dual graph of the mesh. Specifically, their segmentation technique starts from a node traversing the graph using a Depth First or Breadth First Search. This traversal incrementally collects faces as long as they form a convex patch. If convexity is violated then a new patch is created from an unvisited node and the traversal is resumed. The whole process ends when there are no vertices left unvisited.

The main problem with this approach is oversegmentation which the authors propose to be dealt by merging smaller regions to larger ones according to their area.

Sapidis and Besl [SB95] following a trial-and-error process perform region growing in range image data which segment them in N quasi-disjoint regions which can be approximated by N polynomial functions in a way that:

- (i) The distance between a point of the region and its associated polynomial is smaller than a threshold
- (ii) The total number of data elements that each region contain is maximized
- (iii) A C^1 or C^2 continuity can be enforced between adjacent polynomials

Region growing starts with a seed region to which a polynomial is fitted. This region grows according to a distance and orientation criterion and the approximating polynomial of the expanded region is computed.

The process continues until there can be no further expansion. If the number of points that do not belong to a region is significant then the whole process is applied again beginning with a new seed region.

Sorkine *et. al.* [SCOGL02] partition the mesh in order to map (flatten) it into a region of the 2D plane with bounded distortion. The mesh partitioning is accomplished by region growing. Specifically, they start from a seed triangle and iteratively attach adjacent vertices as long as the distortion caused by the flattening of its adjacent triangles does not exceed a predefined threshold and there is no self-intersection in the 2D plane. If no more vertices can be added a new seed triangle is selected and the process starts again. The main disadvantage of this method is that the segmentation may lead to the creation of very small patches in order to avoid self-intersections and over distortion.

2.3.2 Watersheds

Initially, the watershed transformation was used for 2D/3D image segmentation (see for example [VS91, Pra98]). Mangan *et. al.* [MW99] was the first to extend it to 3D meshes. This method has been named this way because it reminds the way that water fills a geographic surface. As the water floods its basins there will be points where flood regions meet. These points are dividing and watersheds are placed at that points so as to divide the regions, see Figure 2.3.2. Thus watersheds segment the geographic surface into regions. The flooded basins are called catchment basins and they represent the parts of the object.

For segmentation purposes the watershed algorithm is not applied on the original 3D Mesh but it is applied on a transformed version based on a function $f : \mathfrak{R}^3 \rightarrow \mathfrak{R}$ that guide the watersheds to identify the crest lines of the mesh. There is a one to one correspondence between the minima of f and the catchment basins.

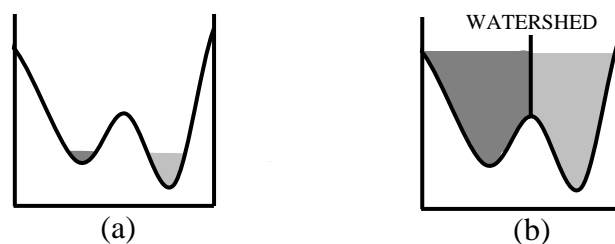


Figure 2.3.2. (a) Initial flooding of the geographic surface, (b) Watershed line emerged at a certain flooding level

The majority of the algorithms use curvature as the watershed function: In [PRF02] various types of curvature has been used like the Gaussian, the Mean, the Absolute etc. In [MW99] the square root of the deviation from flatness has been used ($D = \sqrt{k_1^2 + k_2^2}$, k_1, k_2 principal curvatures). In [PKA03, PAKZ03] the normal curvature has been used.

From the algorithmic point of view the extraction of catchment basins can be achieved in two ways:

- 1) *Steepest Path Following*, where a point on the surface follows its steepest path and slides to a local minima of f . This point and the followed path belong to the part that the minima of the catchment basin define.
- 2) *Flooding*, where beginning from the local minima, points travel towards higher values of f until they reach local maxima or an already labeled point

One of the main disadvantages of the watershed segmentation is that most of the times it leads to oversegmentation. One solution to deal with this is by finding the depths of the basins. Areas of shallow basins are merged with neighboring areas of deeper basins. Page *et. al.* [PKA03, PAKZ03] implemented a scheme that diminishes the problem of oversegmentation. In this scheme, the basins are filled until a certain point by thresholding and applying 3D morphological operators [RKS00] on the minimum curvature. Afterwards the basins continue to be filled with a flooding watershed algorithm using as function the normal curvature.

Zuckerberger [ZTS02] uses the edges of the triangles to flow to the minimum. By this method a better segmentation is achieved in some cases where the vertices of the input mesh might not be dense enough to define the catchment basins. In their case the dihedral angle of adjacent to the edges triangles is used as a watershed function. The main disadvantage of this method is that due to the small local support of the dihedral angle oversegmentation can be very intense.

2.3.3 Reeb Graph Method

Let $f: \mathbf{M} \subseteq \mathfrak{R}^3 \rightarrow \mathfrak{R}$, where \mathbf{M} is a three dimensional surface. The Reeb graph of the surface \mathbf{M} is the quotient space [AA89] of f in $\mathbf{M} \times \mathfrak{R}$ generated using the following equivalence relation:

$$(u_1, f(u_1)) \sim (u_2, f(u_2)) \Leftrightarrow \begin{cases} f(u_1) = f(u_2) \\ \& \\ u_1, u_2 \text{ lie in the same connected component of} \\ f^{-1}f(u_1) \end{cases} \quad (2.3.1)$$

If for example f is a height function which returns the z coordinate of a point on a surface \mathbf{M} , then the equivalence classes of the Reeb Graph of a surface \mathbf{M} are contours perpendicular to the height direction, see Figure 2.3.3.

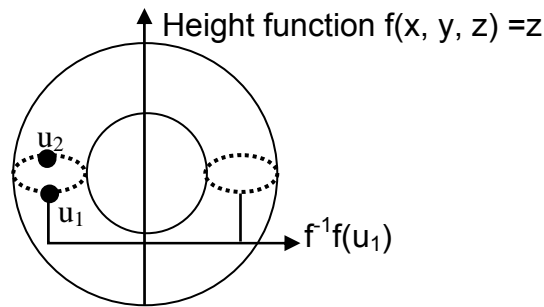


Figure 2.3.3. A torus object associated with the height function $f(x, y, z) = z$. The two points u_1, u_2 on the mesh belong to the same connected component thus $(u_1, f(u_1)), (u_2, f(u_2))$ are equivalent.

One of the main disadvantages of the height function is that it is not suitable for 3D objects since it is not invariant to transformations such as rotation.

Hilaga *et. al.* [HSKK01] first presented a suitable function for 3D meshes which is pose invariant. This function will be called in this work *protrusion function*, pf , and represents the magnitude of a protrusion on the Surface.

This function on a point u on \mathbf{M} is defined as:

$$pf(u) = \int_{p \in \mathbf{M}} g(u, p) dM \quad (2.3.2)$$

where $g(u, p)$ the geodesic distance of the surface points u, p .

As can be observed from equation 2.3.2 the protrusion function on a point of a surface is the sum of the distances from all of the surface points

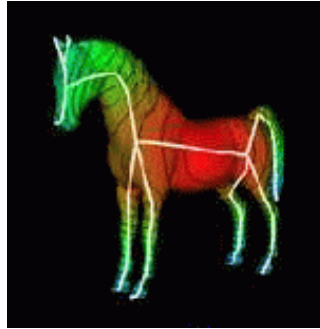


Figure 2.3.4. Chromatic representation of the protrusion function pf , red color denotes small values of pf , blue denote large values of pf and green color denotes values in between, the Reeb Graph of the object is denoted by white while contours with the same value of pf are denote in black (taken from [HILA01])

Large values of pf denote that u belongs on a protrusion while small values denote that u is near the center of the surface, see Figure 2.3.4.

In order to reduce the computational cost of the calculation of the protrusion function the mesh is tessellated uniformly into N compact patches [HSKK01, VC04], see Figure 2.3.5. These patches have also a representative vertex based on a criterion (eg. Voronoi centroid, geodesic centroid, etc). Using these patches and vertices the protrusion function can be approximated as follows:

$$pf(v) = \sum_i g(v, b_i) area(b_i) \quad (2.3.3)$$

where $area(b_i)$ is the area of the patch with representative vertex b_i .

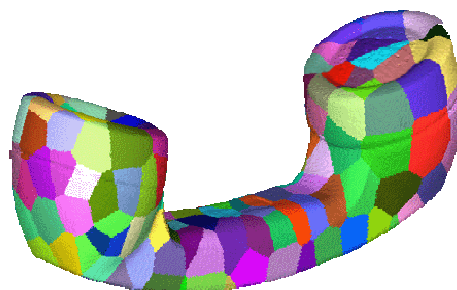


Figure 2.3.5. The tessellation of a mesh into uniformly distributed compact patches using Valette *et. al.* algorithm [VC04].

Antini et. al. [ASBP05] is quantizing the protrusion function in order to achieve segmentation. For example the quantization (in 7 bins) of the protrusion function of a human model is illustrated in Figure 2.3.6(a).

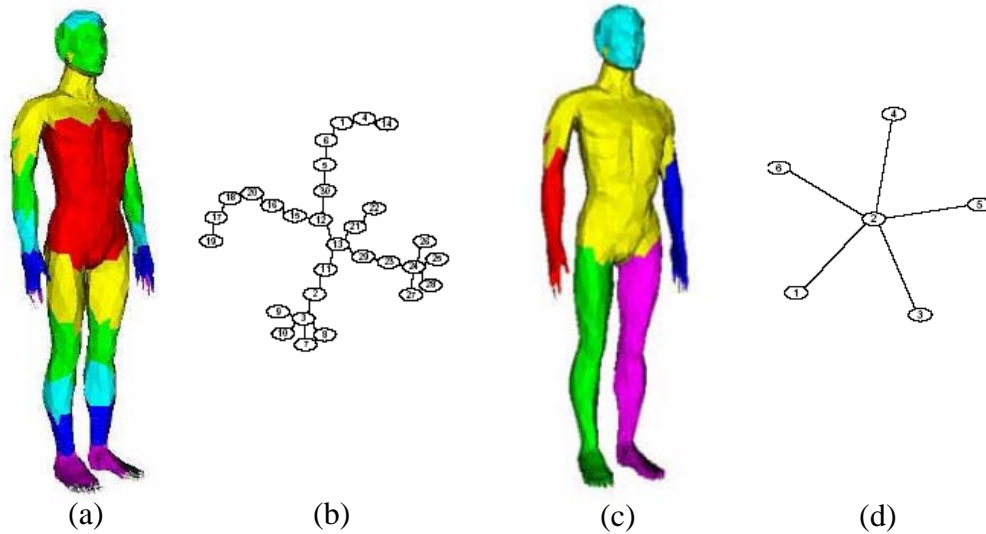


Figure 2.3.6. (a) Quantization in 7 levels, (b) The Discrete Reeb Graph which corresponds to the quantization (c) The parts after the simplification of the Reeb Graph (d) The Simplified Graph (taken from [ASBP05])

Based on this quantization a discrete Reeb can be constructed. Each of its nodes represents a connected component which elements have the same quantized value and two nodes are connected by an edge if the corresponding components are neighbors, see Figure 2.3.6(b).

Their segmentation objective is to acquire perceptually meaningful components. As can be seen from Figure 2.3.6(a) the quantization of the Reeb Graph results to oversegmentation. Antini et. al. managed to merge regions by simplifying the Reeb Graph, i.e. Graph nodes are merged together based on their topological and Mean curvature information, see Figure 2.3.6(c), (d).

While Antini et. al. manage to acquire perceptually meaningful components the boundaries between regions do not reside at areas of concavities where the human visual system expect them to be located [HR84].

Valette et. al. [VKS05] constructed a segmentation algorithm that also takes advantage of the protrusion function. Specifically they introduced the set $C_x = \{p \in M \mid pf(p) \geq x\}$ and examine the way it evolves as x decreases. In this evolution process C_x starts from just containing points of protrusion (e.g. arms, legs, etc.) and expands towards the center of the object. They have called this process

protrusion conquest. In Figure 2.3.7 the result of Valette et. al. protrusion conquest is illustrated, as can be observed segmentation is not done at the concavities of the mesh.

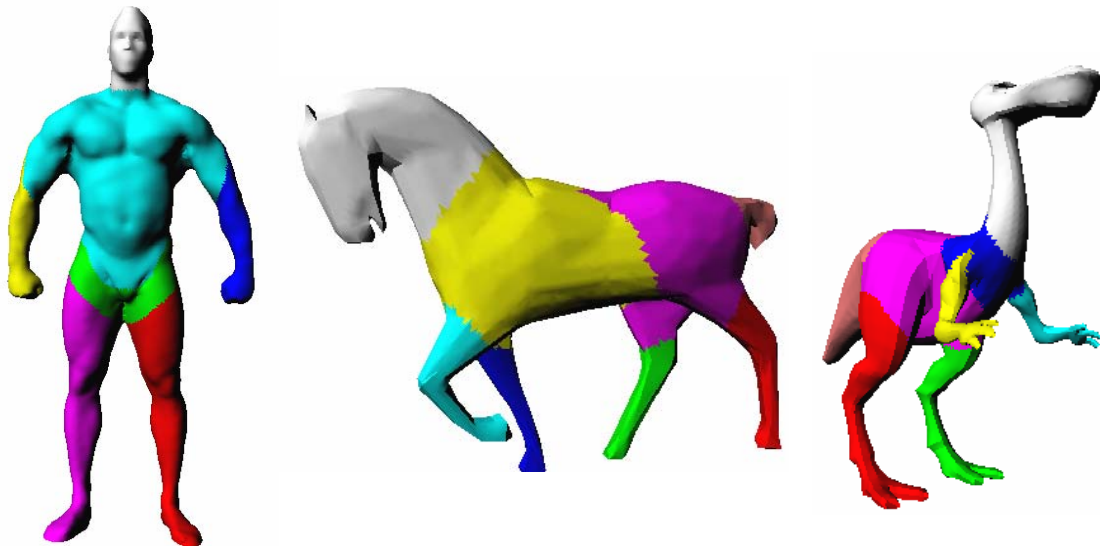


Figure 2.3.7. Results of Valette et. al. Protrusion conquest, different colors denote different parts

2.3.4 Model based Methods

Wu *et. al.* [WL97] use the distribution of electrical charge across the surface in order to achieve segmentation. Specifically they used the property that the density charge is very low at areas of deep concavities of the surface and very high at areas of high convexities of the surface. According to the minima rule [HR84], partitioning boundaries reside on areas of the mesh where minima in the density charge occur.

Assuming that a triangle has a constant density charge Wu *et. al.* created a set of linear equations whose solution provide each triangle's density charge.

The boundary extraction process of a closed surface starts with a node on the dual graph of the mesh where there is a deep local minimum in the density charge. The algorithm proceeds to the neighboring node which has the smallest value in density charge marking it as visited. This process is then repeated until a closed boundary loop is generated.

A similar process is followed when the object is constructed from single view data with the only difference that if a boundary node is reached then a different path from the initial node has to be followed in order to acquire the whole boundary.

After the boundaries of the object are found they are extracted from the mesh dividing it into subparts which can be acquired by applying a floodfill algorithm on the disconnected subgraphs of the dual graph.

One of the advantages of this method is that it uses a global characterization of the surface to find concavities while most other methods use local characterization like curvature. This makes it less sensitive to noise. Disadvantages of this method is that it can only trace boundaries surrounded by concavities.

2.3.5 Skeleton based Methods

Li *et. al.* [LTH01] did segmentation using the skeleton of the object. The skeleton of the object is constructed by performing simplification of the surface using the edge contraction method. The priority for contraction is the length of the edge to be contracted. Their simplification process leaves edges that do not have incident triangles unaltered, they call these edges *skeletal*. At the end of the contraction process only skeletal edges are left, whose union is the skeleton of the object, see Figure 2.3.8(b).

The parts of the object are extracted by using a plane which sweeps the mesh along the skeleton edges. The intersection of the plane and the mesh is one or more polygon contours. The parts are extracted by examining the way that these contours alter as the sweep plane moves. For this purpose a parametric geometric and topological function is defined on these contours. Critical points on these functions signify partitioning boundaries.

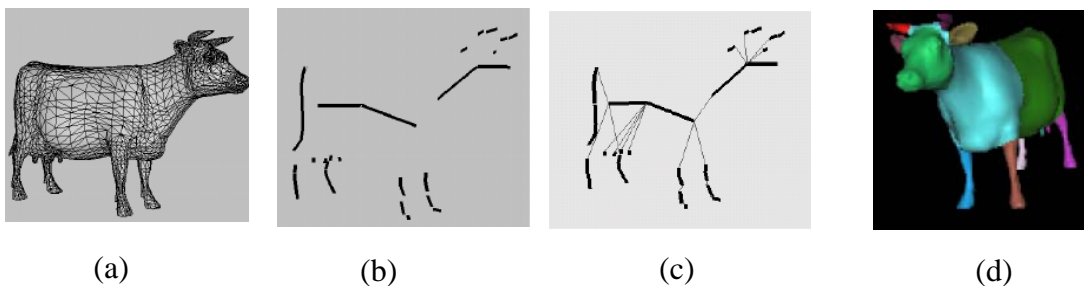


Figure 2.3.8. (a) The polygonal object, (b) The skeleton of the object , (c) Skeleton with virtual edges, (d) Segmented model (taken from [LI01])

In order for this sweeping to be successful the skeletal edges are ordered by joining them with virtual edges, constructing by this way a *skeletal tree*, see Figure 2.3.8(c). This tree defines a sweeping space ordering for path following.

A disadvantage of this method is that due to the application of smoothing filters on the parametric functions there can be cases where some features fail to be extracted.

Reniers and Telea [RT08] also proposed a skeleton-based method in order to achieve segmentation. First they defined the skeleton S of a 3D object Ω with boundary (surface) $\partial\Omega$. The skeleton is defined as those points in Ω having at least two boundary points at minimum distance these points are called feature points. The skeleton consists of 2D manifolds and 1D curves. From the skeleton they create the curve-skeleton C by considering the shortest geodesic curves between the feature points of a point p belonging to S , let this set of geodesic curves for the point p called $\Gamma(p)$. By considering the loops that are created in $\Gamma(p)$ they determine whether a point belongs to the curve-skeleton C as follows:

$$p \in C \Leftrightarrow \text{genus}(\Gamma(p)) \geq 1 \quad (2.3.4)$$

The point p is a normal point of C if $\text{genus}(\Gamma(p))=1$ and a junction point if $\text{genus}(\Gamma(p)) \geq 2$. The set Γ at the junction points of C splits the object into components. In their implementation they find the curve-skeleton by voxelizing the 3D object and approximating the curve-skeleton with voxels proposing a methodology to find the feature voxels of the voxelized skeleton-curve and the shortest geodesics between them.

Using the junction points they define a skeleton-to-boundary mapping as follows: The set $\Gamma(p)$ of a point on C divides the objects surface into a set of components $\text{Comp}(p)$ which are ordered in ascending order in terms of their area.

By considering the sets of components at the junctions of the curve-skeleton they partition the mesh proposing a flat and a hierarchical segmentation algorithm. In their flat segmentation algorithm they simply label all the components that are generated by using the set $\Gamma(p)$.

In their hierarchical segmentation algorithm they define a set F which consists of the union of all the components that the junction points generate and they create a hierarchical segmentation S from F based on the area of the components size, i.e. they

extract from F the components in ascending order of area size and accept them as segments of S if their area are greater than a scale τ .

In [MDA+08] the Medial surface of the object [SBTZ02] is used for segmentation, which is an extension of the Medial Axis Transform of an image [SBTZ98]. First the object is voxelized using a fixed resolution. Afterwards the voxelization is thinned using the Average Outward Flux (AOF) of the Euclidian distance Transform's Gradient field on each voxel x [SBTZ02] defined as:

$$AOF(x) = \frac{1}{n} \sum_{i=1}^n \langle \nabla D(x_i), \mathbf{n}_i \rangle \quad (2.3.5)$$

where n is the number of the neighboring voxels of x and \mathbf{n}_i is the outward normal from the voxel x to voxel x_i . The value of AOF is approaching a negative value around the medial surface points and is close to zero elsewhere. Thinning is done by recursively removing all the voxels that are close to zero and do not affect the topology of the surface [SBTZ02]. Afterwards they segment the medial surface using [MBA93]. Then they readjust the resulting segments into larger more meaningful parts. Finally they segment the boundary voxels of the voxelized mesh according to their proximity with the voxels of the segmented medial axis.

2.3.6 Clustering Methods

Shlafman et. al. [STK02] use the k-means clustering algorithm in order to achieve segmentation. Their segmentation scheme clusters triangles and proceeds as follows:

- (i) Compute the distances between each triangle pair
- (ii) Compute the triangle cluster representatives
- (iii) Assign all other triangles to their nearest representative
- (iv) Update the representatives by minimizing the sum of distances between the triangles belonging to the cluster and the representative.

Steps (iii) and (iv) are iterated until there is no further update or a predefined number of iterations is reached. The final computed clusters are the parts of the object.

The distance function between each pair of adjacent triangles is a weighted sum of the geodesic distance of the barycenters and the angle between the triangles, i.e for two triangles F_i, F_j :

$$dist(F_i, F_j) = \delta \text{geod_dist}(F_i, F_j) + (1 - \delta)(1 - \cos^2(\alpha)) \quad (2.3.6)$$

where α is the dihedral angle of F_i, F_j . The distance between two faces that are not adjacent can be computed by finding the shortest path between them.

Katz *et. al.* [KT03] is using a fuzzy k-means algorithm to segment the object. As in Shlafman *et. al.* [STK02] their algorithm is based on clustering triangles. At each pair of triangles F_i, F_j a distance measure is defined as:

$$Dist(F_i, F_j) = \delta \frac{\text{geod_dist}(F_i, F_j)}{\text{avg}(\text{geod_dist})} + (1 - \delta) \frac{\text{ang_dist}(a_{ij})}{\text{avg}(\text{ang_dist})} \quad (2.3.7)$$

where:

- $\text{geod_dist}(F_i, F_j)$, the geodesic distance between the barycenters of F_i, F_j
- $\text{ang_dist}(a_{ij}) = n(1 - \cos(a_{ij}))$, where $n = 1$ for concave angles and very small for convex angles, a_{ij} the dihedral angle of F_i, F_j .
- $\text{avg}()$, the average of all the values

They first consider binary clustering, i.e. the classification of the triangles into two clusters. Each triangle of the mesh will belong to a cluster according to a fuzzy membership function, which denotes the likelihood of a face to belong in one of the two clusters. This function is based on the distances of the triangle from the representatives triangles of each cluster which are iteratively updated by a fuzzy k-means algorithm.

After the convergence of the k-means algorithm by thresholding appropriately the membership function they simulate hard-clustering, acquiring three groups, i.e. two groups with triangles belonging to only one cluster (part) and a third fuzzy group which contain triangles that are not certain to belong in a specific part. See Figure 2.3.9 for a visualization of this grouping.

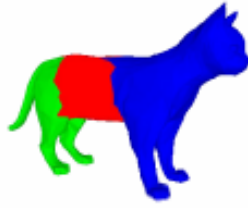


Figure 2.3.9. Katz et. al. fuzzy clustering. Blue and Green denote parts and Red denotes the fuzzy unclassified region.

In order to classify the members of the fuzzy group to one of the parts of the object a minimum cut algorithm [CLRS01] is applied on the dual graph of the mesh belonging to the fuzzy group. The weight applied on the edges of the graph is small on concavities and large on convexities. Thus the minimum cut algorithm bisects the fuzzy group in deep concavities. Triangles from the fuzzy group are assigned to a component according to which side of cut they belong.

They have created also a hierarchical structure based on this decomposition. Each node of the hierarchical tree denotes an unsegmented area that is segmented in two new nodes with their decomposition algorithm. The initial node is the whole object.

They also generalize this binary hierarchical clustering into a $k > 2$ hierarchical clustering. The k clusters are computed automatically by an iterative algorithm which maximizes the minimum distance from already assigned cluster representatives.

Sander *et. al.* [SWG+03] followed also the k-means idea to do partitioning. Specifically, they have divided their partitioning technique in two phases, assuming that an initial estimation of seed triangles is present.

Phase 1: All selected seed triangles grow simultaneously using the Dijkstra algorithm on the dual graph of the mesh with edge weights that are dependent on the geodesic distance of the dual triangles and the angle between the average normal of the already generated part and the triangle added. This phase stops when all triangles are labeled, belonging to distinct parts of the object.

Phase 2: The seed triangles are refined by starting from the partitioning boundaries using again the Dijkstra algorithm on the dual graph of the mesh with weights that depend only on the geodesic distance of the triangles. The last face reached by the Dijkstra algorithm becomes the new seed.

Processes 1 & 2 are repeated until there can be no additional refinement of the seed triangles.

The initial seed triangles are generated incrementally by applying phases 1 and 2 initially for one seed. A new seed is added at the end of process 1, this new seed is the last assigned to a part during region growing. When the required number of seeds are generated phases 1 and 2 are repeated until convergence.

Garland *et. al.* [GWH01] propose a hierarchical clustering methodology suitable for applications that require close to planar parts. Specifically they do edge contraction on the dual graph of the mesh. With each edge contraction on the dual graph a new node on the hierarchical structure is generated whose children are the nodes incident to the edge and the two areas that represent the children nodes are grouped into their parent common cluster. The edge contraction cost is based on a planarity, orientation bias and shape bias measures. The contraction sequence is settled by a priority queue which sorts the dual edges of the dual graph according to the contraction cost. The hierarchical structure leaves are the faces of the mesh and at its higher levels the nodes represent areas of triangles (parts of the object) which are grouped together by edge contraction.

Attene *et. al.* [AFS06] used the same hierarchical clustering to classify the mesh triangles into a clusters that can be approximated by a plane or sphere or cylinder primitive. The contraction cost of a dual edge is the minimum of the errors generated after fitting the plane, sphere and cylinder primitives to the triangles of the merged regions. The primitive corresponding to the minimum fitting error approximates the unified cluster.

Gelfand and Guibas [GG04] base their hierarchical clustering technique on slippable motions. These are rigid motions which, when applied to the shape, slide their transformed transformed version against the stationery version without forming any gaps. This happens when the instantaneous motion at each point is tangent to the mesh at x . A kinematic surface is a surface which has a slippable motion. A slippable component is a collection of vertices from the mesh which can be approximated by a kinematic surface. Their algorithm segments the input surface into a set of slippable components using a hierarchical clustering technique similar to [GWH01] on the point set of the mesh.

