Dipartimento di Informatica, Sistemistica e Telematica

3D Shape Similarity Through Structural Descriptors

by

Simone Marini

DIST, Università di Genova

Via all’Opera Pia 13 - 16145 Genova, Italy http://www.dist.unige.it/

Università degli Studi di Genova

Dipartimento di Informatica, Sistemistica e Telematica

Dottorato di Ricerca in Scienze e Tecnologie dell'Informazione e della Comunicazione

Ph.D. Thesis in Science and Technology of Information and Communication

3D Shape Similarity Through Structural Descriptors

by

Simone Marini

May, 2005

Sommario

A causa dei miglioramenti tecnologici dei settori dell'acquisizione, visualizzazione e modellazione di oggetti tridimensionali, il numero di modelli 3D attualmente disponibili sta crescendo enormemente. Allo stesso modo, sta incredibilmente crescendo la domanda di strumenti per la ricerca automatica di oggetti e sotto parti di oggetti 3D. Mentre attualmente esistono tecniche in grado di estrarre automaticamnte la conoscenza codificata in volumi massivi di testo (ad esempio google), è molto più difficile strutturare, filtrare, organizzare, estrarre informazione e mantenere archivi di forme digitali come immagini, oggetti 3D, animazioni 3D in contesti di realtà virtuale oppure aumentata. Queste considerazioni suggeriscono che la sfida principale per il futuro della grafica comutazionale consiste nella ricerca di modelli aventi simile apparenza globale e/o locale.

Per queste ragioni i descrittori di forma e le metodologie utili al loro confronto occupano un ruolo cruciale all'interno della grafica computazionale. Il primo contributo di questa tesi consiste in una analisi critica dello stato dell'arte dei più rappresentativi descrittori di forma, rispetto ad un insieme di proprietà, utili ad evidenziarne le differenze in modo da comprendere dove e quando un descrittore fallisce mentre un altro ha successo.

Il secondo contributo della tesi consiste nello studio del problema del confronto tra descrittori di forma strutturali come strumento per il confronto di oggetti 3D. Molti descrittori di forma strutturali possono essere codificati come grafi diretti e a-ciclici con attributi, quindi il problema del confronto tra descrittori strutturali può essere facilmente risolto con tecniche di confronto tra grafi. Le tecniche per il confronto tra grafi hanno complessità computazionale esponenziale, è quindi necessario definire algoritmi euristici, computazionalmente efficienti, in grado di fornire come risultato una approssimare della soluzione ottima. Le metodologie comunemente utilizzate nella grafica computazionale, per il confronto tra descrittori strutturali, adottano tecniche euristiche definite per applicazioni specifiche, mentre, al momento, manca una metodologia per il confronto di descrittori strutturali capace di incorporare differenti tecniche euristiche da potere utilizzare in differenti contesti applicativi. In questa tesi viene quindi proposto un algoritmo originale per il calcolo del massimo sottografo comune in grado di incorporare in modo modulare differenti tecniche euristiche che lo rendono adattabile a diversi contesti applicativi. Inoltre viene proposta una nuova misura di similarità tra grafi le cui proprietà vengono confrontate e discusse con quelle di altre misure già esistenti.

Infine, le differnti tecniche euristiche vengono discusse e ne vengoono illustrati i risultati nell'analisi

della corrispondenza tra sottoparti comuni di oggetti 3D e nel recupero di oggetti 3D da basi di dati, attraverso l’utilizzo di diversi descrittori di forma strutturali.

Abstract

Due to the recent improvements to 3D object acquisition, visualization and modeling techniques, the number of 3D models available is more and more growing, and there is an increasing demand for tools supporting the automatic search for 3D objects and their sub-parts in digital archives. Whilst there are already techniques for rapidly extracting knowledge from massive volumes of texts (like Google [htt]) it is harder to structure, filter, organize, retrieve and maintain archives of digital shapes like images, 3D objects, 3D animations and virtual or augmented reality. This situations suggests that in the future a primary challenge in computer graphics will be how to find models having a similar global and/or local appearance.

Shape descriptors and the methodologies used to compare them, occupy an important role for achieving this task. For this reason a first contribution of this thesis is to provide a critical analysis of the most representative geometric and structural shape descriptors with respect to a set of properties that shape descriptors should have. This analysis is targeted at highlighting the differences between descriptors in order to better understand where a descriptor fails and another succeed.

As a second contribution, the thesis investigates the problem of using a structural descriptor for shape comparison purposes. A large class of structural shape descriptors can be easily encoded as directed, a-cyclic and attributed graphs, thus the problem of comparing structural descriptors is approached as a graph matching problem. The techniques used for graph comparison have an exponential computational complexity and it is therefore necessary to define an algorithmic approximation of the optimal solution. The methods for structural descriptors comparison, commonly used in the computer graphics community, consist of heuristic graph matching algorithms for specific application tasks, while it is lacking a general approach suitable for incorporating different heuristics applicable in different application tasks. The second contribution presented in this thesis is aimed at defining a framework for expressing the optimal algorithm for the computation of the maximal common subgraph in a formalization which makes it straightforward usable for plugging heuristics in it, in order to achieving different approximations of the optimal solution according to the specific case.

Implemented heuristics for robust graph matching with respect to graph structural noise are discussed and experimented on sub-part correspondence between similar 3D objects, and shape retrieval application with respect to different structural graph descriptors.

Acknowledgments

This thesis represents four year of work at the Istituto di Matematica Applicata e Tecnologie Informatiche of the Consiglio Nazionale delle Ricerche. During this period I have met many people, and I have to sincerely thank all of them for their support over the doctoral course, thus I apologize in advance for any I have not mentioned here.

I especially thank my supervisors Bianca Falcidieno and Michela Spagnuolo for guiding and encouraging me during these years. I thank them for the useful discussions and because they shared with me their scientific experience.

A special thank goes also to Remco Veltkamp, Thomas Funkhouser Jarek Rossignac and Patrick Min for the interest they showed in my work and the helpful suggestions they gave me during the conversations on skeletons and shape comparison.

I would to thank all of the teachers I met, the students and the staff of the Dipartimento di Informatica, Sistemistica e Telematica of the Università di Genova.

This work has been partially supported by the National Project ”MACROGeo: Metodi Algoritmici e Computazionali per la Rappresentazione di Oggetti Geometrici”, FIRB grant and the FP6 Network of Excellence ”AIM@SHAPE”.