2.3.7 Spectral Analysis

The spectral analysis method uses the eigenvalues of properly defined matrices based on the connectivity of the graph in order to partition a mesh.

Let G be a connected graph with weights w . Its Laplacian matrix is defined as:

$$L = D - A,$$

where $A=[a_{ij}] = \begin{cases} w_{ij}, & \text{if nodes } i, j \text{ are connected} \\ 0, & \text{otherwise} \end{cases}$, and D is a diagonal matrix with

$$D=[d_{ii}] = \sum_{j \in \text{Neighbors}(i)} w_{ij}$$

The eigenvector y corresponding to the second largest eigenvalue of the Laplacian L can be used to partition the graph G into two equal parts. Specifically, this can be done by thresholding the values of y with their weighted median.

The above partitioning algorithm was one of the algorithms used by Karypis *et. al.* to achieve a hierarchical partitioning of a Graph into k subsets with approximately equal number of nodes. A mesh can be seen also as a graph. Its nodes and edges are the vertices and edges of the mesh respectively.

Karypis *et. al.* [KK95] algorithm (MeTiS [KK98]) has been used by Karni *et. al.* [KARN00] to partition a mesh. They used this segmentation for compression which was accomplished by doing spectral analysis on the normalized Laplacian $D^{-1}L$.

Liu and Zhang [LZ04] used spectral analysis on the dual graph of the mesh. They define an affinity matrix W with elements $0 \leq w_{ij} \leq 1$ denoting the likelihood that faces i, j can be clustered into the same part. These weights are created using the distance function of Katz *et. al.* [KT03] (see section 3.6). They also normalized W to $W_n = D^{-1/2}WD^{-1/2}$, where D is a diagonal matrix defined as $d_{ii} = \sum_{j \in \text{Neighbors}(i)} w_{ij}$.

Then they constructed a normalized matrix Q using the k largest eigenvectors of W_n and apply the k -Means clustering on the row vectors of Q using the standard euclidian distance. Each of the above clustered vectors correspond to the faces of the mesh.

One of the disadvantages of their clustering method is that they emphasize mostly in concavity. This means that they segment sometimes shallow concavities making the quality of segmentation of some areas low.

Zhang and Liu [ZL05] use the affinity matrix as in Liu and Zhang [LZ04] to classify the mesh triangles into two parts using the same methodology as Shi *et. al.* [SM00] in their Normalized Cut algorithm for image processing. In order to lower the complexity the affinity matrix is partially defined. Specifically it is constructed from the distances of two faces (belonging to different parts of the mesh) from all other faces. The two leading eigenvectors of the affinity matrix are extracted using Nyströms [FBCM04] method and are used in a line search algorithm to find the most salient cut which bipartitions the mesh. Based on the salience theory of Hoffman *et. al.* [HR97] they define a measure to repartition the resulting parts in a hierarchical manner.

Liu and Zhang [LZ07] use two kinds of spectral embeddings in order to project the 3D mesh onto the 2D plane. The first embedding uses the Laplacian matrix and it is used to enhance the structural characteristics of the mesh (protrusible parts) and the other embedding uses the minimum curvature information of the vertices and enhances the geometrical characteristics of the mesh (concavities). When the object is projected on the 2D plane using these spectral embeddings the outer outline of the projection is extracted constructing by this way a curve, which they smooth and simplify deriving its closed polygon approximation. By using convexity measures on this polygon they define a criterion on whether the mesh that was projected should be further segmented. If the object should be further segmented they find, using inner distances, two points on the polygon which belong to two different parts of the object. Using these points they approximate their corresponding faces on the mesh and apply the segmentation algorithm followed in Zhang and Liu [ZL05] using these two faces.

2.3.8 Explicit Boundary Extraction Methods

Lee *et. al.* [LLS+04, LLS+05] concentrated on the explicit extraction of the segmentation boundary. Specifically, they threshold minimum negative curvatures using hysteresis thresholding in order to acquire high feature areas. Afterwards they do thinning on the high feature points extracting their skeleton. By removing the branching

points of the skeleton graph a set of non-branching feature contours are acquired. Some of these contours will be selected in order to segment the mesh.

This selection is done by choosing the most central and long boundary contour using the normalized protrusion function of Hilaga *et. al.* [HSKK01]. This contour is then closed by finding the shortest path on the mesh using as edge cost a weighted combination of :

- A Distance function which is small for points that are far away from the contour. This function drives the contour on the other side of the mesh
- A Normal function which is small for normals that face in opposite directions from the contour. This function also drives the contour on the other side of the mesh
- A Centricity function which forces the contour to be perpendicular to the medial axis of the mesh
- A Feature function which drives the contour near feature points

Sometimes the contour can also be closed manually by 2D line drawing on the mesh.

The validity of this contour is checked upon the saliency of the segmented part [HR97]. Specifically the part segmented by the selected contour has to pass a salience test based on its relative size, boundary strength and degree of protrusion [HR97].

If the contour pass the salience test then it is further refined using a geometric snake. Thus a smooth partitioning boundary is extracted. This partitioning boundary scissor the mesh into two parts. Figure 2.3.10 illustrates their method.

The same process is then repeated on each of the extracted parts in order to acquire further partitioning.

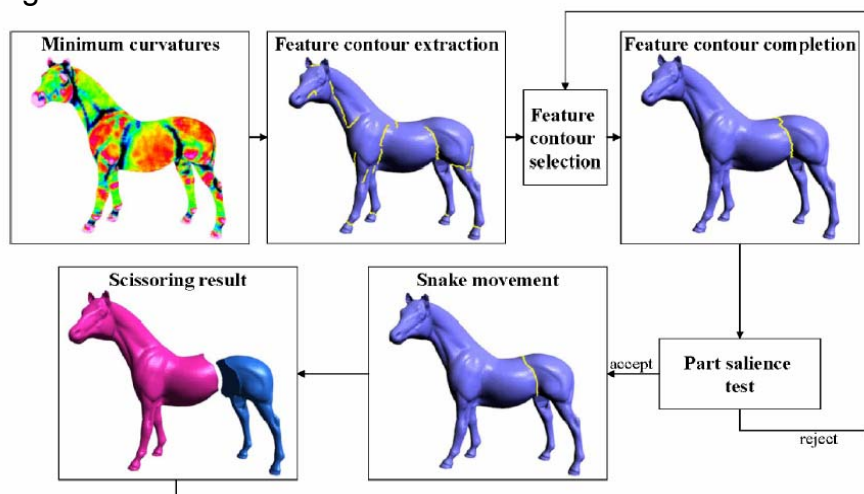


Figure 2.3.10. Lee *et. al.* Boundary Extraction method (taken from [LEE05])

One of the main advantages of their method is that it produces smooth and closed partitioning boundaries accepted also by a salience test. Sometimes though, the accepted contour may not cut the mesh into meaningful components with a consequence of the need of a user to cut the mesh manually.

2.3.9 Volumetric Methods

Shapira *et. al.* [SSCO05] defined a scalar function on the mesh which they call the *Shape Diameter Function*. Specifically its value at a point p on the mesh is equal to its *neighborhood-diameter*. The neighborhood-diameter of a point on the mesh is the average of the distances of a set of rays from the surface cast in opposite directions from its normal, see Figure 2.3.11. This function is further smoothed by local averaging.

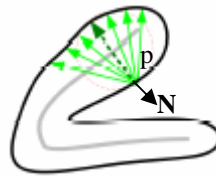


Figure 2.3.11. The normal N of point p and the rays cast from it.

The shape diameter function takes a characteristic value for different parts of the object and changes smoothly when there is a transition between them. Thus different parts can be segmented by finding an appropriate separating value which creates an iso-contour on the surface separating the mesh parts.

Kim *et. al.* [KYL05] also created a Volumetric method to partition the object. Specifically they first did a Voxelization of the object and then they applied the opening morphological operations using the ball-shaped structuring element. Their aim is to extract the convex parts of the object.

In order to achieve this they have created a three stage decomposition scheme.

At the *first stage* they do an initial decomposition of the volumetric object by applying the opening morphological operation using the ball-shaped structuring element. Its radius is chosen so that parts with maximum convexity are extracted.

The part that is extracted from the opening operation is called the *body class* and the parts that belong on the complement of this operation are called the *branch class*, see

Figure 2.3.12. They also apply a post processing scheme in order to discard very small branch parts and to straighten their boundaries.

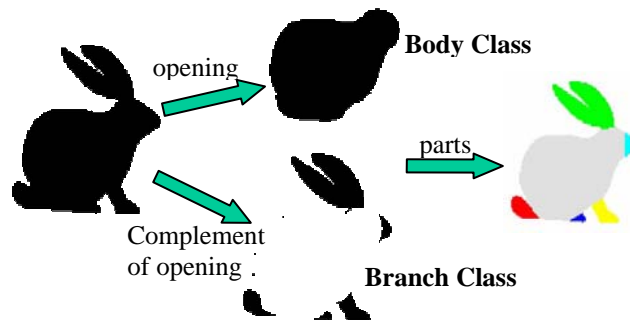


Figure 2.3.12. Applying the opening morphological operator to the bunny voxelized object (taken from [KYL05])

At the *second stage* they apply recursively the same morphological operation on the already extracted parts using as rule that a part belonging to the branch class should be further segmented if at least two parts are generated and each of them have only one adjacent part, see Figure 2.3.13.



Figure 2.3.13. Segmentation of the ears of the bunny object. Three new components are generated and two of them are adjacent to only one part (taken from [KYL05])

As a final *third stage* they merge together parts with similar convexities, see Figure 2.3.14.



Figure 2.3.14. Merging parts with similar convexities (taken from [KYL05])

One advantage of their decomposition scheme is that it can be applied to practically all of the 3D models due to the voxelization, which treats all objects in the same manner. A disadvantage of their method is that due to the voxelization resolution some parts that should not be connected might become connected, thus producing an erroneous segmentation

2.3.10 Critical Points

Algorithms that follow this method use critical points defined on the mesh to guide the segmentation process.

Katz *et. al.* [KLT05] presented a segmentation method based on critical points defined on the mesh. They first transform the mesh into a pose invariant representation using the multi-dimensional scaling method (MDS) [CC94]. Then they find on this representation critical points. These critical points are local maxima of the protrusion function (section 2.3.3) that belong on the convex hull of the representation, see Figure 2.3.15(b). Afterwards they create a surrounding sphere of the representation and use it to mirror its points so that vertices of the core become external points of the sphere, see Figure 2.3.15(c). The points that belong on the convex hull of this mirroring consist the central part of the object which they call the core see Figure 2.3.15(d). These points and their incident triangles are extracted isolating all the critical points (with the exception of those that are close to each other).

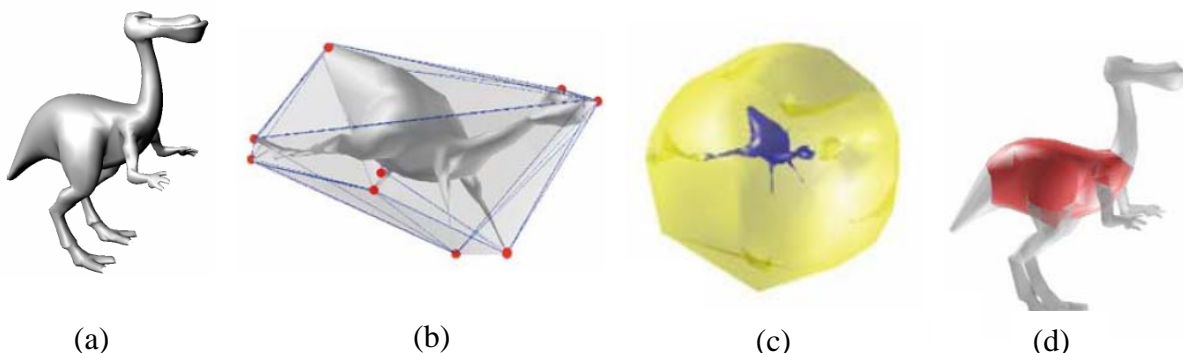


Figure 2.3.15. (a) The original Model, (b) The pose invariant representation of the model and its critical points (red colour), (c) Mirroring of the pose invariant representation, (d) The extracted core of the object (Red color).

At the end the segmentation is further refined by a graph cut method that cuts on the concavities of the object similarly to [KT03].

This process is repeated hierarchically to acquire further levels of detail as was done also in [KT03].

Zhou *et. al.* [ZH04] used also critical points in order to achieve segmentation. These critical points are local extremum and saddle points of a function defined on the points of the mesh. This function is defined as the geodesic distance of a point from a reference root point which can be selected either manually or automatically (it is usually placed at an extremity of the object).

The local extremum points are sorted in descending order according to their geodesic distance. Afterwards by traversing this ordered list a shortest path graph is constructed starting from each of the extremums. The construction of this graph stops when a saddle point or an already labeled point is found and all the unlabeled points of the graph are labeled with a part label number. At the end only vertices belonging to the main body will remain unvisited, so a label number is also given at all of these points, see Figure 2.3.16.

Some disadvantages of their technique are that it only segments manifolds of genus 0, it is susceptible to noise, since some critical points may be erroneously found on noisy meshes and the decomposition result depends on the choice of the root vertex.

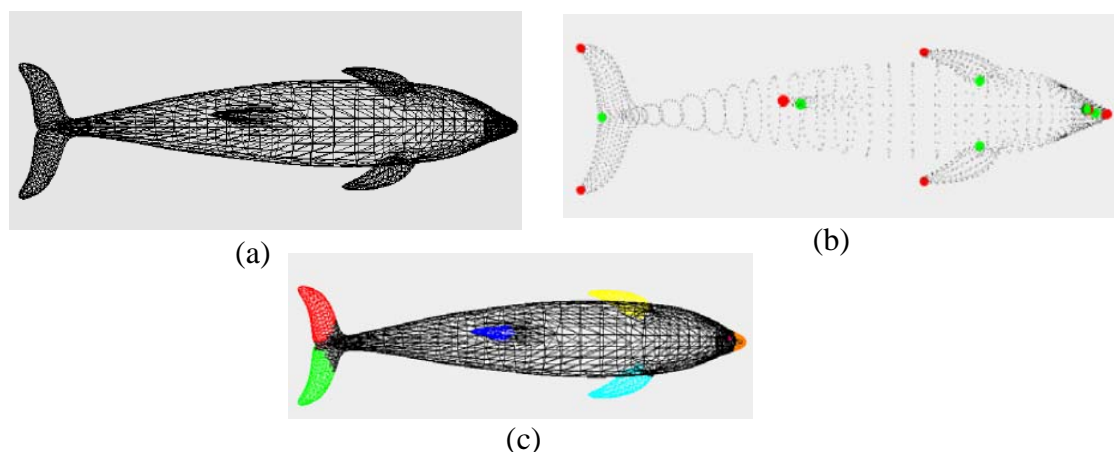


Figure 2.3.16. (a) The mesh to be segmented, (b) The Extremum (red), Saddle (green) points of the mesh, (c) Labeled segmented parts after the application of their segmentation scheme (here color is used to denote the labeling).

Lin *et. al.* [LLL07, LLL04] also use critical points on the mesh. They call them salient representatives. These representatives are local maximums of the protrusion function defined on the DCG of the mesh. More formally:

$$r \text{ salient Representative} \Leftrightarrow \forall u \in W_r, pf(r) > pf(u) \quad (2.3.8)$$

where W_r a geodesic neighborhood of r and pf the protrusion function. Figure 2.3.17(a) shows these salient representatives on a 3D Mesh.

Each representative create L geodesic zones called locales which have the following definition:

$$L^x = \{x e \leq D(u) < (x + 1)e, x \in \{0, \dots, L-1\}\} \quad (2.3.9)$$

where $D(u)$ a modified geodesic distance from u to the salient representative and e the width of the locale zone, see Figure 2.3.17(b).

Since the locales of various representatives intersect each other a Base K is introduced to constrain the locales and is defined as :

$$K = \{v \mid \forall v \in V, pf(v) \leq thr_b\} \quad (2.3.9)$$

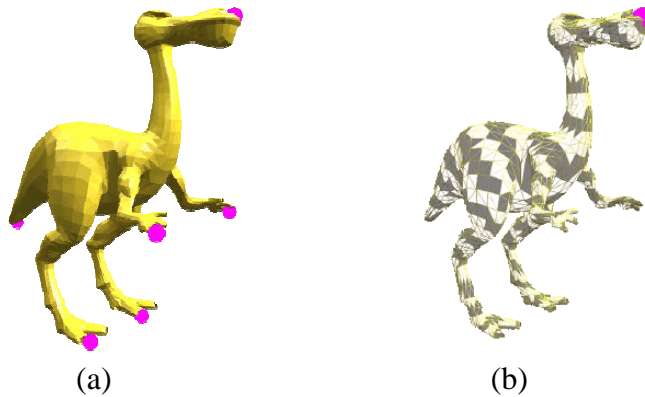


Figure 2.3.17. (a) The salient representatives of the above mesh colored in pink, (b) The locales generated by the representative colored in pink

Where pf the protrusion function value of a node in V , and thr_b a user defined threshold, see Figure 2.3.18. Each of these trimmed locales contains the boundary of the protrusion represented by the salient representative.

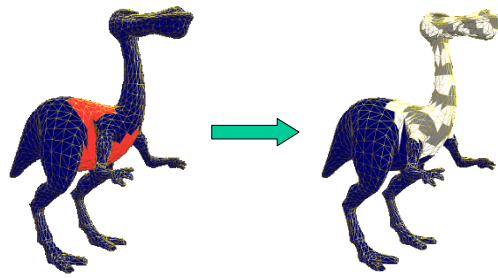


Figure 2.3.18. The red region is the Base and constrain the locales of Figure 2.3.18 and the output are the locales shown on the right of Figure 2.3

A border function is then defined based on these locales which identifies those containing the border of the protrusion. After finding these locales the minimum cut algorithm of Katz et. al. [KT03] is used to separate the protrusion from the rest of the object.

An advantage of their method is that it can work on noisy meshes also. A disadvantage of their work is that it uses several parameters that are not fixed.

Chapter 3

Proposed 3D mesh segmentation algorithm

After a thorough study of the presented methods of Chapter 2 as well as from experimental results in some of them the critical points methods was adopted in this work. This choice was made because it was assessed that the results produced with this method were more meaningful in comparison with the results produced by other methods. This can be ascribed to the fact that the critical points methods utilize the human ability to distinguish the main particulars of an object (protrusible parts, critical and salient points, boundaries etc.) and to introduce this perception suitably into the algorithm.

Therefore in this dissertation a new segmentation algorithm that belongs into the critical points methods will be presented. It produces high quality segmentation results and is very competitive in comparison to state of the art segmentation algorithms.

The contribution of the presented work is twofold :

- i. A novel way to trace the partitioning boundaries of the 3D object using closed boundaries constructed with the aid of a distance function is presented;
- ii. A novel algorithm for the core approximation of the 3D object is introduced.

3.1 Proposed Surface Segmentation Methodology

In this section the proposed surface segmentation methodology is going to be presented. The whole philosophy is based on the premise that the 3D object consists of a main body and its protrusible parts. For example in Figure 3.1.1 several 3D objects are partitioned in this manner, all of them consisting of a main body and protrusible parts.

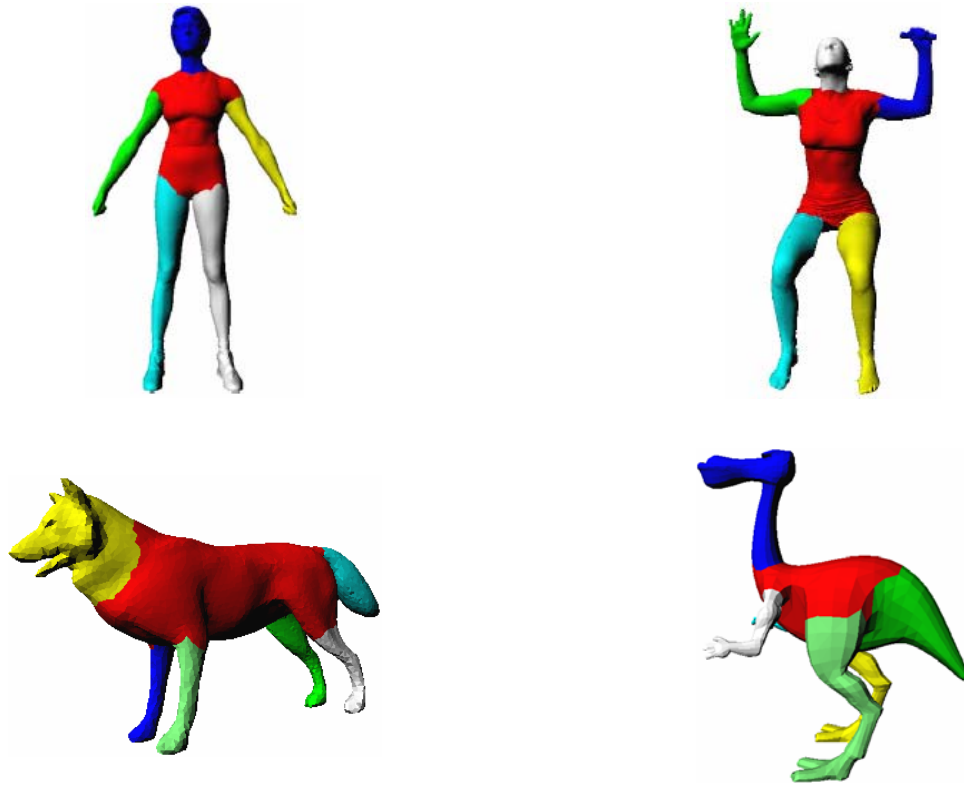


Figure 3.1.1. 3D objects consisting of a main body (coloured in red) and protrusible parts

The proposed segmentation methodology goal is to extract all of these components. Figure 3.1.2 illustrates the flowchart of the proposed surface segmentation.

The input that the algorithm accepts is a 3D mesh representing a 3D *manifold*. In the discrete case a mesh represents a 3D manifold when each of the edges of a triangle in the mesh is incident with at most one other triangle. *The procedure is as follows:*

- (i) Initially the salient points of the mesh are extracted; these points characterize the protrusions of the mesh.
- (ii) The salient points are further grouped according to their geodesic proximity where each cluster represents a main component of the object and each of them is assigned a unique representative salient point.
- (iii) Next the core (main body) of the mesh is approximated using the minimum cost paths that the aforementioned representatives create with each other.

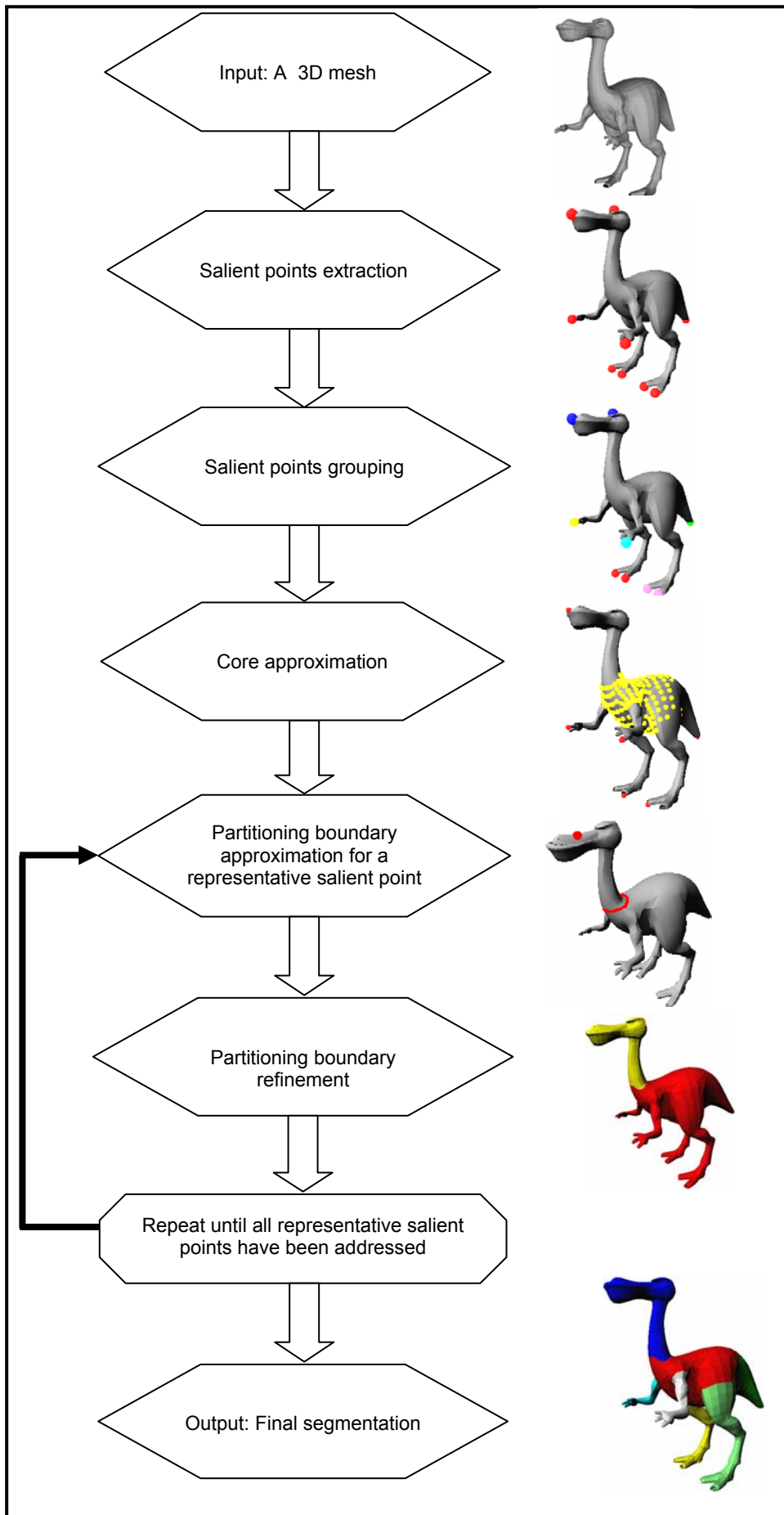


Figure 3.1.2. The flowchart of the proposed surface segmentation algorithm

- (iv) In the sequel the boundary between the core and each of the protrusible parts (Partitioning Boundary) is approximated using closed boundaries which span the area containing the true partitioning boundary.
- (v) Finally the approximated partitioning boundary is refined using the minimum cut algorithm of Katz and Tal [KT03].