I would like also to thank the other researchers with whom I shared my work, in particular my colleagues and coauthors Silvia Biasotti, Giuseppe Patané, Michela Mortara. I also thank all the people of the Istituto di Matematica Applicata e Tecnologie Informatiche for their kind friendship, in particular Marco Attene, Chiara Catalano, Riccardo Albertoni, Alessio Bertone Maria Grazia Ierardi, Francesco Robbiano, Corrado Pizzi for his helpful technical support, Marinella Pescaglia and Sandra Burlando for their helpful management of all administrative stuffs.

A special thank goes to Marina that carried me in the world of the research and to my mother and my father that encouraged me since the beginning of my studies.

For Marina, the beautiful companion of my life
that will become soon my beautifully wife forever.

Contents

Listings

List of Figures

Introduction

The impact of emerging technologies leads to a gradual but inescapable reduction of the material essence of our reality in a process that we can call dematerialization, that is a progressive contraction of the universe of material objects, substituted by processes and services more and more immaterial. Therefore it is evident the crucial role that shape models play in this process, the relationship between object, model and representation, is a fundamental passage in the evolution of our scientific reasoning. This new relationship between reality and representations is raising new interest for matters already discussed in the past, and it is posing new questions on how to fill the gap between technology and user.

The need to extract knowledge from massive volumes of digital content is rapidly increasing and new forms of content are coming into evidence, such as 3D animations and virtual or augmented reality. Whilst it has become relatively easier to generate 3D information and to interact with the geometry of shapes, it is harder to structure, filter, organize and retrieve it. These considerations are changing the approach to 3D object modeling.

Until now a primary challenge in computer graphics has been how to build and render complete and effective models; now the key issue is how to find, interpret and interact with them. In this sense, methods for automatically extracting the semantic content of digital shapes will become increasingly necessary. This will allow browsing the web or digital object repositories using enhanced search engines based not simply on text-searches but on shape and semantics (e.g. content and context based search engines capable of answering semantics-based queries) as proposed in the AIM@SHAPE NoE[1]. In this context, the knowledge about a digital shape may be organized at three different levels of representation: geometric, structural and semantic levels [FS98b].

A first organization of shape data into a computational structure gives access to the *geometric level of representation*, where different types of geometric information can be coded in a computer processable structure and used to represent the same object shape. Examples of geometric descriptors encoding the shape object through its spatial distribution, harmonic decomposition of the object extent, weighted point-set or set of 2D images representing the object views, are shown in chapter 2.

[1] EU, FP6 - AIM@SHAPE Network of Excellence. Advanced and innovative models and tools for the development of semantic-based systems for handling, acquiring, and processing knowledge embedded in multidimensional digital objects, http://www.aim-at-shape.net

Then, a *structural level of representation* is reached by organizing the geometric data to reflect and make explicit the decomposition of the shape object into its main parts/components, each one presumably simpler than the whole object, nevertheless, providing a global description of the shape object. This approach relates to the human perception theories proposed by Marr [Mar82] and Biedermann [Bie87, Bie95], where experimental results are used for showing that people, when interpret the meaning of a novel scene, attend only to a few details and recognize an object on the basis of basic level shapes. In particular, they assume that each basic shape may be represented as the combination of a few generalized primitives called geons. Finally, a shape representation that specifies parts, attributes and relations between them, independently and explicitly is called a *structural decomposition*. Examples of structural descriptors as skeletons obtained from volumetric thinning and Reeb graphs, are shown in chapter 3.

At the highest level of segmentation, the *semantic* one, there is the association of a specific meaning to structured and/or geometric models through annotation of shapes, or shape parts, according to the concepts formalized by the semantic domain. Therefore, a semantic model is the representation of a shape embedded into a specific context or domain of knowledge.

The two issues discussed in this thesis investigate first the differences between shape descriptors of the geometric and structural level, with respect to the set of properties that descriptor should have; second the problem of using a structural descriptor for shape comparison purposes. In order to address the first issue of the thesis it is necessary to investigate how geometric and structural shape descriptors can be associated to an object model. Moreover, it is desirable that the shape descriptor captures the salient and relevant features, concisely represents the information stored, is invariant to affine transformations, is suitable for multi-scale analysis and is computationally efficient and simple to store. Furthermore, it has to provide a good support for associating semantics to the object description. The second issue of the thesis is the discussion of the advantages of considering graphs as shape descriptors, in a matching framework able to modularly incorporate different heuristic techniques for reducing the computational costs of the graph matching and for adapting the similarity estimation to the application context for which it is used.

Contribution

In the computer graphics community many shape descriptors have been investigated, and they are mainly devised for shape analysis, shape similarity, animation, reconstruction and editing.

In this thesis, shape descriptors are coarsely classified into two principal categories: geometric and structural shape descriptors. Descriptors belonging to these two categories have both advantages and disadvantages with respect to the application they are used for. For example, the first three models in 1 are better equipped in modern motorboats while the others are best suited for traditional ones. The rudders have a common overall shape, but their structure influences the way the helmsman may grasp them. In a shape retrieval context, one possible goal could be to retrieve all rudders belonging to a given repository (possibly containing a huge number of models), and being similar to a given query rudder. In this case, the descriptor should be able to capture the salient features that discriminate among the class of rudders and the other models, and also the descriptor has to be concise in order to efficiently perform during the comparison of the models. Another possible application context deals with the capability of a virtual human to understand how to grasp a rudder. In this case the descriptor has to provide the necessary information to compute which geometric transformation the virtual human has to perform in order to pose the hands on the rudder. Both geometric and structural descriptors can positively contribute to the two application contexts of the example, even if by capturing different characteristic aspects of the shape objects. For this reason it is important to have a critical survey of such descriptors with respect to a set of *shape descriptors properties*, in order to highlight the differences and better understand where a descriptor fails and another succeed. Therefore, the first contribution of this thesis is the critical survey of the most representative geometric and structural shape descriptors (chapters 2 and 3 respectively) with respect to the following properties (detailed in section 1.4):