All of the above mentioned stages of the proposed segmentation methodology will be detailed in the following sections.

3.1.1 Salient points extraction

The salient points of the mesh \mathbf{M} reside at the tips of the protrusions. In order to find them the protrusion function pf (see section 2.3.3) will be used. As mentioned in section 2.3.3 this function receives low values at the center and high values at the protrusible parts of the 3D object represented by the 3D mesh. Thus, it is natural to search for the salient points at the local maxima of the protrusion function.

In order to find the local maxima of the protrusion function the local areas in the mesh should be defined so as to search in them the local maxima of the function. Specifically for each vertex $v \in \mathbf{M}$ a neighborhood of vertices N_v is considered.

The neighborhood N_v can be either:

- i. a k -ring neighborhood* defined as the set of vertices within k edges away from vertex v ;
- ii. a geodesic neighborhood* defined as the set of vertices for which the geodesic distance from vertex v is less than a threshold. This threshold is called the radius of the geodesic neighborhood

In this work the geodesic neighborhood will be used with radius $\sqrt{5 \cdot 10^{-3} \cdot \text{area}(\mathbf{M})}$ as in [LLL07]. The geodesic neighborhood is computed by using the Dijkstra algorithm.

Formally, a salient point $v \in \mathbf{M}$ is defined as:

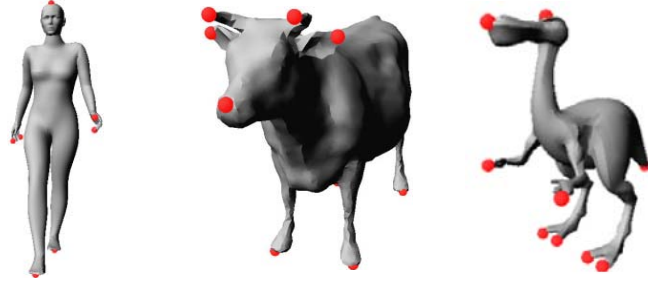
$$v \text{ is a salient point} \Leftrightarrow \begin{cases} pf(v) > pf(v_i) & \forall v_i \in N_v \\ pf(v) > t_{prot} & pf \text{ normalized in } [0,1] \end{cases} \quad (3.1.1)$$

Definition (3.1.1) ensures that the salient point v will reside at the tip ($pf(v) > pf(v_i), \forall v_i \in N_v$) of a protrusion ($pf(v) > t_{prot}$). The threshold t_{prot} is experimentally defined and will be set to a constant value in the experiments section (Section 3.2). Figure 3.1.3(a) illustrates the salient points found in several objects using definition (3.1.1).

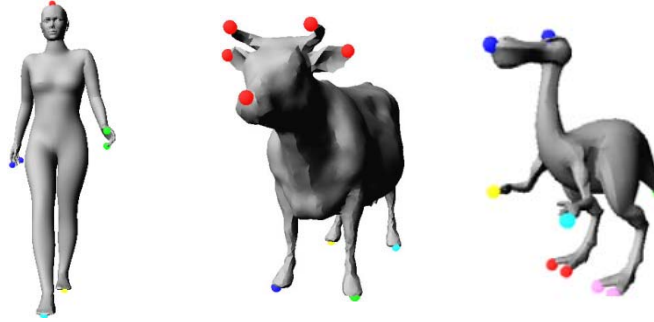
3.1.2 Salient points grouping

The proposed mesh segmentation algorithm aims to extract the meaningful parts of the object. It often happens that the extracted salient points belong to sub-parts of the objects. For example in Figure 3.1.3(a), there exist salient points that correspond to fingers in the 'human' model, as well as ears, horns and nose in the 'cow' model and toes, head, nose in the 'dinopet' model. Since the proposed algorithm uses the salient points to extract the protrusions of the object, it is necessary to group them, each one of the groups representing a single part of the object. Thus, given the example of Figure 3.1.3(a), the salient points in each of the hands of the 'human' model need to be gathered in one group in order to represent its arms, the salient points on the head of the 'cow' model should be gathered in one group in order to represent its head and the salient points on the feet and head of the 'dinopet' model should be gathered in groups in order to represent the objects feet and head respectively.

The salient points that are required to be grouped are those which are close to each other in terms of geodesic distance. In order to achieve this grouping, we use half of the mean of the geodesic distances between the salient points as a threshold T_s and group the salient points for which the geodesic distance is less than T_s .



(a)



(b)

Figure 3.1.3. Example of 3D meshes ('human', 'cow', 'dinopet') with their corresponding salient points at the (a) extraction stage (red dots) and (b) Grouping stage - each color represents a different group

Formally, let assume that $S = \{s_i, i = 1, \dots, N_s\}$ be the set of the salient points of the mesh, then the threshold T_s is defined as :

$$T_s = \frac{\sum_{i=1}^{N_s-1} \sum_{j=i+1}^{N_s} g(s_i, s_j)}{N_s(N_s - 1)} \quad (3.1.2)$$

where N_s denotes the number of the salient points and $g(s_i, s_j)$ denotes the geodesic distance between the salient points s_i and s_j .

A group C of salient points is defined as:

$$C = \{s_i \in S : \forall s_k \in C, g(s_i, s_k) < T_s\} \quad (3.1.3)$$

The salient point with the highest protrusion value is chosen as a *representative* of each group C , i.e. $Rep(C) = \{s_i \in C : pf(s_i) > pf(s_k), \forall s_k \in C\}$.

The efficiency of the proposed grouping method in representing the main parts of the 'human', 'cow' and 'dinopet' models is shown in Figure 3.1.3(b).

3.1.3 Core approximation

The core approximation is the process where the main body of an object is approximated. An algorithm which approximates the main body of the object is the one that can acquire all the elements (vertices or faces) of the mesh (representing the object) except those that belong to the protrusions of the mesh. These elements should separate all the protrusions from each other.

In Katz *et. al.* [KLT05] this is achieved by spherical mirroring of the pose invariant representation of the mesh. When they cannot achieve the separation entirely, they proceed in core extension until all features are separated.

In Lin *et. al.* [LLL07] a simple thresholding of the protrusion function is used, i.e. they define that the core of the mesh are the faces whose value of protrusion function (defined on the centroids of the faces) is lower than a predefined threshold. However, this thresholding is not reliable since it might lead to a significant overestimation or underestimation of the main body. In Figure 3.1.4 Lin *et. al.* [LLL07] core approximation is illustrated on the 'human' and 'cow' 3D objects. It can be observed that the main component (core) of the objects is significantly underestimated.

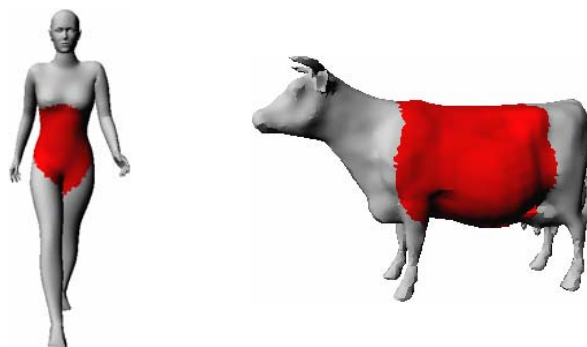


Figure 3.1.4. The core approximation of Lin *et. al.* on the 'human and 'cow' 3D objects. The triangles representing the core component are coloured in red

In the proposed algorithm, the core approximation is addressed by using the minimum cost paths between the representative salient points. Let us assume that $\hat{S} = \{\hat{s}_i, i = 1, \dots, N_C\}$ be the set of representative salient points, where N_C denotes the number of clusters found in section 3.1.2 and \hat{s}_i the representative of the i^{th} group. Also, let $P = \{P_{ij}, i, j \in \{1, \dots, N_C\}\}$ be the set of all minimum cost paths of the points of \hat{S} , where P_{ij} denotes the minimum cost path between \hat{s}_i, \hat{s}_j .

The key idea of the proposed core approximation algorithm is to expand a set of vertices in ascending order of protrusion function value until the expanded set touches a certain percentage of all elements of P . The motivation of this approach stems from the fact that the minimum cost paths cover a significant amount of the protrusible parts, thus by expanding a set of vertices by this way gives a guarantee that it will reach the protrusible parts and cover also an area inside them. The pseudo-code for the core approximation algorithm is shown in Table 3.1.1.

First, the vertices of the mesh \mathbf{M} are inserted in a priority queue *PFHeap*; in which the vertices with the minimum protrusion function inserted first. The vertices of the core approximation are stored in a list, named *CoreList*. The algorithm proceeds by extracting points from the priority queue which incrementally expand *CoreList*. Every point extracted is examined whether it belongs in P . A path P_{ij} in P remains *active* if the ratio of the number of vertices in the path P_{ij} which have been visited during expansion over the total number of vertices is less than t_c , where t_c has been experimentally defined. This threshold denotes the aforementioned percentage of the points of the minimum cost path that the core approximation can touch. Using this threshold it is expected that the core approximation will cover even slightly the protrusible parts that are traversed by the minimum cost path traces. A salient point $\hat{s}_i \in \hat{S}$ remains *active* if $\exists P_{ij}$ for some $j \in \{1, \dots, N_C\} \neq i : P_{ij}$ active. A vertex v of the Mesh *CanBeAdded* in *CoreList* if the nearest salient point in \hat{S} is active. *StopGrowing* becomes 'TRUE' when all salient points become non-active.

The proposed core approximation method has two main advantages over [KLT05]:

- i. There is no need to do multidimensional scaling, which is a time consuming process, in order to extract the core. Instead only the minimum cost paths are

used in order to check whether the core has expanded sufficiently. This implies far less complexity;

```

1:  for all vertices  $u \in M$  do
2:      insert  $u$  into  $PFHeap$  with priority  $pf(u)$ 
3:  end for
4:   $StopGrowing = false$ 
5:  while  $!StopGrowing$  do
6:      pop a vertex  $u$  from  $PFHeap$ 
7:      if  $u$   $CanBeAdded$  then
8:           $CoreList.add(u)$ 
9:      end if
10:     for all  $P_{ij} \in P$  do
11:         if  $P_{ij}.active$  then
12:             if  $u \in P_{ij}$  then
13:                 increment  $P_{ij}.counter$ 
14:                 if  $\frac{P_{ij}.counter}{P_{ij}.SizeOfPath} \geq t_c$  then
15:                      $P_{ij}.active = false$ 
16:                 end if
17:             end if
18:         endif
19:     end for
20:     for all  $\hat{s}_i \in \hat{S}$  do
21:         if  $\hat{s}_i.active$  then
22:              $\hat{s}_i.active = false$ 
23:             for all  $\hat{s}_j \in \hat{S} - \{\hat{s}_i\}$  do
24:                 if  $P_{ij}.active$  then
25:                      $\hat{s}_i.active = true$ 
26:                 end if
27:             end for
28:         end if
29:     end for
30:      $//StopGrowing$  becomes true if all  $\hat{s}_i$  become non active
31: end while

```

Table 3.1.1. The pseudo-code of the proposed core approximation algorithm

- ii. We have introduced a percentage of minimum cost path traces that should be covered for the termination of core expansion. Those traces span the protrusible parts at most. Thus, the selection of a percentage of the traces provides a high confidence that the core points will cover areas of the protrusible parts or being very close to the neighboring areas in which the real boundary is situated. Several examples are given in the experiments section (section 3.2)

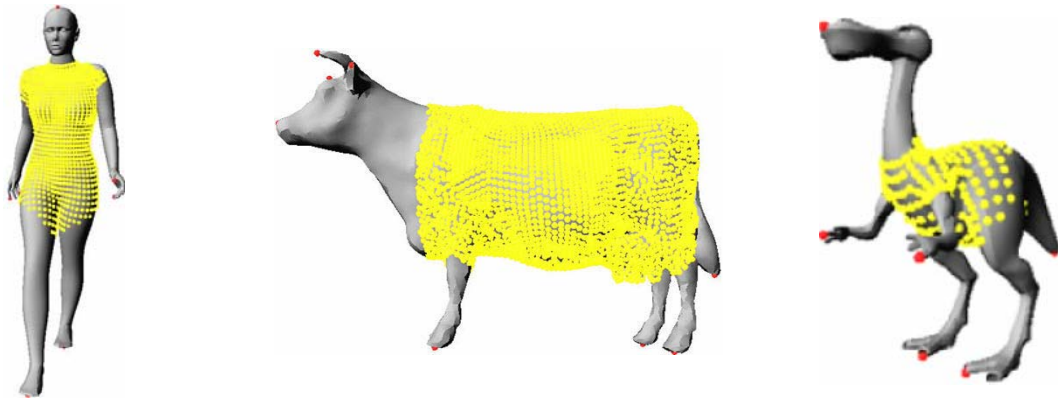


Figure 3.1.5. Examples of core approximation for the 3D objects 'human', 'cow' and 'dinopet'. The vertices representing the core are coloured in yellow

Figure 3.1.5 illustrates the proposed core approximation on the 'human', 'cow' and 'dinopet' 3D objects. As it can be observed, the algorithm produces consistent approximation of the core and its boundaries are near the actual boundaries identifying the initial approximation of the partitioning boundaries.

In general, it can not be guaranteed that the core approximation overlaps exactly the partitioning boundaries. This happens because the minimum cost paths reside mostly in the protrusions. Thus, a further step is required that can detect the partitioning boundaries.

3.1.4 Partitioning Boundary Detection

The partitioning boundary is the boundary between a protrusion and the main body of the mesh. It is considered that in the area that is enclosed by the desired boundary between the protrusion and the main body, an abrupt change in the volume of

the 3D object should occur, see Figure 3.1.6(a). The goal is to detect this abrupt change. To accomplish this, we construct closed boundaries which span the area containing the partitioning and are defined by a distance function D associated to a salient representative of the group which represents the protrusion, see Figure 3.1.6(b). The abrupt change of volume is detected by examining the closed boundaries perimeter.

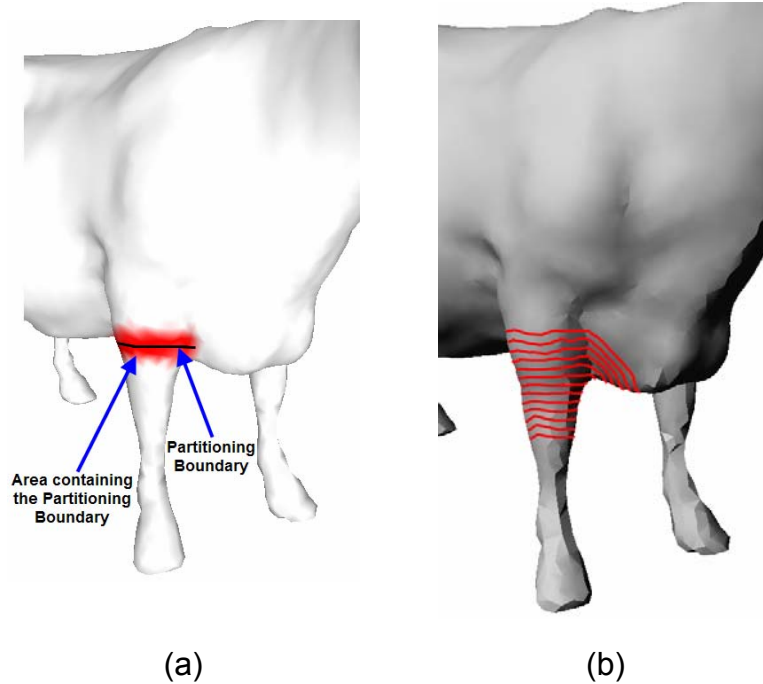


Figure 3.1.6. (a) The area containing the Partitioning boundary (coloured in black) separating the foot of the ‘cow’ 3D object from the main component is coloured in red, (b) The closed boundaries which span the area where the partitioning boundary lies are shown.

For each representative salient point \hat{s} , the distance function D is defined for each vertex v of the mesh as the shortest distance between v and \hat{s} . The shortest distance is computed using the Dijkstra algorithm with source \hat{s} and cost for each edge (u,v) of the mesh denoted as :

$$\text{cost}(u,v) = \delta \frac{\text{length}(u,v)}{\text{avg_length}} + (1 - \delta) \frac{\text{prot}(u,v)}{\text{avg_prot}} \quad (3.1.4)$$

where $\text{prot}(u,v) = |pf(u) - pf(v)|$ and avg_length , avg_prot denote the average values of the length and protrusion difference of the edges of the mesh, respectively. This distance function was introduced in [LLL07]. A proper balance between the two terms of

equation (3.1.4) creates a closed boundary whose approximating plane is nearly perpendicular to the principal direction of the corresponding protrusion. In this work, δ is set equal to 0.4.

The closed boundaries which are used in the proposed algorithm are constructed by interpolating on the edges of the mesh the iso-contour generated by setting a constant value D_c on D . The iso-contour C_{D_c} intersects the edge (u,v) of the mesh if $(D(u) - D_c)(D_c - D(v)) > 0$ and the intersection point is found by linear interpolation and is defined as $v_{int} = (1 - \lambda)u + \lambda v$, where $\lambda = \frac{D_c - D(u)}{D(v) - D(u)}$. Figure 3.1.7 shows the interpolation process.

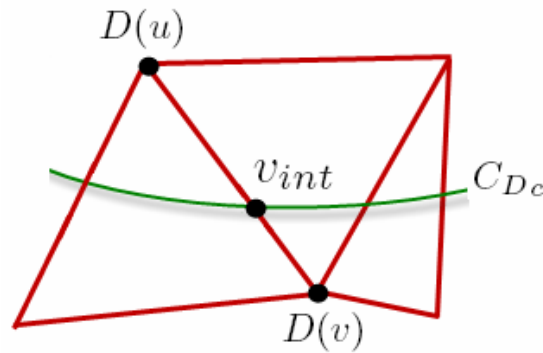


Figure 3.1.7. Interpolation process

By tracing all the triangles which are incident to all the intersecting edges, a linear approximation of this iso-contour can be constructed which in most of the cases is a single closed boundary. In the case that more than one closed boundaries are generated, we choose the one with the largest perimeter. The perimeter of a closed boundary is defined as the sum of the length of the edges which are contained in the closed boundary. In Figure 3.1.8, an illustration of a closed boundary constructed by the method described above is shown.

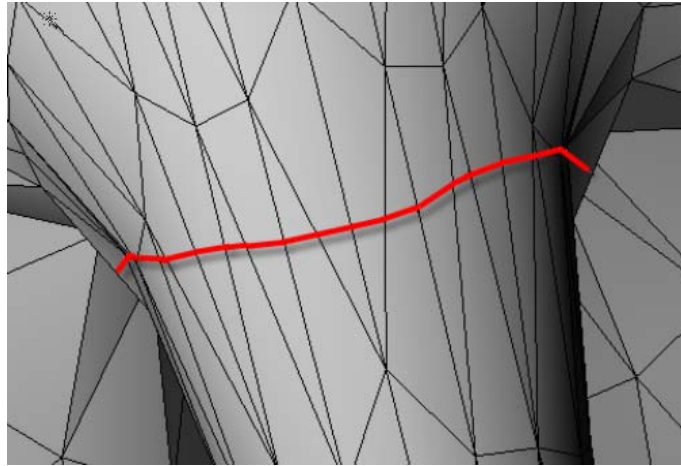


Figure 3.1.8. The closed boundary approximating an iso-contour of the distance function D

As already mentioned in section 3.1.3, the core approximation has its boundaries near the actual boundaries of the distinct parts of the model. Taking advantage of this an area containing the partitioning boundary can be created. Specifically, this area is defined by the arithmetic interval $[(1-d_1)D_{\text{coremin}}, (1+d_2)D_{\text{coremin}}]$, where D_{coremin} denotes the value of the distance function between the nearest point of the core approximation and the salient representative \hat{s} , d_1 , d_2 denote the extent of the interval ($0 < d_1 < 1$, $d_2 > 0$) and have been defined experimentally, see Figure 3.1.9(a).

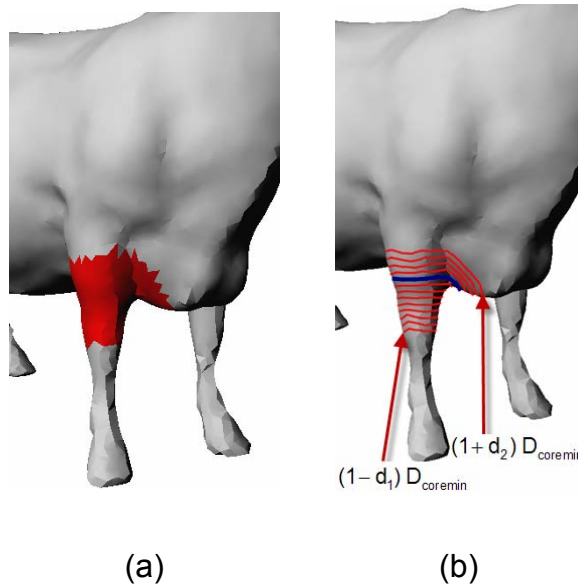


Figure 3.1.9. (a) The area defined by the arithmetic interval $[(1-d_1)D_{\text{coremin}}, (1+d_2)D_{\text{coremin}}]$, (b) The closed boundaries sweeping the area of (a), the approximation of the protrusion boundary is shown in blue

For the approximation of the partitioning boundary, the area defined by the aforementioned arithmetic interval is swept by a predefined number I_{per} of closed boundaries in fixed steps equal to $e = \frac{(d_1 + d_2)D_{coremin}}{I_{per}}$, see Figure 3.1.9(b). Sweeping of the area will be terminated when the change of the perimeter between successive closed boundaries will be greater than a threshold. Specifically, let per_i be the perimeter of the i^{th} closed boundary. The following ratios are defined:

$$r_i = \begin{cases} \frac{per_{i+1}}{per_i} & \text{if } per_{i+1} > per_i \\ \frac{per_i}{per_{i+1}} & \text{if } per_{i+1} \leq per_i \end{cases}, i = 1, \dots, I_{per} \quad (3.1.5)$$

These ratios represent the change of the perimeter between successive closed boundaries. The algorithm searches for the r_k which is the first ratio that is greater than a threshold r_{max} that is defined experimentally and sets the k^{th} closed boundary as the partitioning boundary. If such a ratio cannot be found, the algorithm finds the maximum ratio r_{kmax} and sets the k_{max}^{th} closed boundary as the partitioning boundary. In this way, we detect the aforementioned abrupt change in the volume of the object at the boundary between its main body and the protrusion.

Initially, in the proposed methodology to approximate the partitioning boundary the simple solution is chosen in order to select the representative of the group, defined in section 3.1.2, as the source point from which the distance function D is computed. This choice may lead to the creation of skewed closed boundaries near the real boundary of the protrusion. This has the consequence that the algorithm may not be able to trace the real partitioning boundary properly as it will largely deviate from it. See Figure 3.1.10(b) for a demonstration of this problematic case.

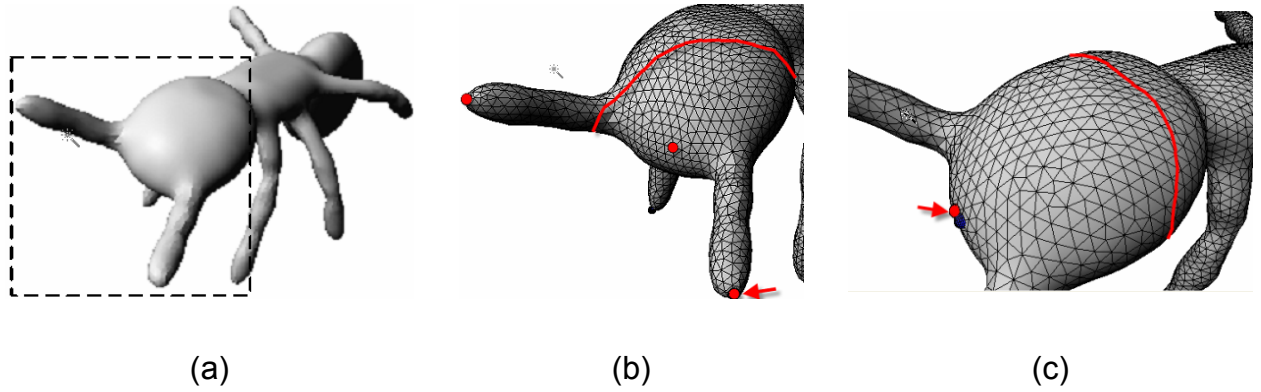


Figure 3.1.10. (a) The original 3D model ‘ant’, (b) Skewed closed boundary generated by the salient representative, indicated by an arrow, of the group denoted by the red spheres, (c) The closed boundary generated by the indicated refined representative

In the sequel, it will be presented how to find a proper *refined representative* \hat{s}_r in order to be used instead of the representative of the group. The proper refined representative should create closed boundaries that are parallel to the protrusion boundary like the one presented in Figure 3.1.10(c).

In order to find the proper refined representative in a group of salient points we proceed as follows.

First, we find the vertex s_{\min} of the mesh for which the distance to all of the salient points in the group is minimal. Then, we find the point c_{\min} of the core approximation with the minimum geodesic distance from s_{\min} . The geodesic distance of s_{\min} and c_{\min} is denoted as d_{\min} . Afterwards, the point p_{thres} is found which is the first point on the minimum cost path from s_{\min} to c_{\min} for which the geodesic distance from s_{\min} is greater than $0.3d_{\min}$. Next, the iso-contour $C_{p_{\text{thres}}}$ is considered, which is generated by the protrusion function by setting it to the constant value p_{thres} . The part of the iso-contour which belongs to the protrusion of the object being examined is close to the salient points of the group and in most of the cases its best fit approximating plane is nearly perpendicular to the principal direction of the protrusion. $C_{p_{\text{thres}}}$ is approximated using the same interpolating technique discussed above.