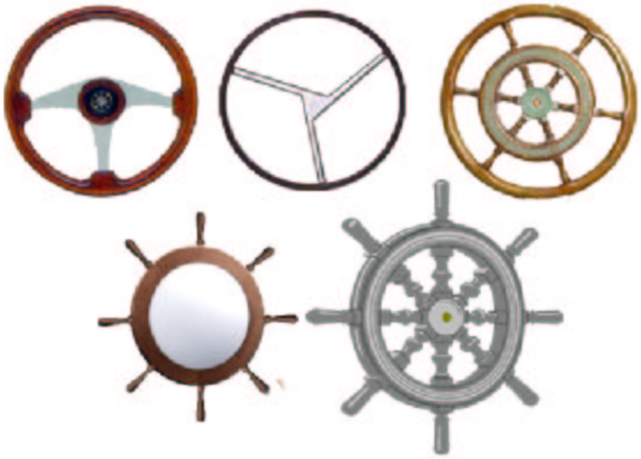


Figure 1: *These rudders can be grasped in different ways.*

- *saliency*: is the capability of the descriptor to characterize the shape object;
- *relevance*: the salient features may be relevant or not depending on the application context;
- *conciseness*: is the capability to minimize the memory needed to store the descriptor, maximizing the amount of information represented;
- *uniqueness*: depending on the methodology used, the shape descriptor associated to an object may be not unique;
- *invariance to affine transformation*: depending on the application context, the shape descriptor have to be invariant to rotations, translations and scaling transformations;

- *invariance to model representation*: the object to be described, can be represented by a 2-manifold polygonal mesh, a polygon soup, a volumetric model, etc. Not all the methods used to extract the descriptors are able to manage these differences;
- *scalability*: is the capability of a shape descriptor to represents the stored information at different levels of detail;
- *meaning of the information stored*: shape descriptors have to provide the suitable information needed to accomplish an application task.

The second contribution of the thesis deals with shape comparison through structural shape descriptors coded as directed a-cyclic graphs (chapter 4). Graph matching has been used extensively in a variety of applications and it is particularly useful when the graphs code the structure of a shape. By graph matching, or more commonly by subgraph isomorphism, it is possible indeed to assess the similarity among shapes as well as among parts of a shape. Inexact graph matching is also very important for matching structural descriptions of shapes because small features of a shape can cause small differences in the structural descriptions, while similarity should be assessed with stability with respect to noise.

While shape descriptions are usually coded into graph with a relatively small number of nodes, shape databases are composed by hundreds of models of different types and there exists no single similarity measure which optimizes the retrieval results for all shape types. Conversely, heuristics and algorithm flexibility should be left to the user in order to tune the matching to the particular context.

Inexact graph matching has been topic of research since many years and several techniques are available: recently, Demicri et al. in [DSD$^+$04] reduce the problem of inexact directed graph matching to the problem of geometric point matching. They use the Earth Mover's Distance as many-to-many matching algorithm among the points. In [MB98] Messmer and Bunke defined an algorithm for the error correcting subgraph isomorphism (ECSI) detection where the two input graphs are recursively decomposed into smaller subgraphs and the ECSIs with the least edit cost operation are recursively combined to form the complete node matching. Another widely investigated form of inexact graph matching method is the maximum common subgraph (MCS) detection. For example, three algorithms for the exact computation of the MCS are presented in [CGC03]: the first is a state space representation (SSR) algorithm performing a depth-first search in the space of the states, while the other two detect the maximum clique of the association graph built from the two input graphs. In [CFSV04] another SSR algorithm for the MCS detection among large graphs is presented. It moves from a generic state to the following one selecting a candidate pair of nodes according to a set of feasibility rules guaranteeing that each state is a common subgraph of the two input graphs.

Many of these techniques have exponential computational complexity and it is therefore necessary to define an algorithmic approximation of the optimal solution. The work presented in this thesis is aimed at defining a framework for expressing the optimal algorithm in a formalization which makes it straightforward usable for plugging heuristics in it, to achieve different approximations of the optimal solution according to the specific application case. An optimal algorithm for the computation of the maximum common subgraph is defined on a slight modification of the most naive algorithm: starting from the list of all mappings among graph nodes, the common subgraphs is grown from each mapping through a process which attempts to add, step by step, more nodes to the empty initial common subgraph, by expanding at each step two isomorphic subgraphs. This approach may be also seen as a generalization of the state space representation (SSR) algorithm proposed in [BMM$^+$03].

In chapter 4, it will be described in more details how the subgraph expansion works; it will be shown that using this procedure it will be obtained exactly the same results of the naive algorithm. Some useful heuristic techniques are also introduced that approximate the optimal solution and improve the computational performances of the similarity evaluation, moreover a similarity measure between attributed graphs is shown and discussed with respect to the metric properties. Finally experiments on sub-parts correspondence among shape objects and shape retrieval will be shown in chapter 5.

Overview

Before going through the detailed layout of the thesis, some general remarks about its structure are provided: chapter 1 shows the problem statement of shape perception, representation and comparison; chapters 2 and 3 are a critical survey and analysis of geometric and structural shape descriptors; chapter 4 defines the new framework for comparing structural shape descriptors; chapter 5 shows two application contexts where the framework for comparing structural shape descriptors has been applied; appendix A includes the basic definitions on graph theory needed in the thesis. Each chapter starts with a short preamble motivating the discussed topics and it ends with a critical discussion on its content. The thesis ends comparing the main results, drawing the conclusion and with a perspective for possible future work. In more detail this dissertation is organized as follows:

Chapter 1. Section 1.1 presents how shape concepts are perceived and represented by psychologists. Section 1.2: is a short survey on the perception of similarity and on the properties that a similarity measure has to satisfy, as psychologists have formalized. In section 1.3 the distance properties commonly used in the computer graphics community are described. Section 1.4 draws what is a shape descriptor and which are the properties that it has to satisfy. Finally section 1.5 summarize the topics present in the chapter.

Chapter 2. Section 2.1 describes two examples of statistical geometric shape descriptors. Section 2.2 shows four kinds of deterministic geometric shape descriptors. In section 2.3 the methodologies used to compare the geometric shape descriptors mentioned in the chapter are explained and discussed. Finally section 2.4 analyzes the geometric shape descriptors mentioned in the chapter with respect to the properties drawn in section 1.4.

Chapter 3. In section 3.1 two of the most used structural descriptors for two-dimensional objects are shown and discussed. Section 3.2 is a survey of the most meaningful three-dimensional structural shape descriptors. Section 3.3 shows the inexact subgraph isomorphism techniques used for the evaluation of the shape similarity between the structural shape descriptors mentioned in the chapter. Section 3.4 analyzes the structural shape descriptors mentioned in the chapter with respect to the properties drawn in section 1.4.