Next it is taken into account only the part of the mesh which contain s_{\min} and is constrained by $C_{p_{\text{thres}}}$ for which we compute the protrusion function. The *refined*

representative \hat{s}_r is the point of the constrained mesh with the minimum value of the protrusion function.

In Figure 3.1.11 the process of finding the refined representative is illustrated. The salient points that belong to the group representing the head are the red spheres. The yellow sphere represents s_{\min} and the red line approximates the iso-contour $C_{p_{\text{thres}}}$ (the red line is generated by the triangles that $C_{p_{\text{thres}}}$ intersects). The refined representative \hat{s}_r is the point of the constrained mesh (coloured in blue) with the minimum protrusion value and is illustrated by the green sphere.

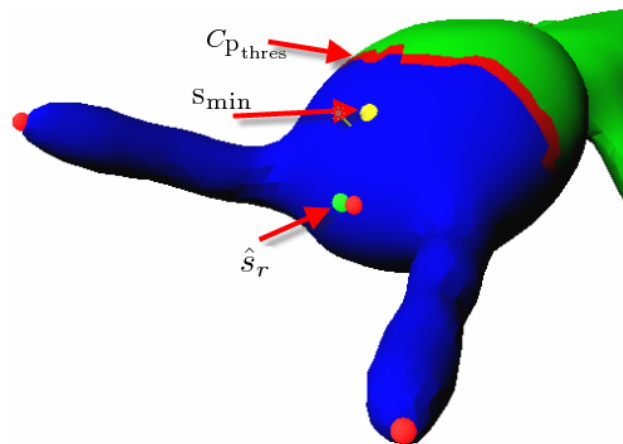


Figure 3.1.11. Refined representative selection (green sphere)

3.1.5 Partitioning Boundary refinement

The partitioning boundary detected in section 3.1.4 is an iso-contour generated by the distance function D approximating the true partitioning boundary. In most of the cases this approximation is rough meaning that it deviates from the true partitioning boundary. As mentioned in section 3.1.4 the partitioning boundary is delimited at the area where there is a sudden change in the volume between the main body and the protrusion. Also according to Hoffman and Richards theory [HR84] it resides at the concavities of the object. The partitioning boundary approximation described in section 3.1.4 is not constrained to the concavities where the true partitioning boundary passes, thus there is a need to refine the partitioning boundary approximation so that it passes also through the concave regions of the 3D object. In Figure 3.1.12(a) the initial

boundary approximation, found by applying the methodology of section 3.1.4, is shown. As it can be observed it does not pass through the concavities of the object as it should.

In order to achieve this, we use the minimum-cut methodology as in Katz and Tal [KT03]. Specifically, a flow network graph is constructed using the dual graph of the mesh. In order to construct the network, three regions should be defined. Specifically, we define a region **A** containing the triangles of the protrusible part of the mesh, a region **C** containing the partitioning boundary and a region **B** containing the faces of the rest of the mesh.

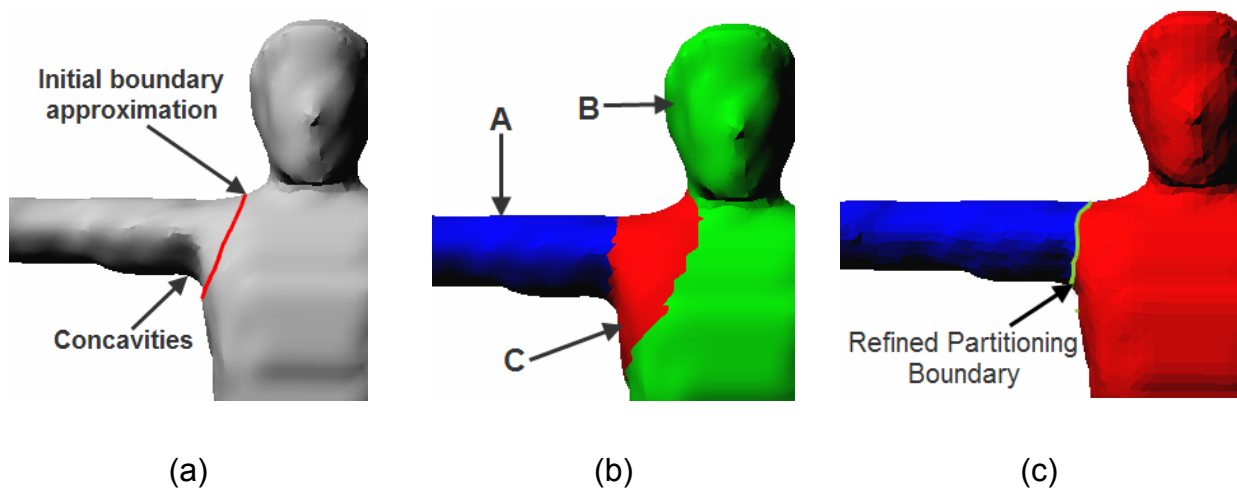


Figure 3.1.12. (a) The initial rough partitioning boundary approximation, (b) region **A** is shown with blue, region **B** is shown with green and region **C** is shown with red, (c) The refined partitioning boundary

Region **C** is constructed as follows. First we find the average geodesic distance, denoted as $AvgGeodDist$, between the initial boundary approximation extracted in section 3.1.4 and the refined representative calculated also in the same section. Then, region **C** is defined as the triangles of the mesh which vertices geodesic distance from the refined representative lie all in the interval $[0.9 \cdot AvgGeodDist, 1.1 \cdot AvgGeodDist]$. This interval denotes a small area around the estimated partitioning boundary where it is expected that the true partitioning boundary should reside (red triangles of Figure 3.1.12(b)). Region **A** is constructed by performing a breadth first search starting from the refined representative of the protrusion until region **C** is reached (blue triangles of Figure 3.1.12(b)).

Let V_C, E_C be the nodes and edges of the dual graph representing \mathbf{C} . Let V_{CA} be the nodes of the graph which represent the triangles of \mathbf{A} that share a common edge with the triangles of \mathbf{C} , and V_{CB} be the nodes of the graph which represent the triangles of \mathbf{B} that share a common edge with the triangles of \mathbf{C} .

The flow network graph $G = (V, E)$ is constructed by adding also two more nodes s, t , and V, E are defined as:

$$\begin{aligned} V &= V_C \cup V_{CA} \cup V_{CB} \cup \{s, t\} \\ E &= E_C \cup \{(s, v), \forall v \in V_{CA}\} \cup \{(t, v), \forall v \in V_{CB}\} \cup \{e_{uv} \in E : u \in V_C, v \in \{V_{CA} \cup V_{CB}\}\} \end{aligned} \quad (3.1.6)$$

A capacity function on E is defined as:

$$\text{Cap}(u, v) = \begin{cases} \frac{1}{1 + \frac{\text{Ang_Dist}(\alpha_{uv})}{\text{avg}(\text{Ang_Dist})}} & \text{if } u, v \neq s, t \\ \infty & \text{otherwise} \end{cases} \quad (3.1.7)$$

$$\text{Ang_Dist}(\alpha_{uv}) = n(1 - \cos \alpha_{uv}) \quad (3.1.8)$$

where α_{uv} denotes the dihedral angle between the two faces which share the edge (u, v) , and $\text{avg}(\text{Ang_Dist})$ is the average angular distance. Note that $n=1$ for concave angles and $n \ll 1$ for convex angles. Using this capacity function, the minimum cut algorithm applied on this network creates a partitioning boundary that passes through the concave regions, see Figure 3.1.12(c).

3.2 Experimental Results

The experimental results which have been made are based on twenty 3D models segmented with the proposed algorithm which are further compared with the results produced by three other popular segmentation algorithms, namely Lin *et. al.* (LIN) [LLL07], Valette *et. al.* (VALETTE) [VKS05] and Kim *et. al.* (KIM) [KYL05]. The results of LIN algorithm were produced by the original authors, VALETTE was implemented in this work, while KIM is the original implementation which is used in the MPEG-7 experimentation model. The reason for considering these segmentation algorithms for comparison purposes is that it is believed that they are representative algorithms

belonging to different methods, i.e. the critical points, the reeb graph and the volumetric methods.

In the comparison, the evaluation criteria set in [AKM+06] will be used, namely, (i) Type of segmentation; (ii) Extracting the “correct” segments; (iii) Boundaries; (iv) Hierarchical / multi-scale segmentation; (v) Sensitivity to pose; (vi) Asymptotic complexity; (vii) Control parameters. Furthermore, the behaviour of the proposed segmentation algorithm will be given in the presence of noise and lastly the core and partitioning boundary approximations for all of the twenty 3D models will be presented showing by this way the robustness of the segmentation algorithm.

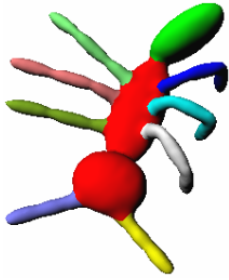
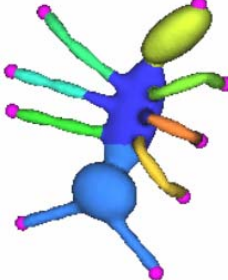
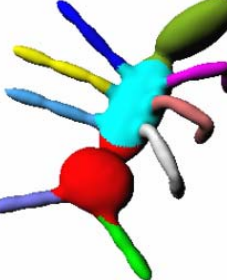

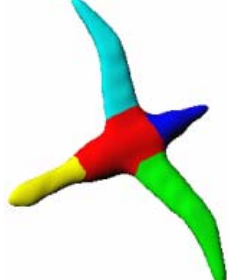
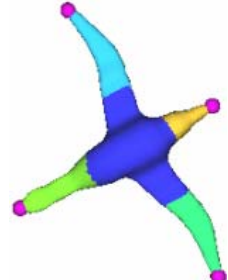
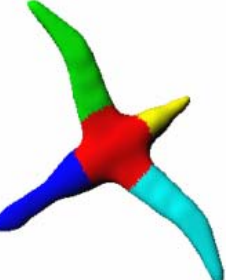

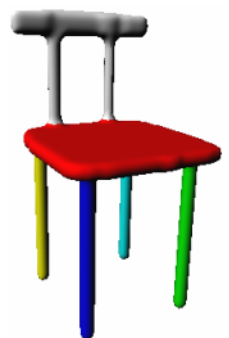
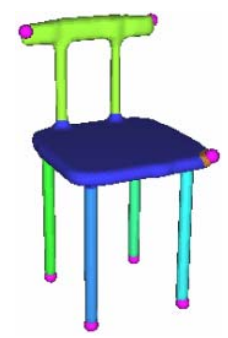
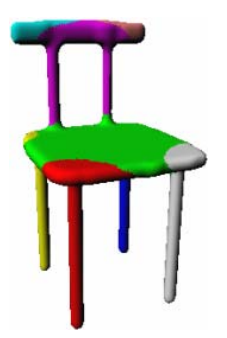
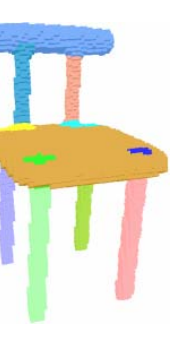
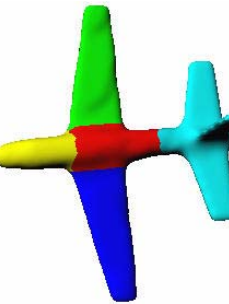
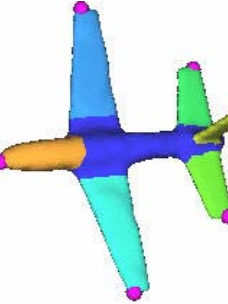
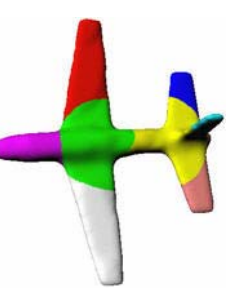


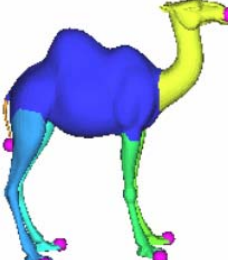
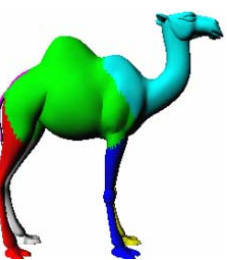

- i) **Type of segmentation.** All of the algorithms considered are part-based and are designed to segment the mesh into components that are meaningful in human perception.
- ii) **Extracting the “correct” segments.** In this criterion, the “correct” segmentation depends on the application, the viewer's perspective and knowledge of the world which can only be judged qualitatively by looking at the images of the output of the segmentation algorithms. The application that will use the proposed mesh segmentation scheme is 3D model retrieval. Judging from the overall segmentation results presented in Figure 3.2.1 it can be seen that the proposed approach and LIN perform best, i.e. both algorithms segment the meshes in a perceptually correct way and it can be observed that meshes that belong in the same category, for example all humans, are segmented consistently which is very important in the shape retrieval context. VALETTE manages to extract also perceptually meaningful results but as there are many cases where the segmentation among meshes of the same category are not consistent (eg. models (12)-(13) of Figure 3.2.1 which has a negative effect in retrieval. KIM manages also to extract perceptually meaningful components but as in VALETTE there are cases where the segmentation among meshes of the same category are non-consistent (e.g. models (7)-(8), (9)-(10), (12)-(13), (14)-(19) of Figure 3.2.1.
- iii) **Boundaries.** Here the “correctness” of the boundaries will be judged upon two geometric properties, (a) the smoothness of the boundary and (b) its location along concave features. From Figure 3.2.1 it can be seen that the proposed segmentation algorithm and LIN produce the smoothest results. This is expected because both algorithms use a minimum cut algorithm based on the concave features of the mesh in order to refine the boundary. In respect to the location of the boundaries along





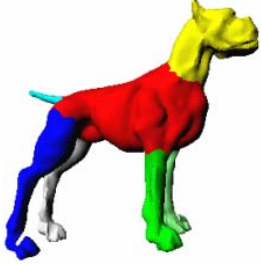
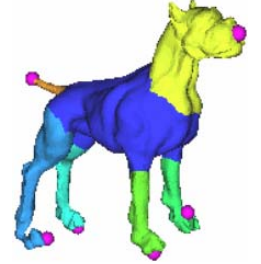
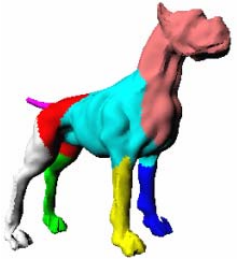


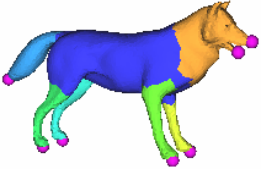
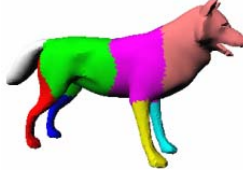

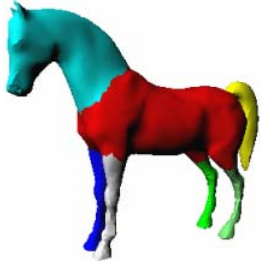
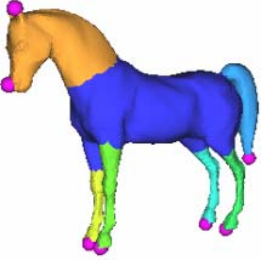
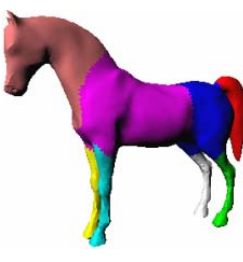
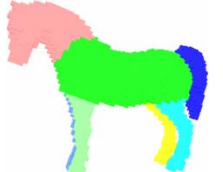

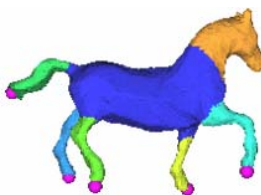


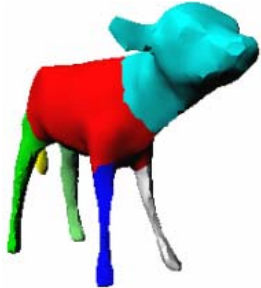

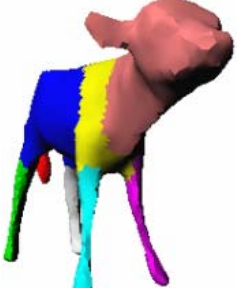

concave features it can be seen that the proposed segmentation algorithm performs best.






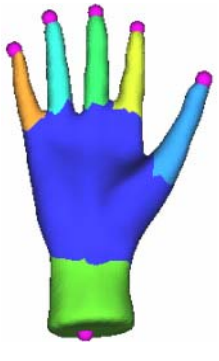



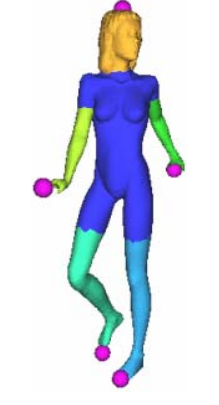




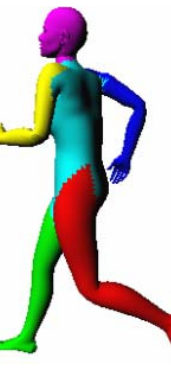

- iv) **Hierarchical / multi-scale segmentation.** Among all tested algorithms only KIM produces hierarchical results.
- v) **Sensitivity to pose.** Models (15)-(18) of Figure 3.2.1 illustrate the pose sensitivity of the four algorithms. These models represent the same human in different poses : running, jumping, sitting and walking. It can be seen that the proposed approach and LIN algorithms manage to remain invariant throughout all of the pose changes of the mesh. VALETTE and KIM though show sensitivity to the pose of the object.
- vi) **Complexity.** The proposed algorithm : Let N be the number of points and N_s the number of salient points of the mesh.





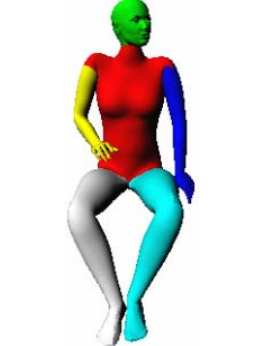
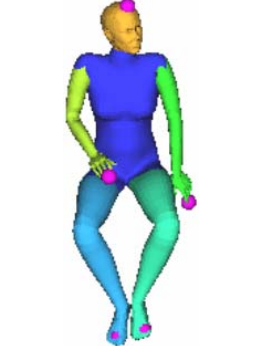

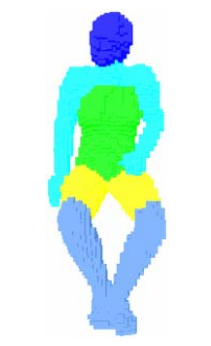

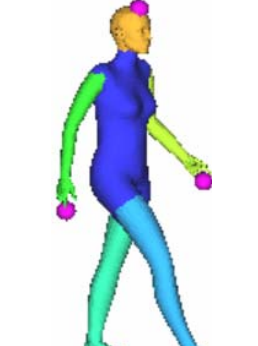



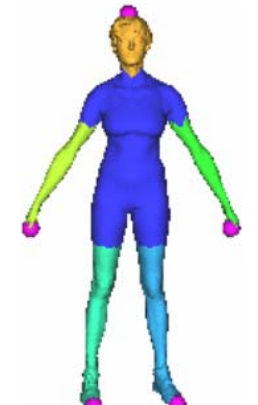
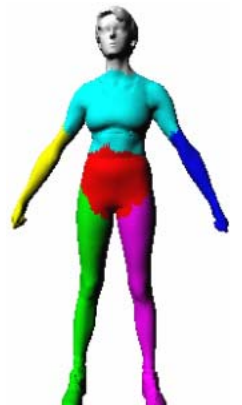

a) Protrusion function computation : In this work the geodesic distances are computed using the Dijkstra algorithm. In order for the computation to be more accurate midpoint subdivision is used in order to increase the resolution of the mesh. In most of the cases the subdivision is done twice, so the number of points of the mesh increase at the magnitude of $8N$. Lets assume that the number of compact regions are M . The complexity of the protrusion function calculation is $O(8MN \log(N))$.

b) Salient Points Extraction computation. Lets assume that the maximum number of points in a geodesic neighborhood is N_{max} then the complexity of the salient points extraction computation is $O(NN_{max} \log(N_{max}))$. Salient Points Grouping. The complexity of the salient point grouping is $O(N_s^2 N \log(N))$.

3D Model Nr	Proposed algorithm	LIN	VALETTE	KIM
(1)				
(2)				
(3)				
(4)				
(5)				

3D Model Nr	Proposed algorithm	LIN	VALETTE	KIM
(6)				
(7)				
(8)				
(9)				
(10)				
(11)				

3D Model Nr	Proposed algorithm	LIN	VALETTE	KIM
(12)				
(13)				
(14)				
(15)				

3D Model Nr	Proposed algorithm	LIN	VALETTE	KIM
(16)				
(17)				
(18)				
(19)				

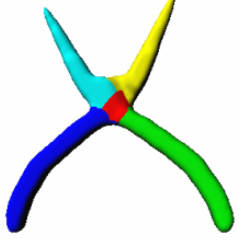
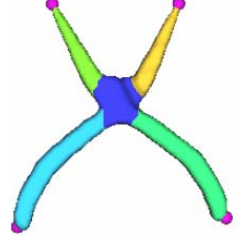
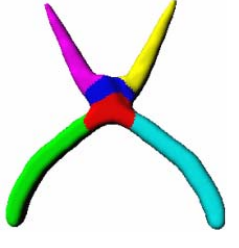
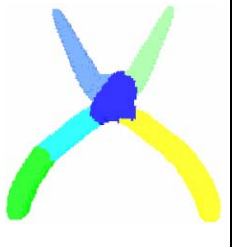
3D Model Nr	Proposed algorithm	LIN	VALETTE	KIM
(20)				

Figure 3.2.1. Segmentation results

- c) The complexity of the core approximation is $O(NN_S^2)$.
- d) The complexity of the protrusion boundary detection is $O(kI_{per}N + kN\log N + 8kmn\log n)$, where m is the number of compact regions, n is the number of points of the part of the mesh where the refined representative is calculated, k are the number of the representative salient points and I_{per} the number of the closed boundaries.
- e) The complexity of the protrusion boundary refinement is $O(N_{ng}^2 \log(N_{ng}))$ where N_{ng} is the number of nodes in the network graph.

Overall the algorithm dominant complexity is $O(8MN \log(N) + NN_{max} \log(N_{max}) + 8kmn\log n)$.

LIN : The total complexity of LIN is $O(MF\log(F) + F^2 \log(F))$, where F the total number of faces of the Mesh and M the number of compact regions used in the calculation of the protrusion function.

VALETTE : The total complexity of VALETTE is $O(N\log(N))$, where N the total number of vertices of the mesh.

vii. **Control parameters.**

The results of the proposed segmentation algorithm in Figure 3.2.1 are obtained by setting, $t_{prot} = 0.45$ (the protrusion function is scaled between $[0,1]$), $t_c = 0.15$, $d_1 = 0.1$, $d_2 = 0.4$, $I_{per} = 12$, $r_{max} = 1.3$. All of them remain fixed and user interactivity is

not required.

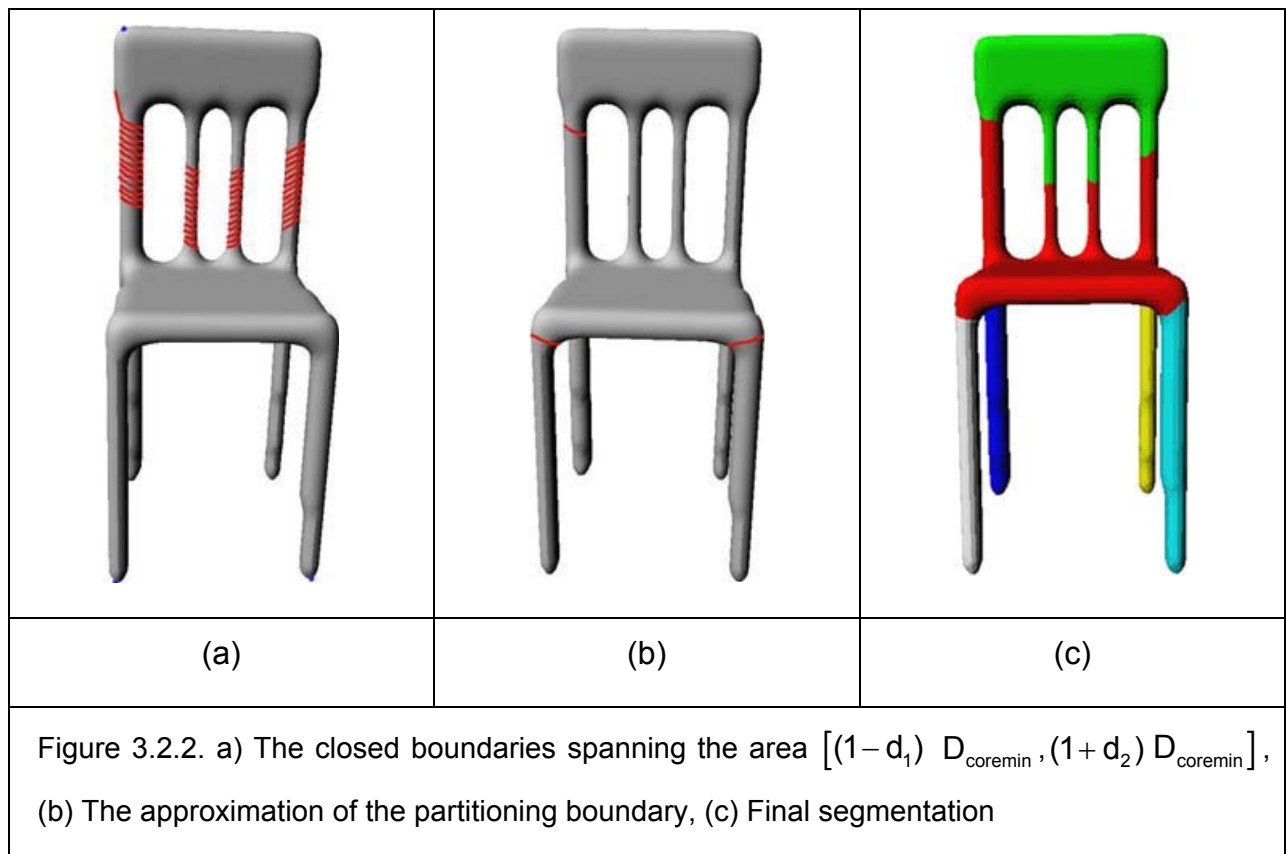
In LIN algorithm there exist three parameters that the user has to set manually and depend on the core approximation and the locales produced by their algorithm. Namely the parameters that have to be set by the user are (1) the parameter Δ_β which controls the extent that the locales overlap with the core of the object (2) the parameters Δ_+ , Δ_- which control the range of the locales that contain the protrusion's boundary.