Chapter 4. Section 4.1 formalize the problem of comparing structural descriptors encoded as graphs. Section 4.2 provides the pseudo-code description of the algorithm for comparing the structural descriptors and the proof of its correctness. Section 4.3 shows the heuristics proposed to improve the computational and qualitative performance of the algorithm. Section 4.6 analyzes the similarity measures between two graph and the properties they have to satisfy. Section 4.7 summarize the results shown in the chapter.

Chapter 5. Section 5.1 shows how the framework proposed in chapter 5 behaves with respect to the detection of the sub-parts correspondences between two similar objects. Section 5.3 shows which are the performance of the framework proposed in chapter 5, with respect to the shape retrieval application contexts.

Appendix A. Section A shows the basic definitions needed to understand the terminology dealing with graphs, used in the thesis.

Chapter 1

Shape, Representation and Similarity

1.1 Shape Perception and Representation

In the daily life we can immediately recognize the use of different levels of mental models for describing geometric objects: it is possible to use natural-language terms to qualitatively describe external shape, or to draw the object itself, or to describe it by listing its differences with respect to some other similar objects, or also to define it according to what it is used for, and so on. The common characteristics of these mental models is that they all generally refer to shape or function, that is, they are different ways of answering questions such as ”What does it look like?”, ”What is its meaning?”. Moreover, experiments in human perception have suggested that people use different high (specific) and low (generic) level models for shape interpretation. In this section two approaches for shape representation, addressed by psychologist and computer scientist, will be outlined.

In [Ede96] Shimon Edelman proposes the following questions: Which is the internal state of an observer seeing a cat? What would make it refer to the shape of the cat? A reasonable answer to this questions states that the internal entity represents an external object as resemblance or isomorphism between the two: the representation of a tomato has something of the redness and of the roundness of the real tomato. The same interpretation of shape representation is proposed in [SMF94]: ”a representation of something is defined as an image, a model or reproduction of that thing”. Obviously, no one believe that the internal brain representation of a cat is ”cat-shaped”, for example stripped or fluffy. Rather, it should be a set of measurements which collectively encode the geometry and other visual qualities of a cat. A different point of view on the concept of shape representation can be guessed from the following dictionary [oCE87] definition of the word ”shape”: ”the outer form of something by which it can be seen (or felt) to be different from something else”. This definition is rather interesting as it establishes a link between the concept of shape and the concept of similarity, in accordance with the common use of shape reasoning not only for describing an object but also for distinguishing among objects. It is based on the concept that similarity equivalence makes shapes emerge from an object.

Before to show the two main conceptualizations of shape perception raising from the previous observations, let us consider the differences between the concepts of portrait, caricature and silhouette, in order to better understand the role of shape in human perception. In figure 1.1 three pictorial examples representing a cat are shown. Beside its artistic nature, the portrait 1.1(a) represents the cat as is, trying to add all the details needed to obtain a realistic result. The caricature in figure 1.1(b), contains much less data with respect to the portrait, but communicates more: for instance, what the artist has recognized as the main feature of the animal or what has caught his/her attention. While some shape features are exaggerated and some are simplified, matching the real face and its caricature is usually immediate. Finally, the silhouette in figure 1.1(c) can be seen as a very concise description of the shape, without a precise emphasis on any particular detail. It is just a simplification of the overall shape. Both the caricature and the silhouette are concise and highly communicative models which provide clear evidence that only few data are sufficient to effectively communicate a high-level set of information about the object. The portrait, caricature and silhouette are examples of different shape-based models which are representative of typical levels at which we reason about objects in a broader sense.

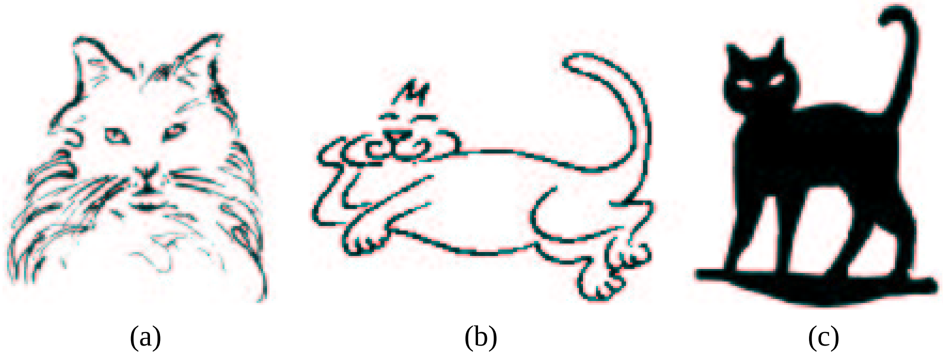


Figure 1.1: *Different pictorial representations of a cat. Portrait (a), caricature (b), silhouett (c).*

These observations highlight how different representations of the entity cat may be used in different application contexts. The next two sections show two orthogonal approaches to shape representation.

1.1.1 Resemblance Between the Object and its Representation

One way to represent the shape of real objects is based on the following observation: in some sense, an object representation should be isomorphic to the real object itself. The main motivation of this approach lies on the fact that it is possible to model the world in term of parts: macroscopic models that, in relatively simple combination, can be used to form rough models of the object in our world and how they behave. This approach comes up from the observation that the evolution repeat its solutions whenever is possible, resulting in great regularities across all species. There are a few type of limb, a few type of skin, a few patterns of branching. For instance a tree can be represented as a composition of a simple branching process and the same branching models can be used for rivers, veins or coral. This internal structure of the natural object allows clustering of objects into groups and simplified category description for common sense reasoning [Pen86]. In [TH84] Tversky and Hemenway, noticed that spontaneous description of people, almost always, include specification of distinctive object parts, for instance, leg, tail and trunk of an elephant or the shadow from the base of a lamp.

The decomposition into parts does not depend on the familiarity with the object. As shown in [Bie87], different observers agreed on the shape decomposition of nonsense objects. Nor it depends on surface color or texture. In general, discontinuity at minima of negative curvature, are guessed from human subjects as regions where the object can be segmented. To strengthen this point of view, in [HR85], has been noticed that whenever two shapes are combined, their joins are almost always marked as matched cusps (minima of negative curvature), as shown in figure 1.2(a) and 1.2(b). In [STK94] the neck (narrowing of the shape without minima at negative curvature) has been defined as another basis for object decomposition. From this observation an animal's neck provide a natural region for shape decomposition between the head and the shoulders. Segmenting at such regions (matched cusps and necks), is the basis for the Gestalt principle of a good figure: if a shape is decomposed at matched cusps or necks, the resulting parts will be convex or only concave. Such parts appear simple.

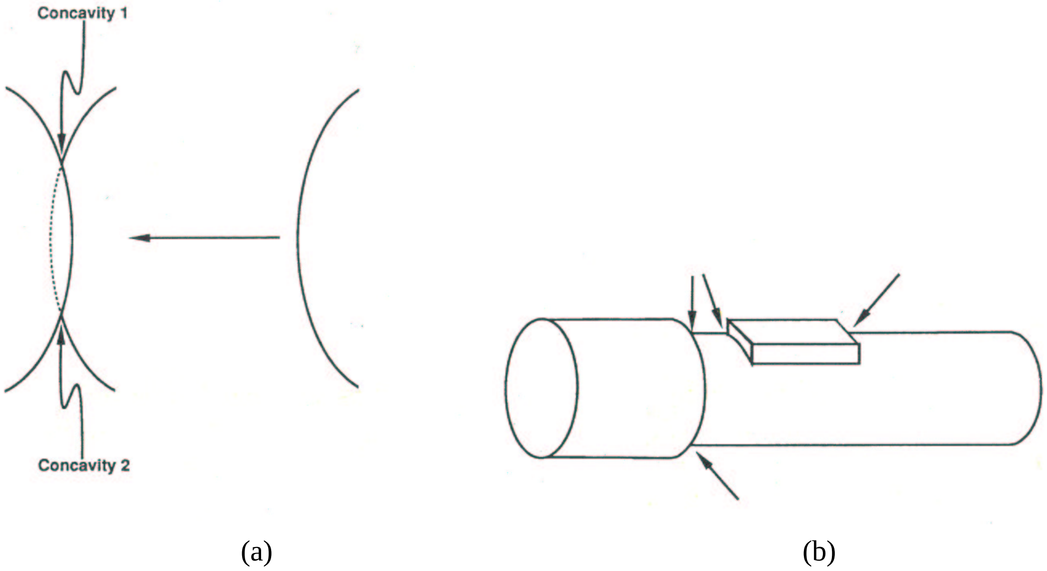


Figure 1.2: *Discontinuities at minima of negative curvature (a) and object segmentation at such discontinuities (b). The figure is from [Bie95].*

By this point of view, the central problem for shape representation is to find a set of generic part models, complex enough to be recognizable and yet simple enough to be used as building blocks for specific object models, in order to find regularities that are lawfully associated with individual parts. This representation approach should be used to recognize the content of a model as combination of these generic primitives. Pentland, in [Pen86], used as primitives a family of *superquadrics* controlled by two parameters. That family of functions includes cubes, cylinders, spheres, diamonds and pyramidal shapes (figure 1.3(a)), as well as the intermediate shapes between these standard shapes (figure 1.3(b)). The most primitive notion, for the Pentland representation, may be thought as a ”lump of clay”, that may be deformed and shaped, but which is intended to correspond roughly to the naive perceptual notion of ”part” (figure 1.3(c)).

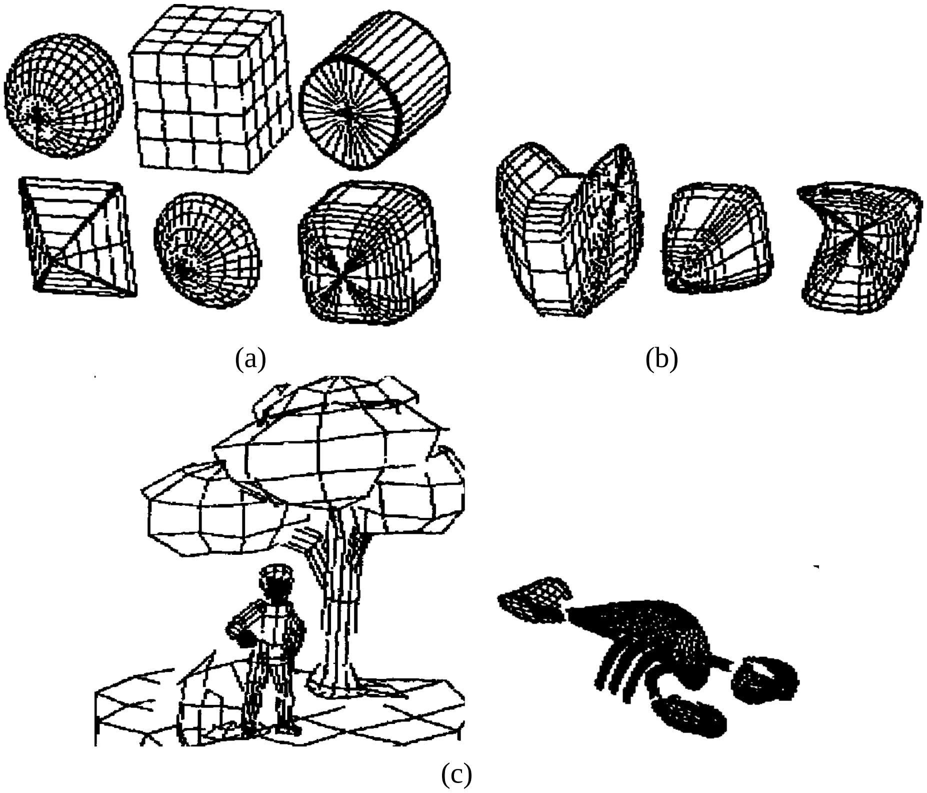


Figure 1.3: *A subset of superquadrics (a). Shapes obtained deforming the primitives through control parameters (b). Two example of rough decomposition of objects (c). The figure is from [Pen86]*

Another approach similar to the ”lump of clay” proposed by Pentland is the one described by Biederman in [Bie87]. He propose to represent an object as a set of view-point invariant volumetric primitives called *geons*, where a geon is a convex or concave volume that can be modeled as a generalized cones. Biederman identified twenty-four geons, five of that are shown in figure 1.4(a). The relationships among geons are specified, so that the same geons in different relation will represent different objects, as the cup and the pail in figure 1.4(b). According to the opinion of Biederman, much of the capacity to represent the tens of thousands of objects that people can classify from a