It can be seen that the proposed algorithm performs equally or better than their results in spite that the parameters remain fixed.

In VALETTE algorithm there exists only one parameter P_{ratio} which we have set after experimentation to 15% to all of the objects. We have found that this parameter value produces the best results in their algorithm.

In KIM algorithm despite that they set the parameters fixed, the output is strongly dependent of the voxelisation resolution. This means that there is a need for human inspection to correct the resolution when the output is not satisfactory.

In general, there is not any particular geometrical or topological feature that limits the functionality of the proposed segmentation methodology for the models used in the experimentation. However, the success of the proposed segmentation is bound to the interval $[(1-d_1) D_{coremin}, (1+d_2) D_{coremin}]$. For a meaningful segmentation, the aforementioned interval should produce a region that includes the partitioning boundary. If this is not the case, there will be a segmentation outcome as in the 3D model of Figure 3.2.2.



In Table 3.2.1 the time required for segmenting the models in Figure is given in seconds. The computation time was calculated in a Pentium 4, 3GHz PC with 1.5MB memory and the segmentation algorithm was implemented in C++.

The proposed segmentation algorithm has been tested also in terms of robustness with respect to noise, i.e. it produces the same segmentation results with different levels of noise. A representative example of the proposed mesh segmentation for different levels of Gaussian noise is shown in Figure 3.2.3.

In Figure 3.2.4 the core and partitioning boundary approximation of the models in Figure 3.2.1 are shown. It can be observed that the core approximation either cover portions of the protrusible parts areas or is very close to the neighboring areas where the real boundary is situated. It can also be observed that the approximation of the partitioning boundary is also effective since it is very close to the real partitioning boundary.

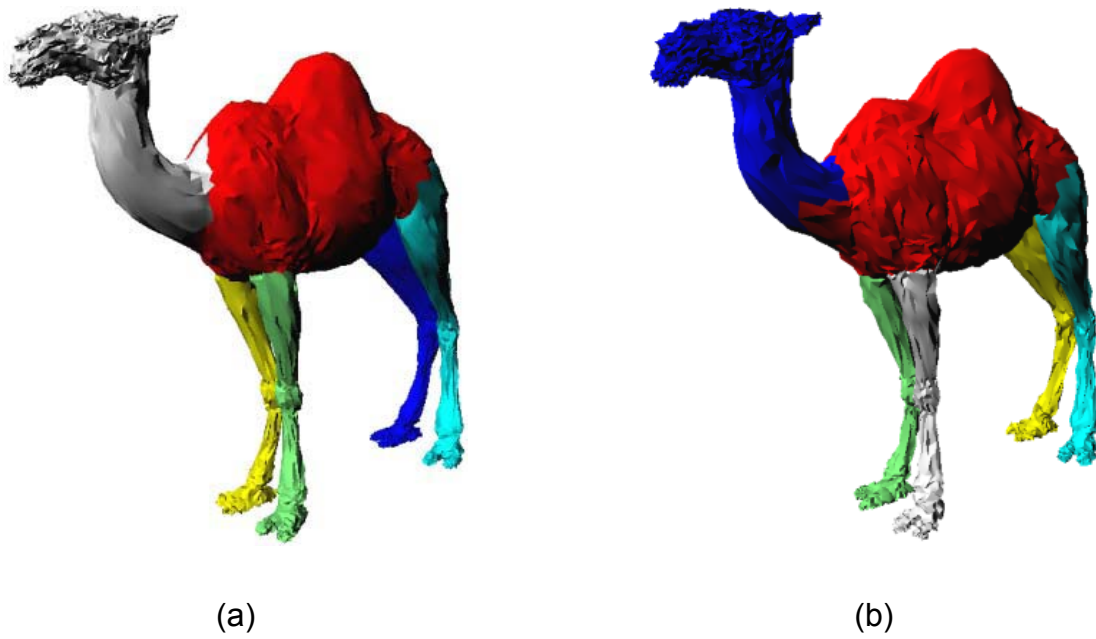


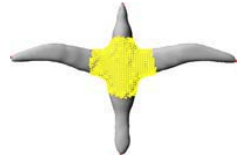
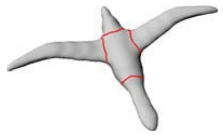




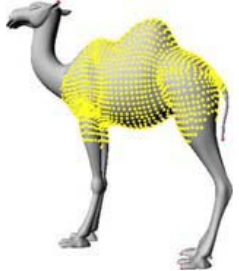


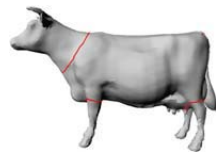

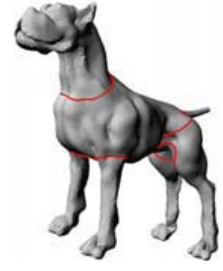

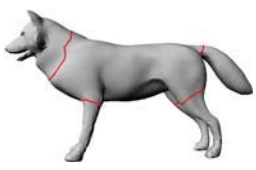

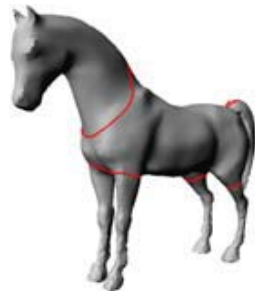
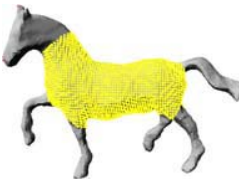
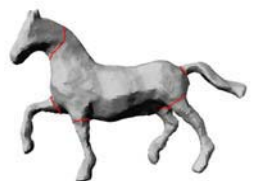
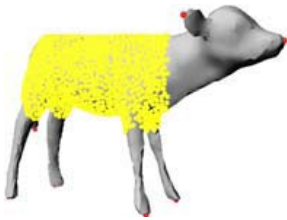
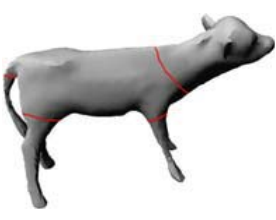



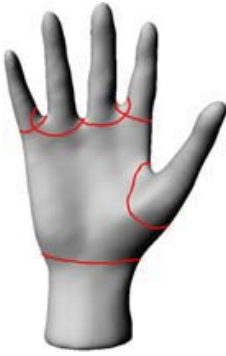












Figure 3.2.3. Segmentation of the 'camel' model under different levels of noise (a) SNR = 52dB, (b) SNR = 50dB}

Model Nr.	Nr. Vertices	Nr. Triangles	secs
(1)	8504	17004	60
(2)	3478	6952	35
(3)	12326	24652	68
(4)	6689	13374	31
(5)	9761	19518	60
(6)	7255	14506	38
(7)	9492	18980	45
(8)	4712	9420	55
(9)	7268	14532	35
(10)	11312	22620	61
(11)	3703	7402	44
(12)	7242	14480	34
(13)	6607	13210	29
(14)	11016	22028	59
(15)	5775	11546	52
(16)	5766	11528	72
(17)	5772	11540	54
(18)	5769	11534	55
(19)	9509	19014	55
(20)	6104	12204	24

Table 3.2.1. Computation time for the segmentation of Figure 3.2.1 3D models

Model Nr.	Core	Partitioning boundaries	Model Nr.	Core	Partitioning boundaries
(1)			(2)		
(3)			(4)		
(5)			(6)		
(7)			(8)		
(9)			(10)		

Model Nr.	Core	Partitioning boundaries	Model Nr.	Core	Partitioning boundaries
(11)			(12)		
(13)			(14)		
(15)			(16)		
(17)			(18)		





Model Nr.	Core	Partitioning boundaries	Model Nr.	Core	Partitioning boundaries
(19)			(20)		

Figure 3.2.4. Core approximation and partitioning boundaries of the models of Figure 3.2.1

3.3 Conclusions

In this chapter a new 3D mesh segmentation algorithm has been presented. The algorithm is based on the basic idea that an object can be segmented into its parts if its main body and salient points are reliably approximated.

The novelty of the approach is twofold:

- i. A novel way to trace the partitioning boundaries of the 3D object using closed boundaries constructed with the aid of a distance function is presented;
- ii. A novel algorithm for the core approximation of the 3D object is introduced.

The proposed algorithm is capable of segmenting a 3D object into perceptually meaningful parts and is pose invariant. From the experimental results presented it has been shown that it can successfully segment a wide range of 3D objects. Also it has been shown that the partitioning boundary detection and core approximation stage is very robust.

The evaluation of the proposed algorithm is addressed in a consistent framework wherein a comparison with the state of the art is performed. In this comparison it has been shown that in terms of extraction of the “correct” segments, boundary quality, and sensitivity to pose the proposed and Lin *et. al.* segmentation methodologies produce the best results, though the proposed algorithm is more robust since it does not rely on any

user defined parameters. Also the quality of the core approximation is superior to the one produced by Lin *et. al.*

The proposed algorithm is capable of producing compatible segmentations, for instance the human objects are always segmented into the head, body, arms and legs, thus it can be successfully applied to applications that require consistent segmentations like 3D shape retrieval based on graph based representations.

Chapter 4

State of the art in retrieval of 3D articulated objects

4.1 Introduction

In this introduction the problem of 3D object retrieval will be stated as well as its extension for 3D articulated object retrieval which is the main topic of this chapter.

4.1.1 The 3D object retrieval problem

Recent advances in 3D object acquisition in digital form have created a plethora of 3D objects ready to be processed from various graphics applications. The wide availability and continuous increase of bandwidth of the Internet is making possible to share these objects with a consequence of the construction of databases containing them. The continuous increase of the size of these databases have made necessary the construction of efficient retrieval algorithms which make possible the retrieval of 3D objects from these databases similar to a query 3D object.

The similarity between the query object and an object in the database is measured by a distance measure, let it be called d , which is defined by the specific retrieval algorithm. In order to compute the distance measure the 3D object is described by a representation. The retrieval algorithms developed distinct themselves according to the descriptors they use in order to represent the 3D object's shape/structure and the distance measure used to judge the similarity of two objects.

Formally, given a query 3D object, Q , and a database, D , consisting of a set of 3D objects, $O = \{M_i\}$, $i = 1, \dots, n$, the retrieval result from the database is defined as the following sequence:

$$\text{Retrieval}(Q,D) = \{d(Q,M_{r_1}), \dots, d(Q,M_{r_n})\} \quad (3.1.1)$$

where $r_i \in \{1, \dots, n\}$ and $d(Q, M_{r_i}) < d(Q, M_{r_j}), i < j$.

The 3D objects of the database D are grouped into classes where each class contains the objects which share a common shape or structure. It is also assumed that the query object belongs to one of the classes of D . The efficiency of the retrieval algorithm depends on whether the first elements of the sequence $\text{Retrieval}(Q, D)$ belong to the same class that the query object belongs.

4.1.2 Graph-based Retrieval of articulated objects

Human perception recognizes a 3D object by first describing it in terms of its components that are attributed with geometrical characteristics and relational connections with each other. This description is referred to as the *structural description* of the object [Bie87]. 3D object recognition is achieved by comparing the query object structural description with the structural descriptions of already classified objects and the object is classified to the class of the best match.

This recognition process can be naturally adopted for 3D object retrieval. Meaningful components according to human perception can be extracted using a segmentation algorithm. The structural description of the object is created by using the *Attributed Relational Graph (ARG)* concept. Specifically the components extracted by the segmentation algorithm are represented as the nodes of a graph and the relationship of the components with each other are represented as the edges of the graph. To each node *unary* attributes are assigned which describe the geometric characteristics of the component it represents and to each edge *binary* attributes are assigned which describe the relationship of the components that the edge connects.

Eventually, the problem of matching a query 3D object with the 3D objects stored in the Database in the retrieval process is transformed into the problem of matching their corresponding ARGs [KPYL04, TZ06].

This graph-based retrieval methodology is very useful for the retrieval of 3D articulated objects. An object is considered articulated if it consists of parts that can make the 3D object take different poses; these parts will be called in this context

articulated parts. For example an object representing a human body is considered articulated since it consists of parts (legs, arms, head) than can make the object take different poses (standing, walking, sitting, etc.). Most of the retrieval algorithms which are based on a global descriptor of the object fail to consistently compensate for the intra-class variability of articulated objects. This occurs because it is not evident how a global descriptor will become invariant to the different poses an articulated object can undertake. On the other hand a graph-based retrieval algorithm can describe an object in terms of its articulated parts thus becoming invariant to the different poses it undertakes.

4.2 State of the art in 3D object retrieval

From the variety of work that deal with 3D object retrieval two main categories can be distinguished :

- i. Methods with global shape representations. These representations describe the objects global shape in order to measure the similarity between them
- ii. Methods with graph-based shape representations. These representations use the structural description of the objects to measure their similarity.

In sections 4.2.1 and 4.2.2 representative works belonging to these two categories will be described. Emphasis has been given to those that can efficiently retrieve articulated objects.

4.2.1 Methods with global shape representations

These methods can be further classified according to the spatial dimensionality of the information used for retrieval, i.e. 2D, 3D and their combination.

Methods that use 2D information for retrieval use descriptors that are generated from image-projections that may be contours, silhouettes, depth buffers, etc.

Chen *et. al.* [CTSO03] describe the 3D object by projecting it to the 2D plane under different viewpoints taken by placing cameras at the vertices of a dodecahedron

surrounding the 3D object. This projection results in the acquirement of ten silhouettes of the object, see figure 4.2.1.

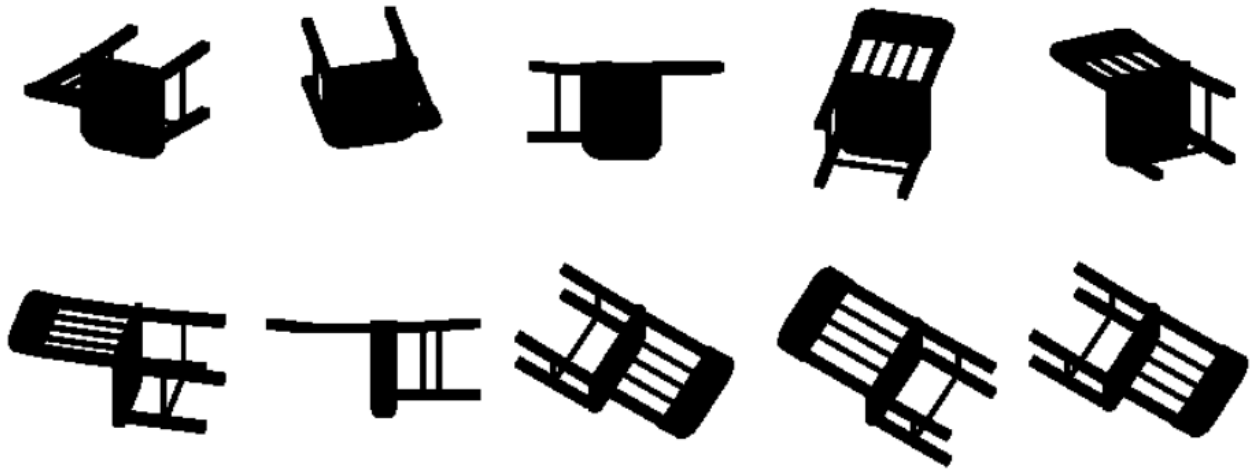


Figure 4.2.1. Ten silhouettes of a ‘chair’ model derived from its projection to the 2D plane with viewpoints taken from the vertices of a dodecahedron surrounding the 3D object.

These ten silhouettes describe the 3D object and they call this descriptor *lightfield* descriptor. The descriptor is comprised of region based (Zernike moments) coefficients and contour shape (Fourier) coefficients derived from each of the silhouettes.

In order to accomplish retrieval the lightfield descriptor of the query model is compared with the lightfield descriptors of the models stored in the database. Rotation invariance between the two models that are compared each time is achieved by rotating the cameras (60 possible rotations) on the vertices of the dodecahedron surrounding one of the 3D models. This results in the acquirement of 60 different lightfield descriptors for one of the models all of which need to be compared with the lightfield descriptor of the other model and acquire the minimum value of this comparison as the similarity measure.

Specifically let $\{I_{1k}\}_{k=1}^{10}$ the 10 silhouettes acquired from the first model and $\{I_{2ki}\}_{k=1}^{10}$ the 10 silhouettes of the second model that need to be compared, $i = \{1 \dots 60\}$. The similarity of these two models is given by the following equation:

$$D = \min_{i=1}^{60} \sum_{k=1}^{10} d(I_{1k}, I_{2ki}) \quad (4.2.1)$$

where d is the L_1 norm between the coefficients of the two silhouettes I_{1k}, I_{2ki} .

An advantage of their retrieval methodology is that they can achieve rotation invariance. The main disadvantages of their algorithm is that the time complexity for retrieval is too high and to handle efficiently articulated objects.

Ohbuchi *et. al.* [OOFB08] created the bag-of-features SIFT algorithm. In their work the 3D model is projected to the 2D plane under different viewpoints taken at the vertices of a icosahedron deriving by this way 42 range images of the object. To each of the images local features are extracted using the Scale Invariance Feature Transform (SIFT) algorithm. From all of the objects in the database and all images derived by their projection they create a visual codebook of all the calculated features using the k-means algorithm. The visual words of the codebook are the centroids of the clusters created by the k-means algorithm. The descriptor of an object is obtained by quantizing all the features calculated from all of its image projections using the visual codebook. In this way the descriptor of an object is a vector containing all the frequencies of the visual words. Retrieval is achieved by computing the Kullback-Leibler divergence between the descriptors of the object.

An advantage of their retrieval methodology is that it can handle efficiently articulated objects. One of the main disadvantages in their methodology is that in order to construct the visual codebook training needs to be applied.

Vranic [Vra04] proposed a shape descriptor that is constructed by calculating the Fourier coefficients on the depth buffers derived by projecting the object on the four sides of the cube which surrounds the 3D object. Similarity between the query's object and each of the object stored in the database is judged by comparing their corresponding descriptor Fourier coefficients.

A disadvantage of his retrieval methodology is that it can not deal with articulated objects efficiently.

Methods that use 3D information derive their descriptors from the geometry of the 3D object.

Jain and Zhang [JZ07] constructed an eigenvalue descriptor (EVD). Specifically they do spectral analysis on the 3D object. Their spectral analysis method is based on the affinity matrix A of the object the elements of which are exponentials of the geodesic distances between 20 sample vertices of the mesh. The descriptor of the object is the eigenvalues of the affinity matrix. In the retrieval process the query's object eigenvalues

are compared against the eigenvalues of each of the model stored in the database. Due to the nature of the geodesic distance being invariant to bending their descriptor is very suitable for articulated objects.

An advantage of their algorithm is that due to the pose invariance nature of the geodesic distance their retrieval methodology can handle efficiently articulated objects.

Gal *et. al.* [GSCO07] created a pose oblivious global shape descriptor. For their descriptor they use the Shape Diameter Function (SDF) of Shapira *et. al.* [SSCO05] (see section 2.3.9) and the protrusion function (PF) (see section 2.3.3). In order to make SDF scale invariant the values of the shape diameter function are first divided by the maximum diameter of all values of SDF. Then they weight each of the values according to each influence area (e.g. the areas of the triangles surrounding it divided by the whole boundary area). Using the weighted values of SDF a histogram is constructed quantizing SDF in 64 bins. The values of the PF are divided with its maximum value and a histogram is created quantizing PF in 32 bins.

The histogram of the SDF provides a pose oblivious description of the object, while the histogram of the PF provides a pose oblivious measurement of the spatial distribution of the values of the function. Thus combining both of these two histograms their properties are also combined, i.e. a 2D rectangular histogram is constructed by both of these histograms with values ranging from [0,0] to [1,1] becoming their descriptor.

In Figure 4.2.2 the 2D histograms of the 3D object 'cat' under different poses is illustrated. The values are illustrated with colours ranging from blue (low values) to (red values). It can be observed that the descriptors remain consistent despite that the object takes different poses, thus their descriptor is quite suitable for articulated objects.

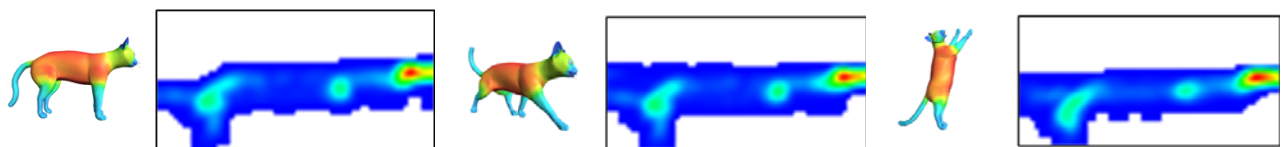


Figure 4.2.2. Shapira *et. al.* shape oblivious 2D histograms.

For retrieval purposes the matching between two objects is accomplished by the comparison of their respective descriptors using various metrics like L_1 , L_2 by unfolding the 64x32 matrix into a vector, the Minkowski L_n norms, the χ^2 measure and the earth

movers distance and correlation coefficient. Among all of these measures the χ^2 measure and correlation coefficient performed best.

An advantage of their retrieval scheme is that due to the pose invariance nature of the shape diameter and protrusion function their descriptor is also pose invariant this means that they can handle efficiently articulated objects.

Ben-chen and Gotsman [BCG08] base their descriptor on the uniformization theorem. The uniformization theorem states that any 2-manifold surface can be conformally mapped to a surface with the same topology having constant Gaussian curvature. This mapping can be achieved by defining a positive scalar function on the surface which locally changes the surface metric in order to achieve constant Gaussian curvature.

In the discrete case where a mesh represents a 2-manifold, the conformal factor ϕ is defined on the vertices and its value is derived from the following sparse set of linear equations:

$$L\phi = K^T - K^{\text{orig}} \quad (4.2.2)$$

where L is the discrete Laplace-Beltrami with cotangent weights [HPW06], K^{orig} is the Gaussian curvature on the vertices of the mesh and K^T is the target Gaussian curvature. The values of the conformal factor on two different models are represented with different colors in Figure 4.2.3. It can be observed that ϕ remains consistent in spite of the different poses that the two models undertake. This is justified because the conformal factor is based on the Gaussian curvature of the mesh which is invariant to the quasi-isometric transformations that articulated bodies undertake. Also ϕ is capable of identifying the protrusible parts of the objects.

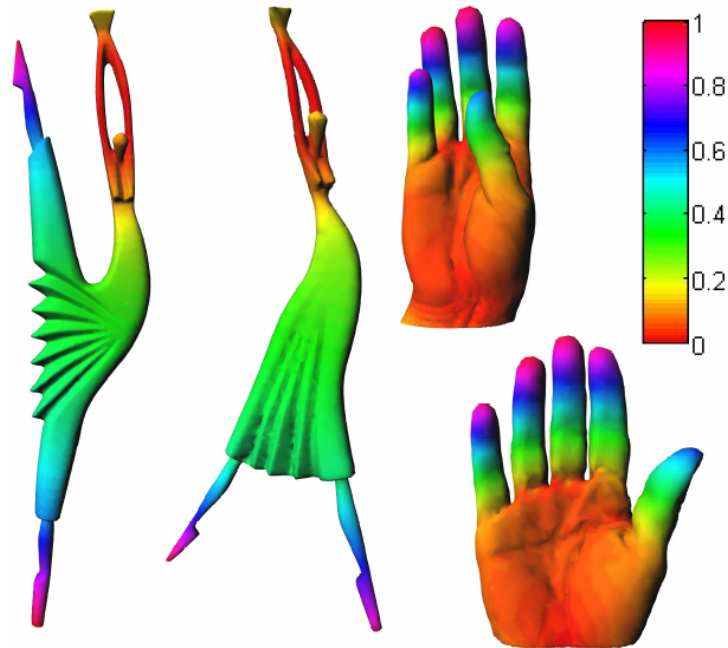


Figure 4.2.3. Visualization of the conformal factor on two 'dancer' and 'hands' models

The conformal factor is then transformed into a histogram quantizing its values into 200 bins becoming the descriptor of the object. Retrieval is achieved by comparing the query's descriptor with the descriptors of the objects stored in the database. The comparison is done using the L_1 norm.

An advantage of their retrieval scheme is that their descriptor is pose invariant, thus they can handle articulated objects.

Papadakis *et. al.* [PPPT07] created also a global representation of the object. First they align the object using principal component analysis based on the centroid of the object (CPCA) and on the normals of the object (NPCA). After alignment, the model's surface is represented by a set of spherical functions. A spherical function describes a bounded area of the model, defined by a lower and an upper radius which delimit a spherical shell. This shell is the volume in which the underlying surface of the model is represented by the equivalent spherical function points. In the sequel, they expand the spherical functions to their spherical harmonic representation. Retrieval is achieved by comparing the spherical harmonic coefficients of two objects using the L_1 norm.

The advantages of their retrieval methodology are : (i) they can achieve a highly accurate alignment of the object for retrieval purposes, (ii) their spherical harmonic

descriptor is very rich in information thus they manage to achieve high quality retrieval results. A disadvantage of their methodology is that they cannot handle efficiently articulated objects.

4.2.3 Methods using a graph-based representation

In the second category of retrieval methods a descriptor is constructed based on the structural description of the object which in most of the cases is represented by a graph structure.

Hilaga *et. al.* [HSKK01] quantized the protrusion function (see section 2.3.3) defined on the mesh using different levels of quantization creating by this way a multiresolution reeb graph structure. Figure 4.2.4 shows an example of a multiresolution reeb graph structure on a mesh at different levels of hierarchy.

Retrieval is achieved by matching the multi-resolution reeb graph structures of two objects starting from the coarsest level of hierarchy and reaching to the finer level.