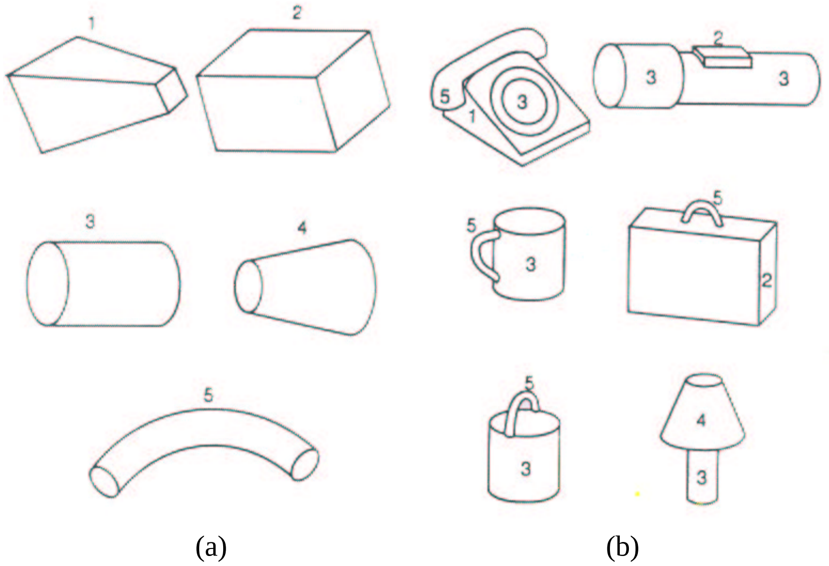


Figure 1.4: *Five geons (a). Some example of objects obtained by combining geons (b). The figure is from [Bie95].*

small alphabet of geons, derives from several view-point invariant relations between pairs of geons and some coarse metric attributes of individual geons. Some example of the relation that have been hypothesized are:

- **vertical position**: above, below, beside;
- **join type**: end-to-end, end-to-middle centered, end-to-middle off-centered;
- **relative size**: larger, smaller, equal to;
- **relative orientation**: parallel, orthogonal, oblique;

These relations are defined for joined pairs of geons, so that the same subset of geons can represent different objects. Two other coded metric aspects has been specified:

- **relative-aspect ratio**: five levels of the length of the axis compared to the diameter of the cross-section;
- **orientation**: vertical, horizontal, oblique;

In [Bie95] it has been also observed that with twenty-four possible geons, eighty-one combinations of relations and fifteen attributes, the variation in relations and aspect ratio can produce $24^2 \times 15^2 \times 81 = 10,497,600$ possible two-geon objects. A third geon, with its attributes and its relations to one other geon, yields 306 billion possible three-geon objects. Since there are approximatively three thousand familiar concrete objects that can be identified on the basis of their shape, surfaces properties, texture and position in a scene [Bie87], and supposing such objects are homogeneously distributed through the space of the possible objects, then, it means that two or three geons would almost always be sufficient to specify any object.

In accordance to Biederman, a representation that specifies parts, attributes and relations, independently and explicitly, is termed *structural description* [Bie95].

1.1.2 Second-order Isomorphism

Another point of view on shape representation, [SC70], states that there is no structural resemblance between an individual internal representation and its corresponding external object. Considering for instance a square, a human subject, can say that it is not only a square but, that it is *seen* or remember it as a square. This may erroneously lead to suppose that the observer must, at such time, have an internal representation or mental image that is itself square (reconstructionist approach). This interpretation of the structural resemblance between external and internal stimuli has been rejected by psychologists [Ski45], that have pointed out that the appropriate use of words as ”square” has been learned from community that has access only to public object and not to any such private image. The ability to form an association between the internal event corresponding to the perception of a square and the word ”square” requires a relation of causality, and not a structural isomorphism, between the internal event and the external object. In other words, the event could be the activation of some group of neurons, but these neurons do not need to be spatially arranged in the form of a square, in order to trigger the naming response ”square”. Nevertheless an approximate parallelism should hold between the relations among different internal representations and the relations among their corresponding external objects. The crucial step consists in accepting that the isomorphism should be sought not in the sense of a *first-order* relation between:

1. an individual object
2. and its corresponding internal representation

but in the *second-order* relation between:

1. the relations between alternative external objects
2. and the relations among their corresponding internal representations

Thus, although the internal representation of a square need not itself be a square, it should, whatever it is, at least have a closer functional relation to the internal representation of a rectangle, where by functional relation between two internal representations is meant the tendency of a response that has been associated with one to be aroused, also, by the activation of the other. Essentially, this is a representation **of** similarity instead of representation **by** similarity. In Figure 1.5 is shown a graphical representation of the second-order isomorphism concept. The relation (W1) in the world, represents the coincidence between sensory events, while the arrows (B1) from world to brain, represent sensory transduction. Both the objects (B2) and the relation (B3) between them are represented in the brain.

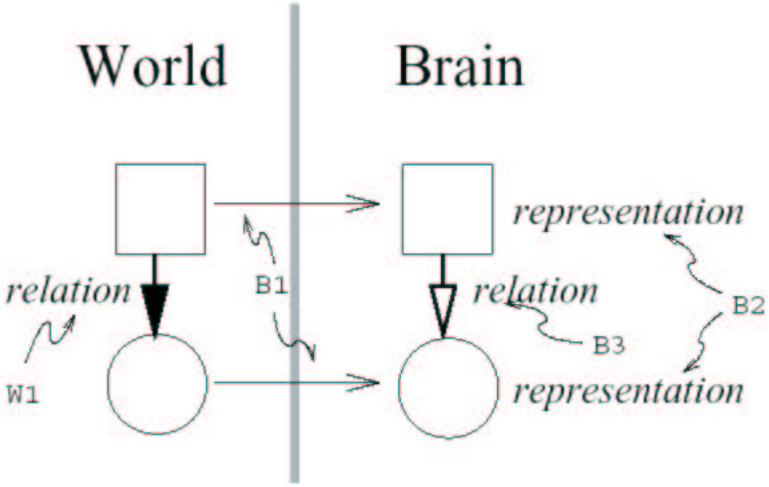


Figure 1.5: *Graphical representation of the second-order isomorphism. Two objects, one square and one round, in the world (on the left) and their correspondent brain representation (on the right). The figure is from [Cho02].*

In [Ede96], Edelman proposes a computational basis for the representation of similarity (second-order isomorphism) that it is based on the following observation: In the case of the representation by similarity, the external information amounts to reconstruction of the visual world, while for representation of similarity, to represent a collection of objects means to reflect in a consistent manner any change that an object may undergo. This notion of representation is conceptually orthogonal to the reconstructionist approach, because the tokens standing for objects need not to resemble the object themselves. The tokens, need only to support perceptual judgment and categorization.