An advantage of their retrieval methodology is that they can deal with articulated objects due to the pose invariant nature of the protrusion function. A disadvantage of their methodology is that the matching scheme depends only on topological attributes.

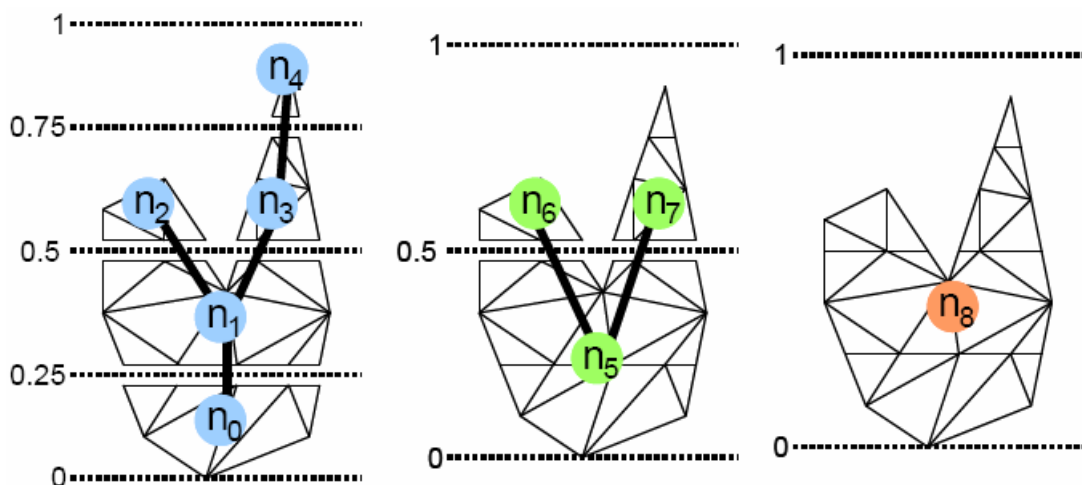


Figure 4.2.4. Hilaga *et. al.* multiresolution reeb graph structure

Tung and Schmitt [TS05] enhanced the retrieval performance of [HSKK01] by augmenting the multi-resolution reeb graph structure with geometrical and visual information.

Biasotti *et. al.* [BMSF06] constructed a descriptor based also on Reeb graph theory. In their approach they generalize the definition of the Reeb graph to a surface S by defining a finite set of contour levels $C(S)$. These contours subdivide S into a set of regions bordered by the elements of $C(S)$. The functions used for the construction of the contour levels are the distance from the center of mass and the protrusion function. To each region a node is associated. If two regions share a contour the associated nodes to them are linked with an edge. Thus a graph structure of the object is constructed which is called Extended Reeb Graph (ERG). The edges of the (ERG) are also oriented according to the monotonicity of the function f used to construct the contour levels, thus the ERG is transformed to a directed and acyclic graph. As a further step the ERG is further simplified by collapsing all nodes whose number of incoming and outgoing edges is one. After this process the ERG consists of nodes representing the regions where the topology of the contour levels varies and the associated connecting edges. In Figure 4.2.5 the process of the construction of the ERG is illustrated using as function f the distance of the center of mass.

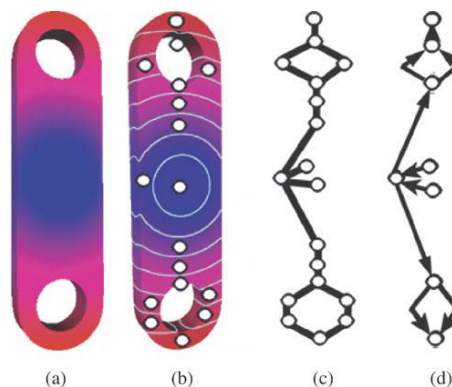


Figure 4.2.5. (a) Chromatic representation of f , (b) The nodes of the ERG, (c) The edges of the ERG, (d) The final ERG after simplification and edge orientation

The descriptor of the object is the ERG to which nodes the value of f and geometrical attributes corresponding to the region represented by the node has been given.

In the retrieval process matching between two objects is obtained by matching the directed attributed graphs (ERGs) corresponding to the two objects structural representations. This matching is based on the notion of error tolerant common sub-

graph isomorphism. Specifically the matching process between the two ERGs is achieved by constructing their most suitable common subgraph; the wider the common subgraph is, the more the two ERGs are similar. The output of the matching process should be the largest maximal common sub-graph that minimizes the geometric and the structural differences of the two objects. These differences express the similarity of the two objects. An advantage of their retrieval methodology is that it is suitable for articulated object retrieval.

Cornea *et. al.* [CDS+05] use the skeleton of the objects in order to achieve retrieval. The skeleton of the object is extracted using its volumetric representation. Specifically they use a generalized potential field generated by charges placed on the surface of the model. The generalized potential at a point of the object, due to a nearby point charge, is defined as a repulsive force, pushing the point away from the charge with a strength that is inversely proportional to the distance between the point and the charge. This potential field creates a vector field from which they extract the skeleton of the object. Retrieval is achieved by matching the skeleton of the query object with the skeletons of the objects stored in the database using the extended EMD similarity measure [RTG00].

While their method is suitable for the retrieval of articulated objects, due to geometrical or topological noise the skeletons of the objects of the same class might not be compatible reducing by this way the retrieval accuracy.

Sundar *et. al.* [SSGD03] use also the skeleton of the objects for retrieval purposes. The skeleton of the object is extracted from its volumetric representation. First a volumetric thinning is applied to the voxels of the object. Specifically a distance transform DT is defined for each of the voxels of the object which is the radius of the sphere centered at the voxel and is tangential to the boundary of the object. Based on the DT a thinness parameter TP for each voxel p is defined as :

$$TP = DT_p - \frac{\sum_{i=1}^{26} DT_{q_i}}{26} \quad (4.2.3)$$

where q_i is one of the 26 neighbours of p.

In order to thin the voxels of the object the TP is computed for every voxel and the values of TP are sorted in decreasing order. For a desired number of voxels, n, at some level of description, they extract the first n voxels from the sorted list.

After the thinning process clustering of the thinned voxels is performed in order to lessen the effects of perturbations on the surface and to reduce the number of nodes necessary for skeletal graph construction.

Afterwards the generation of the skeleton is created by applying on the thinned and clustered set of voxels the minimum span tree algorithm. The edges of the constructed graph are then directed from the voxel with the higher distance transform to the one with smaller distance transform creating by this way a directed acyclic graph (DAG) and the nodes of the graph (except from the leaf nodes) are attributed with a topological signature vector based on the eigenvalue decomposition of the sub-graphs that each of the nodes of the DAG contains.

Retrieval is achieved by matching the DAGs using a recursive, depth first formulation of bipartite graph matching. This matching process can be considered as a coarse to fine strategy, in which matching at higher levels of the tree is used to constrain matching at lower levels.

An advantage of their method is that it can handle articulated objects. A disadvantage is that due to geometrical and topological noise the skeletons of the objects of the same class might not have the same structure, thus the retrieval accuracy will be reduced.

Tal and Zuckerberger [TZ06] constructed also a graph-based representation of the object. First the object is decomposed into a small number of meaningful components using a segmentation algorithm based on the watershed theory (see section 2.3.2 and [ZTS02]). Second its attributed relational graph is constructed (ARG) based on the decomposition of the object by their segmentation algorithm, i.e. a node of the graph corresponds to a component of the object and an edge connects two nodes if their corresponding components have at least one triangle sharing a common edge. To the nodes of the ARG unary attributes are given which is the classification of the corresponding component to a basic shape : a spherical surface, a cylindrical surface, a cone surface or a planar surface. Each edge of the ARG is given a binary attribute which is the relative in size surface area of the corresponding components that the edge is incident to (i.e., greater, smaller, equal).

Retrieval is achieved by matching the query's ARG with the ARGs of the object stored in the database. This involves the comparison between two graphs. In their work they use the error-correcting sub-graph isomorphism in order to compare the two graphs.

The advantages of their methodology are: (i) they can handle articulated objects, (ii) normalization is not required, since their representation is a graph and invariant to non-rigid transformations, (iii) it can tolerate geometrical noise, like small features.

The main disadvantage of their methodology is that the complexity of the retrieval of an object is very high due to the use of error-correcting sub-graph isomorphisms whose complexity is known to be very high.

Mademlis *et. al.* [MDA+08] constructed a graph-based representation retrieval algorithm based on their segmentation algorithm, see Section 2.3.5 using Superquadrics and their 3D Distance Fields Descriptor. This descriptor is constructed by placing a surrounding ellipsoid on the object and then sampling it using θ , φ spherical coordinates. For every sample (θ_i, φ_i) the distance d_{ij} from the surface of the ellipsoid to the surface of the object is computed and then the two dimensional Fourier transform of the distance matrix is calculated forming the 3D Distance Field Descriptor. Their graph structure is based on the attributed graph concept where each object is represented by an attributed graph $G = \{V, E, \mathbf{A}, \mathbf{B}\}$ where \mathbf{A} is the edge adjacency matrix and \mathbf{B} is a vector of attributes given on the nodes of the graph consisting of superellipsoid and Distance Field Descriptor elements. Retrieval is achieved by matching the graph of two objects using van Wyk attributed graph matching algorithm [vW03].

Kim *et. al.* [KPYL04] constructed a Perceptual 3D Shape Descriptor of the object. Their descriptor is an ARG whose nodes represent the meaningful parts of the object partitioned by Kim *et. al.* [KYL05] segmentation algorithm (see section 2.3.9). An edge connects two nodes of the ARG when the parts represented by the nodes have incident voxels. To the nodes of the ARG unary attributes are given describing the parts geometrical characteristics. To the edges of the ARG binary attributes are given describing the parts relationship.

Retrieval is achieved by matching two ARGs with a nested EMD-based point matching algorithm. Specifically their matching process consists of an inner step and outer step. In the inner step the edges of each of the nodes of the ARGs creates a relational vector space. This space is an orthogonal coordinate system where the node of the ARG under consideration is at the center of the coordinate system and the nodes which are incident to this node are placed into the orthogonal coordinate system based on their binary attributes. The two coordinate systems created by the nodes of the two

ARGs are matched with each other with an EMD-based algorithm achieving to compute the difference of the binary attributes D_{binary} between them, see Figure 4.2.6.

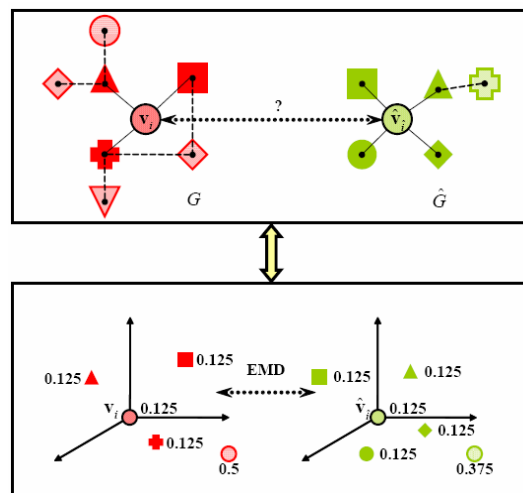


Figure 4.2.6. The inner step of Kim et. al. [KIM04] matching algorithm

In the outer step the points of the two ARGs are matched with each other using the EMD similarity measure using as ground distances the unary attribute differences and the binary differences computed from the inner step.

The advantages of their methodology are that their descriptor is invariant to non-rigid transformations due to its graph-based representation and it can tolerate geometrical noise. As it will also be shown in the experimental section (section 5.3) a disadvantage of their method is that it cannot handle efficiently articulated objects.

Chapter 5

Proposed 3D retrieval algorithm

In this chapter a graph-based retrieval algorithm for articulated objects will be presented. The segmentation algorithm which will be used in order to build the graph structure is the proposed algorithm described in Chapter 3. In this context the protrusible parts that the segmentation algorithm extracts are the articulated parts of the articulated object.

The graph matching algorithm which will be used is based on the Earth Movers Distance (EMD) similarity measure utilizing a newly introduced ground distance assignment. Specifically the graph matching process will be transformed into a signature matching process where matching will be achieved based on the ground distances defined by the unary and binary attributes of the ARG.

As will be shown from the experimental results the proposed methodology is very efficient in retrieving articulated objects showing that the combination of the proposed segmentation algorithm with the proposed retrieval methodology can achieve high quality retrieval.

The chapter organization is as follows:

- Section 5.1: Presentation of the Earth Movers Distance Similarity measure.
- Section 5.2: Presentation of the proposed retrieval methodology.
- Section 5.3: Presentation of experimental results.
- Section 5.4: Conclusions.

5.1 Earth Movers Distance Similarity Measure

In this section the Earth Movers Distance similarity measure will be detailed since it will be used by the proposed matching algorithm in order to express the similarity between two ARGs.

The EMD is based on the transportation problem which was first introduced into computer vision by Peleg *et. al.* [PWH89] to measure the distance between two gray-scale images. The EMD was later used by Rubner *et. al.* [RTG00] for image retrieval purposes in order to compare two signatures in color or texture space.

In general the EMD measure is used to efficiently express the similarity of two signatures belonging to two different distributions in a feature space [RTG00]. The two signatures \mathbf{V} , \mathbf{U} consist of two sets of nodes $\{v_i\}_{i=1}^n$, $\{u_j\}_{j=1}^m$ of size n and m , correspondingly and to each of the set of nodes weights $\{w_i^v\}_{i=1}^n$, $\{w_j^u\}_{j=1}^m$ are assigned, respectively.

Intuitively, the set of weights $\{w_i^v\}_{i=1}^n$ can be considered as piles of earth that need to be transferred to the holes that the other set of weights create in the feature space. Each unit of earth is transferred from pile i to hole j with cost $d(v_i, u_j)$ which is called *ground distance* and the total amount of earth that is transferred from pile i to hole j is denoted as $f^{(i,j)}$, that is called the *flow* of weight. The EMD measures the minimum amount of work required to transfer the piles of earth to the holes. Computing the EMD is based on a solution to the well-known transportation problem and it is solved with a linear programming optimization approach that finds the optimal flow of weight $\mathbf{F} = [f_{ij}]$ between the two distributions that minimizes the following cost term:

$$\text{WORK}(\mathbf{V}, \mathbf{U}, \mathbf{F}) = \sum_{i=1}^n \sum_{j=1}^m f^{(i,j)} d(v_i, u_j) \quad (5.1.1)$$

subject to the following constraints:

$$\begin{aligned} f^{(i,j)} &\geq 0 && 1 \leq i \leq n, 1 \leq j \leq m \\ \sum_{j=1}^m f^{(i,j)} &\leq w_i^v && 1 \leq i \leq n \\ \sum_{i=1}^n f^{(i,j)} &\leq w_j^u && 1 \leq j \leq m \\ \sum_{i=1}^n \sum_{j=1}^m f^{(i,j)} &= \min \left(\sum_{i=1}^n w_i^v, \sum_{j=1}^m w_j^u \right) \end{aligned} \quad (5.1.2)$$

The optimal cost of the optimization process is the EMD and expresses the degree of similarity between the two signatures. It is defined as follows :

$$EMD = \frac{\sum_{i=1}^n \sum_{j=1}^m f^{(i,j)} d(v_i, u_j)}{\sum_{i=1}^n \sum_{j=1}^m f^{(i,j)}} \quad (5.1.3)$$

5.2 Proposed retrieval methodology for articulated objects

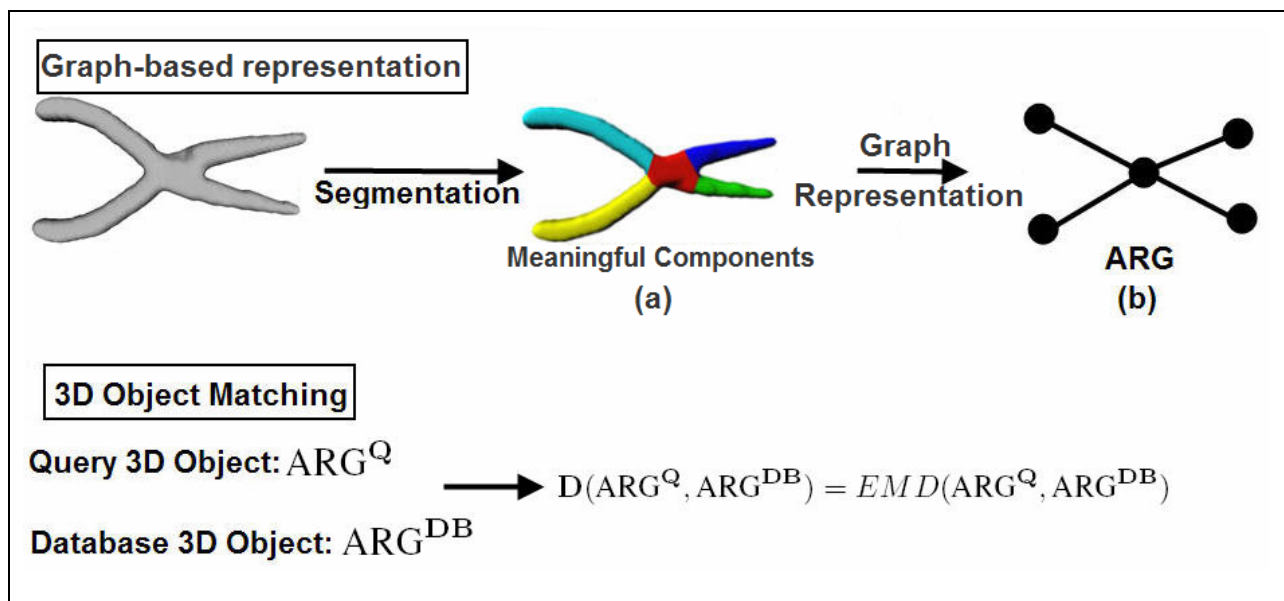


Figure 5.2.1. (a) The 3D object segmented into its meaningful components, (b) The ARG of the object

The proposed retrieval methodology uses a graph-based representation of the object. Specifically the object is first segmented into its meaningful components using the proposed segmentation algorithm described in chapter 3. Then from the components of the object a graph representation, ARG, will be constructed. Retrieval is achieved by matching the ARG of the query object with the ARGs of the objects stored in the database using the EMD similarity measure. See also Figure 5.2.1 for an illustration of the retrieval process.

In the next sections the steps of the retrieval process will be fully described as follows:

- Section 5.2.1. Construction of the ARG of the query object.
- Section 5.2.2. Matching methodology between two ARGs

5.2.1 Construction of the ARG of the 3D object

The proposed segmentation algorithm of chapter 3 is capable to segment an object into its core (main body) and its protrusible parts. Taking advantage of this segmentation a simple ARG can be constructed, its nodes are the segmented components and each of the nodes representing an articulated part is connected with the node representing the core of the object forming by this way the edges of the ARG. Unary and binary attributes will be assigned to the nodes and edges of the ARG respectively.

In Figure 5.2.2 a segmented 'Human' object by the proposed segmentation algorithm and its corresponding graph structure is shown.

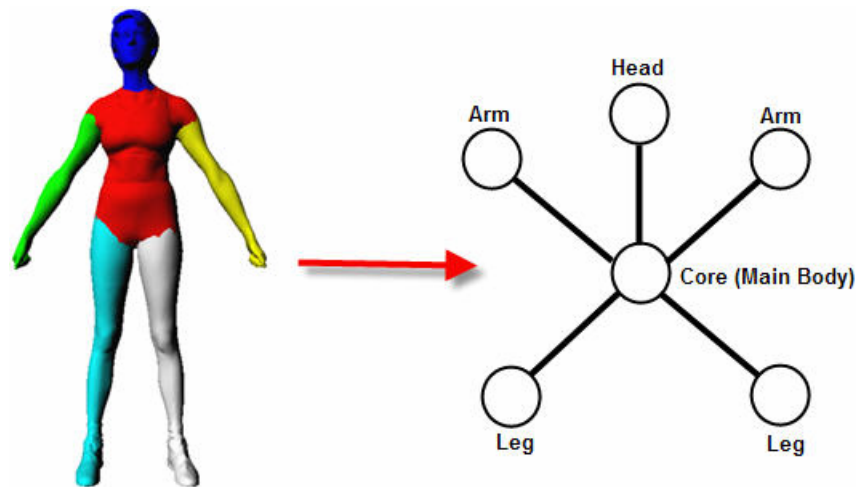


Figure 5.2.2. Segmented 'Human' object and its corresponding graph structure

5.2.2 A methodology for Matching two ARGs

As mentioned in section 5.2 retrieval is achieved by matching the ARG of the query object with the ARGs of the objects stored in the database. This process involves matching between two graph structures. In this work the problem of matching two graph structures is transformed into a signature matching process using the EMD similarity

measure to express the similarity of the two graph structures. Specifically the signatures consist of the nodes of the ARGs to which appropriate weights will be given. The ground distances which express the similarity between two nodes of the two signatures will be defined by the unary and binary attributes assigned to the nodes and edges of the two ARGs. In this way we distinct our matching methodology from Kim *et. al.* [KIM04] (see section 4.2.3) since we treat both unary and binary attributes in a single EMD calculation step without using any relational vector space.

Formally, Let $G = (V, E, \mathbf{U}, \mathbf{B})$, $\hat{G} = (\hat{V}, \hat{E}, \hat{\mathbf{U}}, \hat{\mathbf{B}})$ be the attributed relational graphs that need to be matched, where $V = \{v_i\}_{i=1}^n$, $\hat{V} = \{\hat{v}_j\}_{j=1}^m$ are the nodes (v_1, \hat{v}_1 represent the core component of the two objects, respectively), $E = \{r_{ij}\}_{i=2}^n$, $\hat{E} = \{\hat{r}_{ij}\}_{j=2}^m$ are the edges, $\mathbf{U} = \{u_i\}_{i=1}^n$, $\hat{\mathbf{U}} = \{\hat{u}_j\}_{j=1}^m$ are the unary attributes of the nodes and $\mathbf{B} = \{b_i\}_{i=2}^n$, $\hat{\mathbf{B}} = \{\hat{b}_j\}_{j=2}^m$ are the binary attributes of the edges of the two graphs, respectively. It is also assumed that $n \geq m$. Weights $\{w_i^v\}_{i=1}^n$, $\{w_j^u\}_{j=1}^m$ are assigned to the nodes of the graphs and each of them is equal to $\frac{1}{n}$ so as to attain a uniform distribution. As already mentioned, it is assumed that the nodes v_1, \hat{v}_1 represent the core component of the two objects, respectively. These nodes are considered as *fixed* and are always matched in the matching algorithm. Also additional $n - m$ nodes are inserted in \hat{G} which are called *delete* nodes. The reason for doing this is to penalize the $n - m$ nodes of G that are not mapped to any of the nodes of \hat{G} from within the EMD-based matching process. By this way partial graph matching is taken into account. Unary attributes $\hat{\mathbf{U}}_d = \{\hat{u}_{d_j}\}_{j=m+1}^n$ are assigned to the delete nodes that correspond to components with no information. Weights also equal to $\frac{1}{n}$ are assigned to the delete nodes. All other nodes representing the articulated parts of the objects are considered as *normal*.

As mentioned earlier the matching process between two ARGs can be transformed into a signature matching problem using the EMD similarity measure. For example given two ARGs, the one consisting of 5 nodes and the other consisting of 3 nodes, the matching process is as shown in Figure 5.2.3.

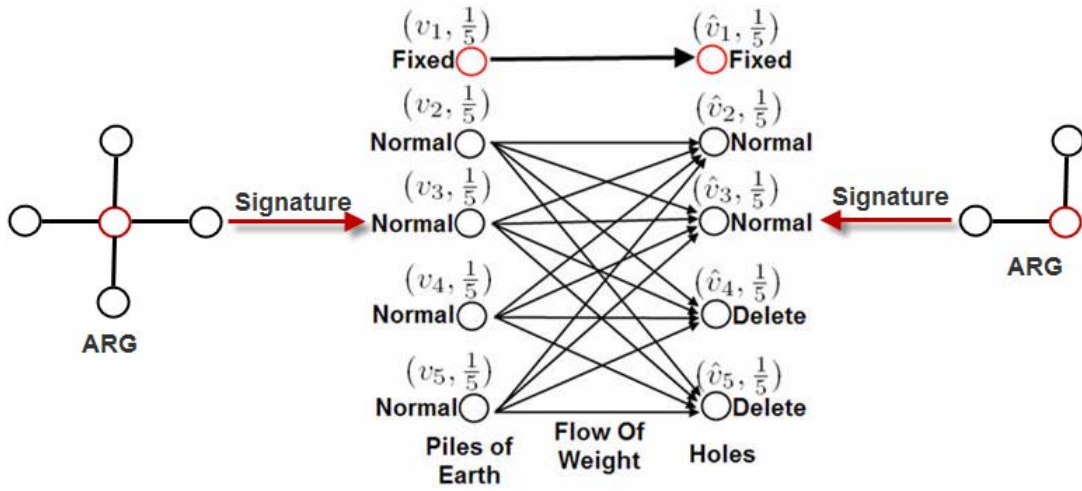


Figure 5.2.3. Matching process between two ARGs

From the definition of the EMD (5.1.3) it can be observed that the ground distances are the definitive terms while computing the EMD, thus the whole matching process is based on their proper definition because they indicate how the nodes are matched. In the case of matching two ARGs the ground distances depend upon the unary and binary attributes of the ARGs since these attributes should define how the matching between the nodes of the graphs should be done. Also there is a need to inflict a constraint in the definition of the ground distances in order for the fixed nodes to always be matched, restraining them to match with any other node, see Figure 5.2.3.