By the algorithmic point of view, the second-order isomorphism have to assure that similarities between (proximal) entities correspond in some orderly fashion to the similarities between represented (distal) shapes, see figure 1.6. In this sense, two similarity functions, one between distal (represented) shapes and the other for proximal (representing) entities, have to be defined. Edelman states that similarity between represented shapes can be defined via an embedding of the objects into a metric space, where the similarity is determined by the distance between the points corresponding to each object. According to this view, changing a shape corresponds to a movement of the point encoding the shape in an appropriate parameter space. The definition of the similarity function in the proximal space is more problematic. The problem of the judged similarity between perceived entities has been deeply addressed by psychologist and will be discussed in section 1.2. What psychologist pointed out is that, in a variety of perceptual tasks, subjects behave as if they represent the stimuli as distributions of points in an internal similarity space suitable for the definition of the similarity function between perceived entities. A graphical representation of distal and proximal spaces is shown in figure 1.6.

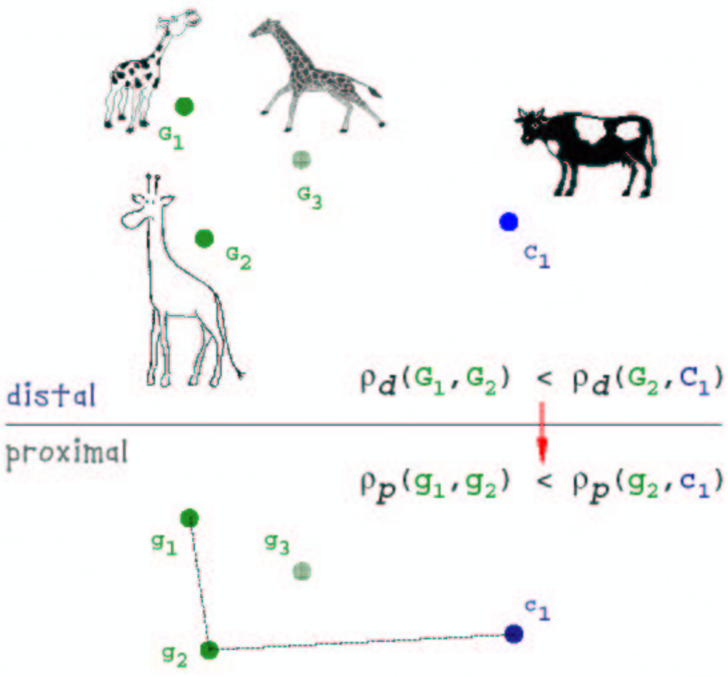

G1
G3
G2
C1
distal
$\rho_d(G_1, G_2) < \rho_d(G_2, C_1)$
proximal
$\rho_p(g_1, g_2) < \rho_p(g_2, c_1)$
g1
g3
c1
g2

Figure 1.6: *An example of proximal and distal spaces. In the proximal space an object representation do not need to resemble the perceived entity beelining to the distal space. The figure is from [EDB97].*

For Edelman, the structure of the world is never perceived directly, but always through the more or less distorting channel of the distal to proximal mapping. The minimal set properties that this mapping should satisfy are:

- **Distinctness**: distinct points in the original space are mapped to distinct points in the representation space
- **Nearest-neighbor preservation**: two points that are nearest-neighbor of each other before the mapping, remain nearest-neighbor after the mapping. All objects more similar to some object O_1 than to O_2 will be represented as such, rather than merely as distinct both from O_1 and from O_2.
- **Full similarity spectrum preservation**: preservation of the rank order among points. The identity of the kth nearest-neighbor of each point is preserved for some $k > 1$. If this constraint is satisfied, the mapping is a similitude. In figure 1.6 is shown that the similarity distance ρ_p,

defined on the proximal space among the objects G_1, G_2 and C_1, preserve the rank among the same perceived entities in the distal space evaluated through the similarity distance ρ_d.

A possible mapping from distal to proximal space, satisfying the previous properties is represented, for Edelman, by $M = f_4 \circ f_3 \circ f_2 \circ f_1$, where the function $f_1(p)$ maps the distal description p of the object into its geometry (e.g. the coordinates of the vertexes of a mesh), $f_2(p; z)$ maps the geometry into the image on the receptor surface of the visual system (in biological vision, one may think of the space of pattern transmitted by the optic nerve to the brain), where p is obtained by the action of f_1 and z represents the viewing conditions (e.g. pose of the object with respect to the observer, to the light sources, and to other objects in the scene). The function $f_3(p; z)$ corresponds to the set of internal measurements performed on the image. Finally $f_4(p; z)$ maps the measurements space into a low dimensional representation of the shape space, removing the dependence on the viewing condition z.

The Shepard's second-order isomorphism is an alternative conceptualization of shape representation, orthogonal to the one proposed by Pentland and Biederman, while the distal to proximal mapping M is the algorithmic tool that allow such isomorphism. The input to an object recognition system can be considered as a raster space $\mathcal{R} = \Re^{n^2}$ and the task of a representational system is, given a pattern $\mathcal{X} \in \mathcal{R}$, to determine the location of $\mathcal{X}$ in a proximal shape space $\mathcal{S} \in \mathcal{R}$. In accordance to this definition, in [Ede96] has been proposed an implementation of M that maps a three dimensional object in a two dimensional proximal space, see figure 1.7. This implementation uses a set of 3D models as training set (objects indicated by small icons in figure 1.7). Each model is represented as point in the proximal space, where similar models resides near each other to form a cluster (e.g. the quadrupeds). Novel models, for instance the giraffe, are positioned into the proximal space according to the cluster arisen by the models belonging to the training set, in the figure the giraffe is positioned near the other quadrupeds (the cow and the leopard) and far from the airplanes or the bipeds. This approach is an interpretation of the second-order isomorphism, in the sense that the similarity distance defined on the proximal space, is used to represent the shape of novel models.