The ground distances are defined as follows:

$$d(v_i, \hat{v}_j) = \begin{cases} 3 \frac{D_{normal}(v_i, \hat{v}_j)}{1 + D_{normal}(v_i, \hat{v}_j)} & \text{if } v_i, \hat{v}_j \text{ normal} \\ 3 \frac{D_{fixed}(v_i, \hat{v}_j)}{1 + D_{fixed}(v_i, \hat{v}_j)} & \text{if } v_i, \hat{v}_j \text{ fixed} \\ 5 \frac{D_{delete}(v_i, \hat{v}_j)}{0.1 + D_{delete}(v_i, \hat{v}_j)} & \text{if } v_i \text{ normal,} \\ & \hat{v}_j \text{ delete} \\ \infty & \text{otherwise} \end{cases} \quad (5.2.1)$$

where,

$$\begin{aligned}
D_{normal}(v_i, \hat{v}_j) &= \sqrt{\|\mathbf{u}_i - \hat{\mathbf{u}}_j\|^2 + \|\mathbf{b}_i - \hat{\mathbf{b}}_j\|^2} \\
D_{fixed}(v_i, \hat{v}_j) &= \sqrt{\|\mathbf{u}_i - \hat{\mathbf{u}}_j\|^2} \\
D_{delete}(v_i, \hat{v}_j) &= \sqrt{\|\mathbf{u}_i - \hat{\mathbf{u}}_{d_j}\|^2}
\end{aligned} \tag{5.2.2}$$

As can be observed in equation (5.2.1) the case of the mismatch of a normal node with a delete node is penalized more than in the cases when the fixed nodes and the normal nodes are mismatched. This is done in order to avoid the matching of normal nodes that hold significant information with the delete nodes that hold no information. In Figure 5.2.4 the plots of the ground distances illustrate their different value assignment.

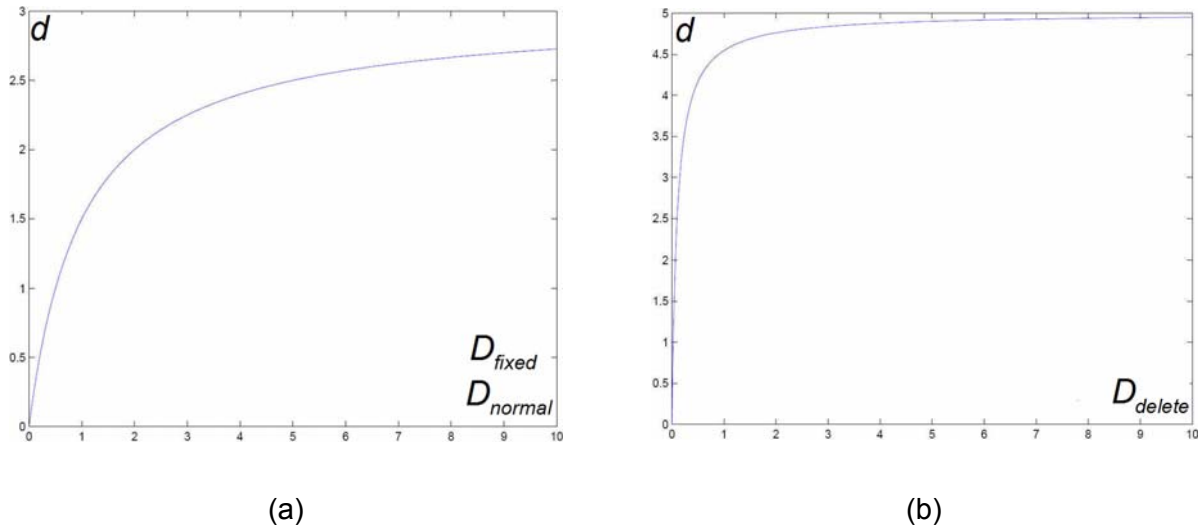


Figure 5.2.4. The ground distances plots of (a) the normal and fixed node matching, (b) the delete node matching

It can also be observed in equation (5.2.2) that the binary attributes are considered only in the normal node matching since only then the relation they have with the fixed node (core) need to be exploited. When the fixed nodes are matched only the unary attributes are considered since the core relation with the other nodes is already considered when the normal nodes are matched. Note also that with the selected ground distances the fixed nodes are always going to be matched.

The unary attributes that need to be defined for the nodes of the ARG should carry the geometric properties of the component they represent. The binary attributes should carry the relationship that the neighbouring components have.

In this work the following unary and binary attributes are used:

- i. The unary and binary attributes of Kim et. al. [KPYL04]. The purpose of this assignment is to compare the proposed matching methodology with that used in [KPYL04] in order to show the efficiency of the segmentation and matching algorithms in the experimental results (see section 5.3).
- ii. Unary attributes defined by Papadakis et. al. [PPPT07] descriptor. Their descriptor consist of spherical harmonic coefficients derived from the object's component after pose normalization. The spherical harmonics provide a description of the component's geometry in the frequency domain.

Considering Kim *et. al.* [KPYL04] attribute assignment, the unary attributes that are assigned to the nodes of the ARG representing the object components are the relative size (rs) of the component, the convexity (c) of the component and the eccentricities (e_1, e_2) of the ellipsoid approximating the component. The relative size of the component is approximated by its area, the convexity is approximated by first voxelizing the component and then dividing the number of voxels of the component by the number of voxels of its convex hull and the eccentricities are approximated by the variances of the component mesh points along the axes created by principal component analysis. The binary attributes that are assigned to the edges of the ARG are the distance (l) of the centroids of the components connected by an edge of the graph and the angles (a_1, a_2) that the two most significant principal axes of the connected components create with each other. All of the attributes are normalized in the interval [0, 1].

By this way, the vector $[rs, c, e_1, e_2]$ is assigned to the normal and fixed nodes and the vector $[l, a_1, a_2]$ is assigned to the edges of the graphs. All delete nodes are assigned the vector $[0, 1, 1, 1]$. In equation (5.2.2), the norm $\|\cdot\|$ denotes the \mathbf{L}_2 norm of the attribute vectors.

Considering Papadakis *et. al.* [PPPT07] attribute assignment, we set to the normal and fixed nodes the spherical harmonic descriptor vector of the parts they represent. The descriptor consists of two sets of coefficients corresponding to two aligned versions of the model using two methodologies based on principal component analysis, namely CPCA and NPCA. CPCA aligns the component according to the surface area distribution and NPCA aligns the component according to the surface orientation distribution. To the delete nodes the vector with zero entries is assigned whose dimension is the same as their descriptor. Please note that in this case only

unary attributes are assigned, thus in equation (5.2.2) there exist no binary term and the norm $\|\cdot\|$ denotes the L_1 norm of the spherical harmonic vector coefficients.

Based on the aforementioned ground distance assignment and ARG definition the EMD measure is computed between the two ARGs which gives the degree of similarity between the two objects that need to be matched. In order to compute the EMD the implementation of Rubner *et. al.* [RTG00] is used.

5.3 Experimental Results

For the experiments made in this work, the McGill 3D object database of articulated objects is used [mcg]. The specific database contains ten classes of 255 articulated objects, namely, ‘Ants’, ‘Crabs’, ‘Spectacles’, ‘Hands’, ‘Humans’, ‘Octopuses’, ‘Pliers’, ‘Snakes’, ‘Spiders’ and ‘Teddy-bears’ each one of them containing approximately twenty to thirty models, see Figure 3.3.1. The objects in this database were transformed from their voxelized form to closed manifolds. For each of the objects this was achieved by first triangulating the sides of the voxels which belong to the exterior of the object disregarding all other sides, thus creating by this way a 3D mesh. Afterwards Laplacian smoothing was applied to the mesh, in the sequel the triangles and vertices which prevented the mesh from being a manifold were removed. Lastly the holes which were created by the latter removal were closed using a hole filling algorithm, thus creating by this way a closed manifold.











				
‘Ants’	‘Crabs’	‘Spectacles’	‘Hands’	‘Humans’
				
‘Octopuses’	‘Pliers’	‘Snakes’	‘Spiders’	‘Teddy-bears’

Figure 3.3.1. The ten classes of the McGill Database of articulated objects

The experiments made in this work have the following goals:

- First, the superior performance of our retrieval methodology will be shown in comparison with two other state of the art retrieval methodologies, namely Kim *et. al.* [KPYL04] (which has become one of the standards 3D object retrieval methodologies in MPEG7) and Papadakis *et. al.* [PPT+08].
- Second, the proposed retrieval methodology will be used in order to refine Papadakis *et. al.* [PPT+08] retrieval results. This will be achieved by applying for each object of the database the retrieval process of Papadakis *et. al.* [PPT+08]. The output of the retrieval is a sequence of 255 distance measures in ascending order indexed with the object that each measure corresponds to. The first 75 objects of this output will be further matched with the proposed matching algorithm against the query object providing a new distance measure sequence which will be shown that it refines the sequence of the initial 150 distance measures (the rest of the distance measures are left unaltered).
- Lastly, the superiority of the proposed segmentation algorithm in terms of retrieval will be shown against the segmentation algorithm used for retrieval in Kim *et. al.* [KPYL04]. This is going to be achieved by accommodating the ARG created by the proposed segmentation algorithm using Kim *et. al.* [KPYL04] attributes to the MPEG7 retrieval process.

In the sequel, the following abbreviations will be used:

- The graph-based retrieval methodology that uses the proposed segmentation and matching algorithm using Papadakis *et. al.* [PPPT07] attributes is denoted as **EMD-PPPT**
- The graph-based retrieval methodology that uses the proposed segmentation and matching algorithm using Kim *et. al.* [KPYL04] attributes is denoted as **EMD-MPEG7**
- The graph-based retrieval methodology that uses the proposed segmentation algorithm and the graph matching of Kim *et. al.* [KPYL04] is denoted as **SMPEG7**
- The graph-based retrieval methodology that uses the segmentation and matching of Kim *et. al.* [KPYL04] is denoted as **MPEG7**
- The retrieval methodology of Papadakis *et. al.* [PPT+08] that uses a global shape representation is denoted as **Hybrid**

- The retrieval methodology of Papadakis et. al. [PPT+08] refined by the proposed retrieval methodology using Kim et. al. [KPYL04] attributes is denoted as **H-EMD-KIM-R**
- The retrieval methodology of Papadakis et. al. [PPT+08] refined by the proposed retrieval methodology using Papadakis et. al. [PPPT07] attributes is denoted as **H-EMD-PPPT-R**

The evaluation of the retrieval results of the aforementioned retrieval methodologies is based upon the Precision Recall (PR) diagrams, where higher Precision values indicate better performance and on the following quantification measures:

- Nearest Neighbour (NN): The percentage of queries where the closest match belongs to the query's class.
- First Tier (FT): The recall for the $(k-1)$ closest matches, where k is the cardinality of the query's class.
- Second Tier (ST): The recall for the $2(k-1)$ closest matches, where k is the cardinality of the query's class.
- Discounted Cumulative Gain (DCG): A statistic that weights correct results near the front of the list more than correct results later in the ranked list under the assumption that a user is less likely to consider elements near the end of the list.

The above measures range from 0% to 100% and higher values indicate better performance.

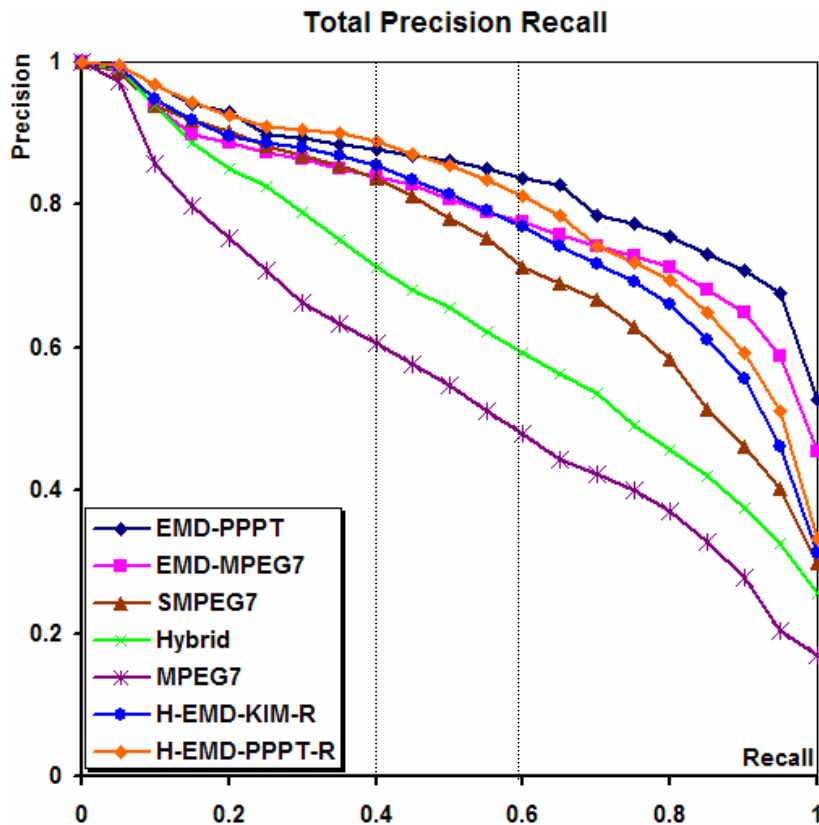


Figure 5.3.2. PR plot of the examined retrieval methodologies

In Figure 5.3.2 the Precision Recall diagram for the whole database is shown. It is obvious that the EMD-PPPT methodology has the best performance. Specifically if 40% and 60% recall levels are considered Table 5.3.1 is obtained.

RETRIEVAL ALGORITHMS	RECALL LEVELS	
	40%	60%
EMD-PPPT	0.88	0.83
EMD-MPEG7	0.84	0.77
SMPEG7	0.84	0.76
Hybrid	0.71	0.59
H-EMD-KIM-R	0.86	0.76
H-EMD-PPPT-R	0.89	0.80
MPEG7	0.61	0.48

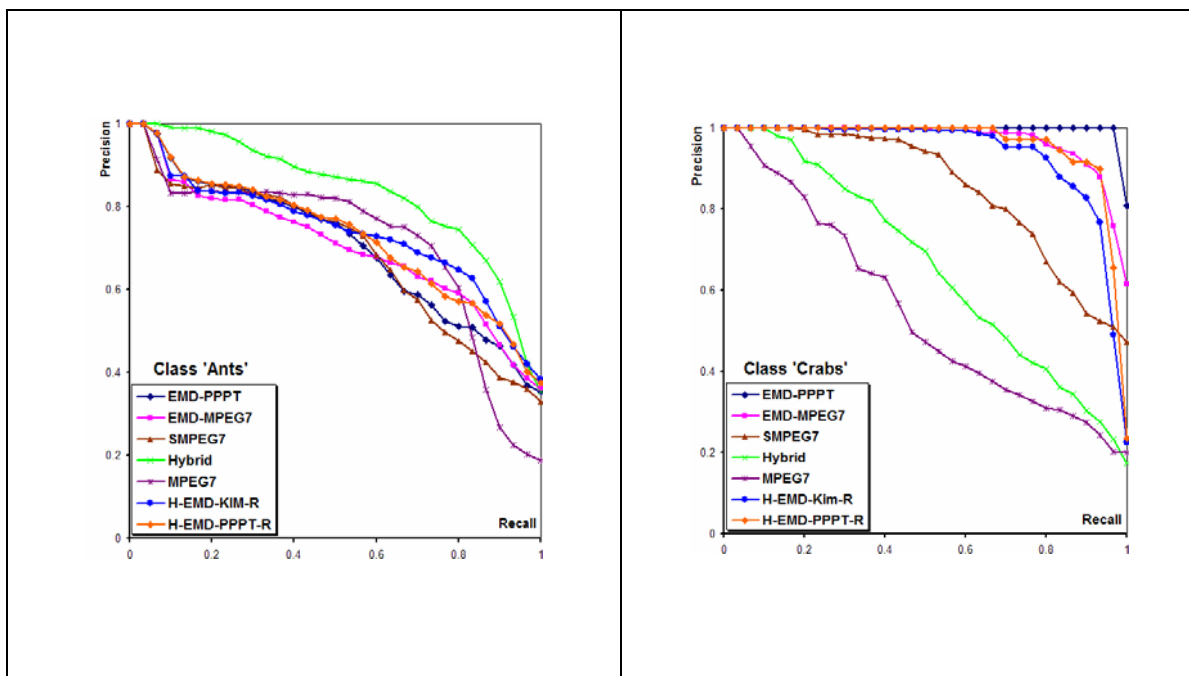
Table 5.3.1. Precision performance values for the recall levels 40% and 60%

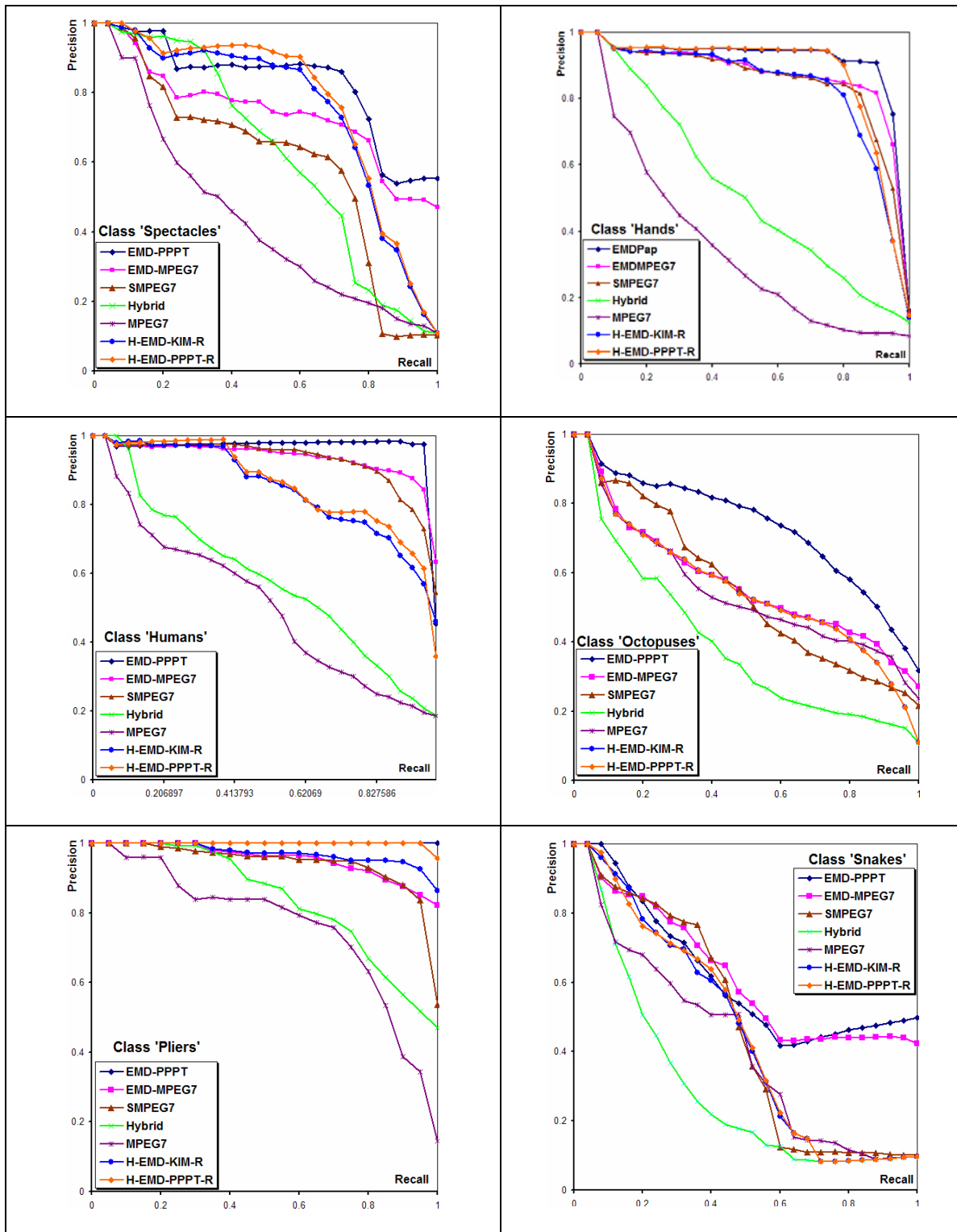
Based on the precision performance values of Table 5.3.1 the following can be observed:

- i. The best precision results are those of EMD-PPPT
- ii. EMD-MPEG7 is the second in precision performance value

- iii. H-EMD-KIM-R and H-EMD-PPPT-R compared to Hybrid has been improved by 20% and 24% approximately respectively in the precision performance value
- iv. SMPEG7 retrieval curve is better than the MPEG7 retrieval curve by an average increase in the precision recall level of the order of 33%

Based on the above statements it can be safely implied that the spherical-harmonics attributes set on the components of the object can describe the structure of the articulated objects very efficiently providing high quality retrieval results. Also the EMD-PPPT retrieval curve is about 5% higher than the EMD-MPEG7 which means that the spherical-harmonics attributes can describe better a component than the attributes set by Kim *et. al.* [KPYL04]. It can also be observed that after refining the retrieval result of the Hybrid algorithm the performance of H-EMD-KIM-R, and H-EMD-PPPT-R is significantly better than the Hybrid performance and comparable to the SMPEG7 retrieval curve. This means that the proposed retrieval methodology is capable to assist the Hybrid retrieval methodology to achieve high quality retrieval results. The better performance of SMPEG7 over MPEG7 means that the proposed segmentation algorithm is performing better than Kim *et. al.* [KPYL04] segmentation algorithm for retrieval purposes in this database.





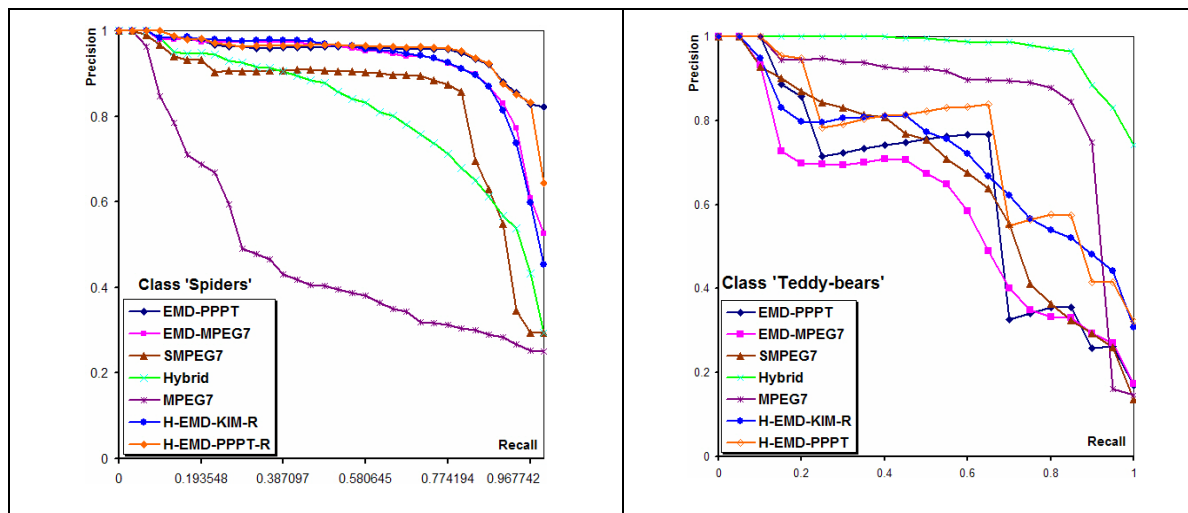


Figure 5.3.3. Precision Recall diagrams of the McGill database classes

In Figure 5.3.3 the Precision Recall diagrams for each one of the classes of the database is shown. Again it can be observed that EMD-PPPT retrieval methodology performs best in the majority of the classes. Especially In the ‘Humans’, ‘Crabs’, ‘Hands’, ‘Spiders’, ‘Pliers’ classes very high PR curves are achieved (especially in the ‘Pliers’ class where the perfect performance is attained). The inferior performance in the ‘snakes’ class is justified by the poor segmentation achieved in this particular class because of the lack of protrusions.

In Table 5.3.2 the corresponding scores for each of the retrieval methodologies for each class of the database as well as the average scores for the complete McGill database are shown. As can be observed the EMD-PPPT methodology performs better in total and in most of the classes of the database. Also it can be noticed the significant gain in score of the H-EMD-KIM-R and H-EMD-PPPT-R retrieval methodology over the Hybrid retrieval methodology.

In Figure 5.3.4 retrieval results are shown for a ‘hands’ and ‘humans’, objects respectively using the EMD-PPPT and MPEG7 retrieval methodologies. It can be observed that the proposed EMD-PPPT retrieval results are significant better than MPEG7.

Class	Method	NN(%)	FT(%)	ST(%)	DCG(%)
Complete McGill DB	EMD-PPPT	97.6	74.1	91.1	93.3
	EMD-MPEG7	93.3	69.2	88.9	90.8
	SMPEG7	91.8	65.2	78.3	89.1
	Hybrid	92.5	55.7	69.8	85.0
	H-EMD-KIM-R	94.1	70.7	82.9	90.2
	H-EMD-PPPT-R	97.3	69.9	75.8	90.5
	MPEG7	97.3	47.5	63.2	79.2

Ants	EMD-PPPT	96.7	54.9	79.7	88.4
	EMD-MPEG7	96.7	58.5	79.9	87.5
	SMPEG7	80.0	57.1	75.6	86.7
	Hybrid	100	73.6	89.2	94.8
	H-EMD-KIM-R	96.7	63.4	83.2	88.9
	H-EMD-PPPT-R	96.7	58.3	81.5	89.2
	MPEG7	90.0	62.1	75.5	87.1
Crabs	EMD-PPPT	100	98.2	99.8	99.9
	EMD-MPEG7	100	89.8	98.2	99.2
	SMPEG7	100	72.9	90.3	95.9
	Hybrid	100	55.2	71.8	88.7
	H-EMD-KIM-R	100	87.5	92.9	98.0
	H-EMD-PPPT-R	100	92.6	94.3	98.6
	MPEG7	90.0	45.9	65.5	82.2
Spectacles	EMD-PPPT	100	70.3	99.8	94.0
	EMD-MPEG7	96.0	63.7	94.3	89.2
	SMPEG7	96.0	55.8	63.7	82.7
	Hybrid	96.0	53.5	63.3	85.9
	H-EMD-KIM-R	96.0	74.0	80.0	90.5
	H-EMD-PPPT-R	96.0	73.8	80.0	91.5
	MPEG7	84.0	37.8	50.8	73.6
Hands	EMD-PPPT	95.0	83.9	88.9	95.2
	EMD-MPEG7	95.0	79.7	88.2	93.4
	SMPEG7	95.0	78.7	87.9	93.0
	Hybrid	90.0	43.4	57.6	77.8
	H-EMD-KIM-R	95.0	77.4	83.7	92.3
	H-EMD-PPPT-R	95.0	79.7	83.9	94.0
	MPEG7	60.0	30.0	41.3	63.1
Humans	EMD-PPPT	96.6	93.5	96.4	98.1
	EMD-MPEG7	96.6	86.8	99.3	97.4
	SMPEG7	96.6	84.5	98.0	97.3
	Hybrid	100	47.0	63.8	83.1
	H-EMD-KIM-R	96.6	79.6	85.2	94.3
	H-EMD-PPPT-R	96.6	82.0	84.7	94.6
	MPEG7	79.3	40.5	59.1	77.9
Octopuses	EMD-PPPT	88.0	58.8	81.8	88.1
	EMD-MPEG7	80.0	45.2	73.2	79.1
	SMPEG7	84.0	42.0	63.0	80.5
	Hybrid	56.0	29.5	45.0	68.9
	H-EMD-KIM-R	76.0	45.7	71.2	78.1
	H-EMD-PPPT-R	88.0	57.8	80.3	87.0
	MPEG7	72.0	46.8	76.2	77.8
Pliers	EMD-PPPT	100	100	100	100
	EMD-MPEG7	100	85.5	100	98.6
	SMPEG7	100	86.1	95.5	97.8
	Hybrid	100	71.6	87.9	94.6
	H-EMD-KIM-R	100	92.4	99.7	99.0
	H-EMD-PPPT-R	100	99.7	99.7	99.9
	MPEG7	95.0	65.5	77.9	89.5
Snakes	EMD-PPPT	100	43.2	95.2	84.7
	EMD-MPEG7	80.0	46.2	85.8	83.4
	SMPEG7	80.0	44.2	48.0	76.6
	Hybrid	80.0	23.7	28.7	62.4
	H-EMD-KIM-R	88.0	42.3	47.3	75.7
	H-EMD-PPPT-R	96.0	43.7	47.3	75.4
	MPEG7	76.0	36.8	40.7	69.3
Spiders	EMD-PPPT	100	87.2	100	98.4
	EMD-MPEG7	100	85.7	97.3	97.5
	SMPEG7	96.8	74.8	86.6	93.9
	Hybrid	100	71.5	91.0	93.7
	H-EMD-KIM-R	100	85.7	96.9	97.6
	H-EMD-PPPT-R	100	87.3	99.0	98.3
	MPEG7	90.3	37.3	61.8	77.8
Teddy-bears	EMD-PPPT	100	45.3	63.2	83.9
	EMD-MPEG7	85.0	42.6	66.3	78.8
	SMPEG7	90.0	55.8	70.8	84.6
	Hybrid	100	90.3	98.4	99.1
	H-EMD-KIM-R	90.0	54.7	87.4	85.5
	H-EMD-PPPT-R	100	52.6	87.4	89.1
	MPEG7	100	79.2	84.5	93.4

Table 5.3.2. Quantitative measure scores of the examined retrieval methodologies

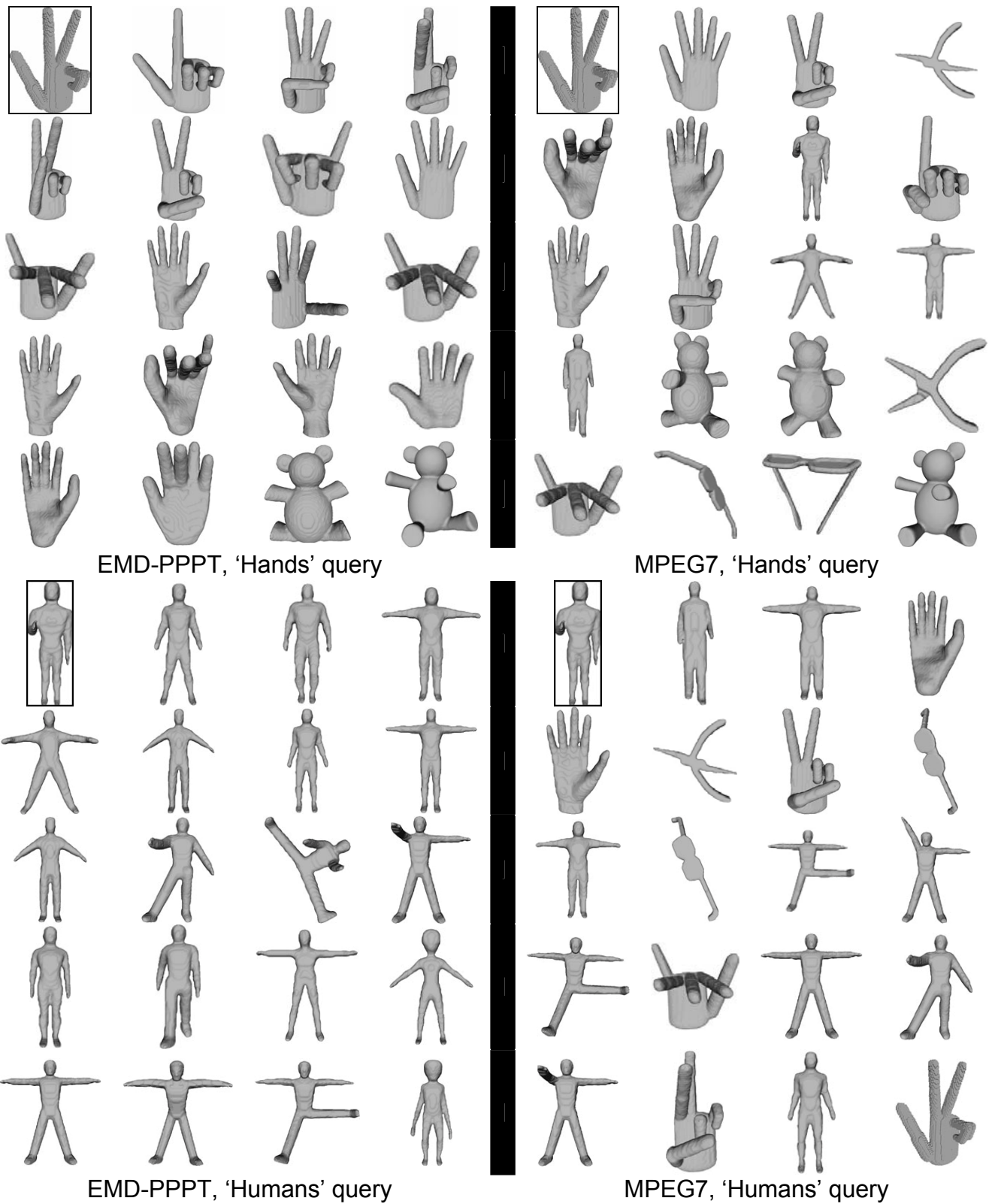


Figure 5.3.4. Retrieval results for a 'hands' and 'humans' query object. The query object is on the top left side and the ranking order goes from left to right.

5.4 Conclusions

In this chapter a 3D object retrieval methodology for articulated objects was presented that is dependant on a graph-based representation of the object. The graph-based representation of the object is constructed by first segmenting the object into its meaningful components and then, using these components, a graph structure of the object is constructed assigning to the nodes and edges proper unary and binary attributes thus constructing an ARG structure. Matching between two ARGs is achieved by a signature matching algorithm using the EMD similarity measure based on newly defined ground distances which are relied on the unary and binary attributes given to the ARGs.

The proposed methodology is very efficient in retrieving articulated objects and was shown to perform significantly better in our extensive evaluation against the compared state of the art retrieval algorithms in the McGill Database of articulated objects. Specifically if 40% and 60% recall levels are considered in the PR diagram for the whole database the following remarks can be stated:

- i. The best precision results are those of EMD-PPPT
- ii. EMD-MPEG7 is the second in precision performance level
- iii. H-EMD-KIM-R, H-EMD-PPPT-R compared to Hybrid has been improved by 20% and 24% approximately.
- iv. SMPEG7 retrieval curve is better than the MPEG7 retrieval curve by an average increase in the precision recall level of the order of 33%

Also it has been shown that in the quantitative evaluation measures like NN, FT, ST, DCG the proposed retrieval methodology using Papadakis *et. al.* [PPPT07] spherical harmonic descriptors (EMD-PPPT) performs best in average for the whole database and it performs better in most of the individual classes of the database. The worst performance, as expected, was attained at the 'snakes' class where the segmentation algorithm fail to correctly segment the members of the class due to their lack of protrusible parts.

Chapter 6

Conclusions

In this dissertation a new 3D object retrieval methodology has been presented based on a new 3D mesh segmentation algorithm and a graph-based representation which is very suitable for retrieval of articulated objects.

In chapter 3 a new mesh segmentation algorithm has been presented. The proposed segmentation algorithm is part-based and partitions the 3D object into its meaningful components. After an extensive experimental evaluation against other state of the art segmentation algorithms, based on established criteria, the proposed segmentation methodology was proven to produce the best results. Also the different stages that the algorithm follows in order to achieve segmentation were shown to be very stable when applied in a plethora of 3D objects without any user intervention. The different threshold values set in the different stages of the algorithm were proven to be correct after a systematic and statistical experimentation.

In chapter 5 a new 3D object retrieval algorithm suitable for articulated objects was presented. The methodology is based on a graph-based representation of the object. Specifically the ARG of the 3D object was constructed based on the segmentation algorithm presented in chapter 3. Retrieval was achieved by matching the ARG representing the query's object with the ARGs representing the objects stored in the database. Matching between two ARGs was accomplished by transforming the graph matching problem into a signature matching problem using the EMD similarity measure based on newly defined ground distances measures. The proposed retrieval algorithm was extensively evaluated using the McGill Database of articulated objects. Different attribute assignments were used for the description of the ARG of the object and the significant superiority of the proposed retrieval methodology against state of the art retrieval methodologies using a plethora of evaluation criteria was established.

Future work involves the extension of the proposed segmentation algorithm in order to do hierarchical segmentation. For example in the segmentation of a mesh representing a human 3D object the first stage of the segmentation hierarchy would be

its partitioning into the main body arms, legs and head. In the next stages of the hierarchy the arm could be further segmented into its hand, fingers, etc.

At the end of this process we must point out the following:

- All “key steps” at the particular phases of the process are optimized in one way or the other in the existing algorithm. The only exception to this rule is the selection we have made regarding the width, $[(1 - d_1) D_{\text{coremin}}, (1 + d_2) D_{\text{coremin}}]$, we have adopted in the section relevant to the partitioning boundary detection, in Chapter 3. Specifically based on a large number of experiments during which we have incrementally changed the specified width it was realized that the set values $d_1 = 0.1$, $d_2 = 0.4$ were the best. In other words an experimental optimization was adopted at this specific point.
- Next to be mentioned is that global optimization process (iteration or trial and error process) is not recommended for the following reasoning:
 - (i) Firstly because we have obtained the best possible results under the given conditions and existing state of the art and,
 - (ii) Secondly, because in an attempt we have made we realized that possible areas of improvement offer small if not trivial interest. For example, in the salient points extraction process if we increase their number in order to improve the ARGs even if some improvement can be obtained it will be counterbalanced by the increase of noise levels; also there is not any significant margin for the improvement of the unary attributes; concerning the binary attributes which express the geometrical features between the parts (main and protrusibles) we expect that it is possible to obtain better results the importance of which must be investigated in a future work.
- Therefore the conclusion which can be made regarding the achievement of the best possible retrieval results for the case of the articulated objects is twofold. Specifically, the first is to extend the segmentation algorithm in order to make it hierarchical; the graph-based representation of the object can be also extended into a tree structure, thus the problem of matching two ARGs can be transformed into the problem of matching trees. The second is to include in the database a well selected number of classes

covering a wide area of existing articulated objects. It is possible to combine these two efforts in order to obtain the best possible cost effective solution to specific demands. There exist also a case in which the improved binary attributes can be applied introducing for example mobility features between the particular parts by the assignment of corresponding degrees of freedom (from one, i.e only one movement to six, i.e three rectilinear and three rotational movements). It is, obviously our intention to extent, if possible, the application of a successful retrieval process into the field of mechanical motion which is governed by geometric requirements of the motion, by dynamic requirements of the forces and by constitutive relations for the deformable parts/elements and velocity-momentum relations for the masses [CKKP68].

Acknowledgement

This thesis is part of the 03 EΔ 520 research project "3D Graphics search and Retrieval", implemented within the framework of the "Reinforcement Programme of Human Research Manpower" (PENED) and co-financed by National and Community Funds (20% from the Greek Ministry of Development-General Secretariat of Research and Technology and 80% from E.U.-European Social Fund).

Publications derived from this PhD Phesis

- A. Agathos, I. Pratikakis, S. Perantonis, N. Sapidis, and P. Azariadis. 3D mesh segmentation methodologies for CAD applications. *Computer-Aided Design and Applications*, 4(6):827–841, 2007
- B. A. Agathos, I. Pratikakis, P. Papadakis, S. Perantonis, P. Azariadis, and N. S. Sapidis. Retrieval of 3D articulated objects using a graph-based representation. In *Eurographics Workshop on 3D Object Retrieval*, pages 29–36, 2009
- C. A. Agathos, I. Pratikakis, S. Perantonis, and N. Sapidis. A protrusion-oriented 3D mesh segmentation. *Visual Computer*. DOI <http://dx.doi.org/10.1007/s00371-009-0383-8>

D. A. Agathos, I. Pratikakis, P. Papadakis, S. Perantonis, P. Azariadis, and N. S. Sapidis. 3D Articulated Object Retrieval using a graph-based representation. *Visual Computer*, **submitted**.

References

- [AA89] S. Andreadakis and P. Azariadis. *Linear Algebra*. Symmetria, 1989.
- [AFS06] M. Attene, B. Falcidieno, and M. Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3):181–193, 2006.
- [AKM⁺06] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal. Mesh segmentation – a comparative study. In *International Conference on Shape Modeling and Applications*, 2006.
- [APP⁺] A. Agathos, I. Pratikakis, P. Papadakis, S. Perantonis, P. Azariadis, and N. S. Sapidis. 3D articulated object retrieval using a graph-based representation. *The Visual Computer*.
- [APP⁺07] A. Agathos, I. Pratikakis, S. Perantonis, N. Sapidis, and P. Azariadis. 3D mesh segmentation methodologies for CAD applications. *Computer-Aided Design and Applications*, 4(6):827–841, 2007.
- [APP⁺09] A. Agathos, I. Pratikakis, P. Papadakis, S. Perantonis, P. Azariadis, and N. S. Sapidis. Retrieval of 3D articulated objects using a graph-based representation. In *Eurographics Workshop on 3D Object Retrieval*, pages 29–36, 2009.
- [APPS] A. Agathos, I. Pratikakis, S. Perantonis, and N. Sapidis. A protrusion-oriented 3D mesh segmentation. *Visual Computer*. DOI <http://dx.doi.org/10.1007/s00371-009-0383-8>.
- [ASBP05] G. Antini, S. Berretti, A. Del Bimbo, and P. Pala. 3D mesh partitioning for retrieval by parts applications. In *IEEE International Conference on Multimedia & Expo*, pages 1210–1213, 2005.

- [BCG08] M. Ben-Chen and C. Gotsman. Characterizing shape using conformal factors. In *Eurographics Workshop on 3D Object Retrieval*, pages 1–8, 2008.
- [Bie87] I. Biederman. A theory of human image understanding. *Recognition-by-Components: A Theory of Human Image Understanding*, 94:115–147, 1987.
- [BMSF06] S. Biasotti, S. Marini, M. Spagnuolo, and B. Falcidieno. Sub-part correspondence by structural descriptors of 3D shapes. *CAD*, 38(9):1002–1019, 2006.
- [CC94] M. Cox and T. Cox. *Multidimensional Scaling*. Chapman and Hall, 1994.
- [CDS⁺05] N. Cornea, M. F. Demirci, D. Silver, A. Shokoufandeh, S. Dickinson, and P. Kantor. 3D object retrieval using many-to-many matching of curve skeletons. In *Proceedings of shape modeling and applications*, pages 366–371, 2005.
- [CKKPB68] S. Crandall, D. Karnopp, E. Kurtz, and D. Pridmore-Brown. *Dynamics Of Mechanical and Electromechanical Systems*. McGraw-Hill Book Company, 1968.
- [CLRS01] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill, 2001.
- [CTSO03] D. Y. Chen, X. P. Tian, Y. T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. *Computer Graphics Forum*, 22(3):223–232, 2003.
- [FBCM04] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nystrom method. *IEEE Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.
- [GG04] N. Gelfand and L. J. Guibas. Shape segmentation using local slippage analysis. In *Eurographics Symposium on Geometric Processing*, pages 214–223, 2004.
- [GSCO07] R. Gal, A. Shamir, and D. Coher-Or. Pose-oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):261–271, 2007.

- [GWH01] M. Garland, A. Willmott, and P. S. Heckbert. Hierarchical face clustering on polygonal surfaces. In *Proc. Symposium on Interactive 3D Graphics*, pages 49–58, 2001.
- [HPW06] K. Hildebrandt, K. Polthier, and M. Wardetzky. On the convergence of metric and geometric properties of polyhedral surfaces. *Geometria Dedicata*, 123(1):89–112, 2006.
- [HR84] D. Hoffman and W. Richards. Parts of recognition. *Cognition*, 18:65–96, 1984.
- [HR97] D. Hoffman and W. Richards. Saliency of visual parts. *Cognition*, 63:29–78, 1997.
- [HSKK01] M. Hilaga, Y. Shinagawa, T. Komura, and T. L. Kunii. Topology matching for full automatic similarity estimation of 3d. In *ACM SIGGRAPH*, pages 203–212, 2001.
- [JZ07] V. Jain and H. Zhang. A spectral approach to shape-based retrieval of articulated 3D models. *CAD*, 39(5):398–407, 2007.
- [KK95] G. Karypis and V. Kumar. Multilevel graph partitioning schemes. In *International Conference on Parallel Processing*, pages 113–122, 1995.
- [KK98] G. Karypis and V. Kumar. Metis: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, version 4.0. Univ. of Minnesota, Dept. of Computer Science, 1998.
- [KLT05] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8-10):649–658, 2005.
- [KPYL04] D. H. Kim, I. K. Park, I. D. Yun, and S. U. Lee. A new mpeg-7 standard: Perceptual 3D shape descriptor. In *5th Pacific Rim Conference on Multimedia*, pages 238–245, 2004.
- [KT03] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM TOG*, 22(3):954–961, 2003.
- [KYL05] D. H. Kim, I. D. Yun, and S. U. Lee. A new shape decomposition scheme for graph-based representation. *Pattern Recognition*, 38(5):673–689, 2005.

- [LLL04] H. S. Lin, H. M. Liao, and J. Lin. Visual salience-guided mesh decomposition. In *Workshop on Multimedia Signal Processing*, pages 331–334, 2004.
- [LLL07] H. S. Lin, H. M. Liao, and J. Lin. Visual salience-guided mesh decomposition. *IEEE Transactions On Multimedia*, 9(1):46–57, 2007.
- [LLS⁺04] Y. Lee, S. Lee, A. Shamir, D. Cohen-Or, and H.P. Seidel. Intelligent mesh scissoring using 3d snakes. In *Pacific Conference on Computer Graphics and Applications*, page 279–287, 2004.
- [LLS⁺05] Y. Lee, S. Lee, A. Shamir, D. Cohen-Or, and H.-P. Seidel. Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design*, 22(5):444–465, 2005.
- [LTH01] X. Li, T. W. Toon, and Z. Huang. Decomposing polygon meshes for interactive applications. In *SI3D*, pages 35–42, 2001.
- [LZ04] R. Liu and H. Zhang. Segmentation of 3D meshes through spectral clustering. In *Pacific Conference on Computer Graphics and Applications*, page 298–305, 2004.
- [LZ07] R. Liu and H. Zhang. Mesh segmentation via spectral embedding and contour analysis. *Computer Graphics Forum*, 26:385–394, 2007.
- [MBA93] G. Malandain, G. Bertrand, and N. Ayache. Topological segmentation of discrete surfaces. *International Journal of Computer Vision*, 10(2):183–197, 1993.
- [mcg] McGill 3d shape benchmark.
- [MDA⁺08] A. Mademlis, P. Daras, A. Axenopoulos, D. Tzovaras, and M.G. Strintzis. Combining topological and geometrical features for global and partial 3-d shape retrieval. *IEEE Transactions on Multimedia*, 2(5):819–831, 2008.
- [MW99] A. P. Mangan and R. T. Whitaker. Partitioning 3D surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):308–321, 1999.
- [OOFB08] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno. Salient local visual features for shape-based 3d model retrieval. In *IEEE Int. Conf. on Shape Modeling and Applications*, pages 93–102, 2008.

- [PAKZ03] D. Page, M. Abidi, A. Koschan, and Y. Zhang. Object representation using the minima rule and superquadrics for under vehicle inspection. In *Proceedings of the 1st IEEE Latin American Conference on Robotics and Automation*, pages 91–97, 2003.
- [PKA03] D. L. Page, A. Koschan, and M. Abidi. Perception-based 3d triangle mesh segmentation using fast marching watersheds. In *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 27–32, 2003.
- [PPPT07] P. Papadakis, I. Pratikakis, S. Perantonis, and T. Theoharis. Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation. *Pattern Recognition*, 40(9):2437–2452, 2007.
- [PPT⁺08] P. Papadakis, I. Pratikakis, T. Theoharis, G. Passalis, and S. Perantonis. 3D object retrieval using an efficient and compact hybrid shape descriptor. In *Eurographics Workshop on 3D Object Retrieval*, pages 9–16, 2008.
- [Pra98] I. Pratikakis. *Watershed-driven image segmentation*. PhD thesis, Vrije Universiteit Brussel, Belgium, December 1998.
- [PRF02] S. Pulla, A. Razdan, and G. Farin. Improved curvature estimation for watershed segmentation of 3D meshes. *IEEE Transactions on Visualization and Computer Graphics*, 2002.
- [PWH89] S. Peleg, M. Werman, and H. Rom. A unified approach to the change of resolution: space and gray-level. *IEEE Pattern Analysis and Machine Intelligence*, 11(7):739–742, 1989.
- [RKS00] C. Rossl, L. Kobbelt, and H.-P. Seidel. Extraction of feature lines on triangulated surfaces using morphological operators. In *In Proceedings of the AAAI Symposium on Smart Graphics*, pages 71–75, 2000.
- [RT08] D. Reniers and A. Telea. Hierarchical part-type segmentation using voxel-based curve skeletons. *The Visual Computer*, 24(6):383–395, 2008.
- [RTG00] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000. <http://www.cs.duke.edu/tomasi/software/emd.htm>.

- [SB95] N. Sapidis and P. Besl. Direct construction of polynomial surfaces from dense range images through region growing. *Transactions on Graphics*, 14(2):171–200, 1995.
- [SBTZ98] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker. The hamilton-jacobi skeleton. In *International Conference on Computer Vision*, pages 828–834, 1998.
- [SBTZ02] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker. The hamilton-jacobi skeleton. *International Journal of Computer Vision*, 48(3):215–231, 2002.
- [SCOGL02] O. Sorkine, D. Cohen-Or, R. Goldenthal, and D. Lischinski. Bounded-distortion piecewise mesh parameterization. In *IEEE Visualization*, pages 355–362, 2002.
- [SH01] M. Singh and D. Hoffman. *In From fragments to objects: Segmentation and grouping in vision: Part-based representations of visual shape and implications for visual cognition*. Elsevier Science, 2001.
- [Sha04] A. Shamir. A formulation of boundary mesh segmentation. In *In Proc. 2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 82–89, 2004.
- [SM00] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [SSCO05] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent partitioning of meshes. Technical report, The Interdisciplinary Center, 2005.
- [SSGD03] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Shape modeling International*, pages 130–139, 2003.
- [STK02] S. Shlafman, A. Tal, and S. Katz. Metamorphosis of polyhedral surfaces using decomposition. In *Eurographics*, pages 219–228, 2002.
- [SWG⁺03] P. V. Sander, Z. J. Wood, S. J. Gortler, J. Snyder, and H. Hoppe. Multichart geometry images. In *In Proc. Symposium on Geometry Processing*, pages 146–155, 2003.

- [TS05] T. Tung and F. Schmitt. The augmented multiresolution reeb graph approach for content-based retrieval of 3D shapes. *International Journal of Shape Modeling*, 11(1):91–120, 2005.
- [TZ06] A. Tal and E. Zuckerberger. Mesh retrieval by components. In *International Conference on Computer Graphics Theory and Applications*, pages 142–149, 2006.
- [VC04] S. Valette and J. Chassery. Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening. *Computer Graphics Forum*, 23(3):381–389, 2004.
- [VKS05] S. Valette, I. Kompatsiaris, and M. G. Strintzis. A polygonal mesh partitioning algorithm based on protrusion conquest for perceptual 3d shape description. In *Workshop towards Semantic Virtual Environments*, pages 68–76, 2005.
- [Vra04] D. V. Vranic. *3D Model Retrieval*. PhD thesis, University of Leipzig, 2004.
- [VS91] L. Vincent and P. Soille. Watershed in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.
- [vW03] B. J. van Wyk. *Kronecker Product, Successive Projection and Related Graph Matching Algorithms*. PhD thesis, Univerisy of the Witwatersrand, Johannesburg, May 2003.
- [WL97] K. Wu and M. D. Levine. 3D part segmentation using simulated electrical charge distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1223–1235, 1997.
- [ZH04] Y. Zhou and Z. Huang. Decomposing polygon meshes by means of critical points. In *Multimedia Modelling Conference*, page 187–195, 2004.
- [ZL05] H. Zhang and R. Liu. Mesh segmentation via recursive and visually salient spectral cuts. In *Vision, Modelling, and Visualization*, pages 429–436, 2005.
- [ZPKA02] Y. Zhang, J. Paik, A. Koschan, and M. A. Abidi. A simple and efficient algorithm for part decomposition of 3d triangulated models based on curvature analysis. In *Proc. Intl. Conf. on Image Processing*, pages 273–276, 2002.

- [ZTS02] E. Zuckerberger, A. Tal, and S. Shlafman. Polyhedral surface decomposition with applications. *Computers & Graphics*, 26(5):733–743, 2002.