1.1.3 Shape and Object Function

Independently by the shape representation shown in sections 1.1.1 and 1.1.2, it is interesting to consider the general setting proposed in [Mal94]. Two shapes are defined as:

- **homologous**: when their structure is similar, but not their form or function;
- **analogous**: when both structure and function are similar, but not their form;
- **isomorphic**: when structure, and form are similar but not their function.

With reference to figure 1.8, chair (c) is homologous to highchair (b) as its structure is similar but the function of a highchair is different (for children); the chair is analogous to armchair (a) as the structure

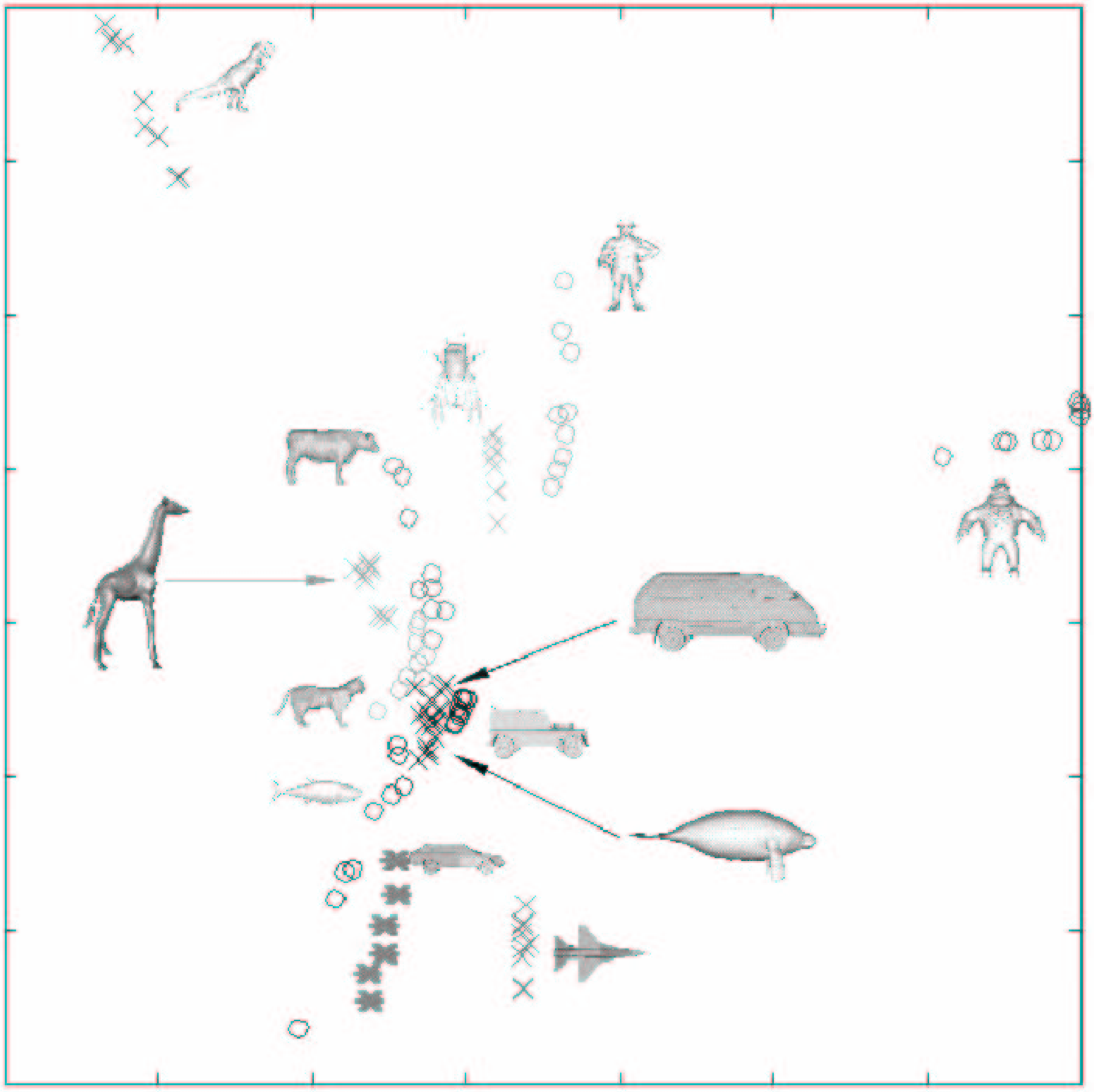


Figure 1.7: *Graphical representation of a two dimensional proximal space. Each point represents an object. The figure is from [Ede96].*

and function are similar but not the form. The last armchair (d) is not analogous and not homologous to the others since its function, form and structure are different (more analogous to a bed). It is also evident the isomorphism between the toy model of a car and the car itself. This proposes again the idea that function is a fundamental aspect of shape, or conversely, a shape may have an associated semantical meaning which goes beyond the pure limits of form. Stressing this aspect, we might say that, in specific cases, it is the function itself which defines the shape, an example being the form-

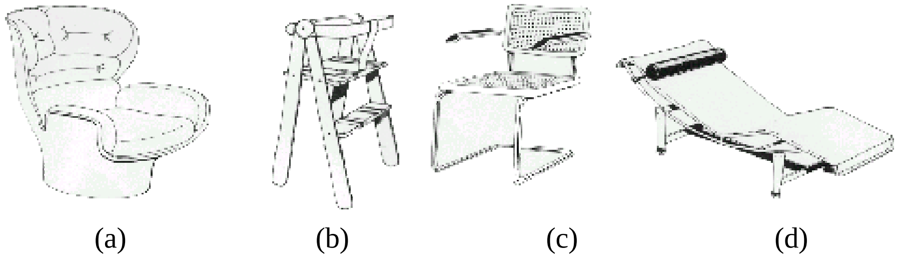


Figure 1.8: *Similarity of different seats: from structure shape and functionality.*

features in engineering and machining contexts.

1.2 Similarity Perception

Before similarity issues can be addressed, it is necessary to investigate how the notion of intuitive similarity is perceived by humans. Into the last century a lot of literature has been produced by psychologists. Such literature concerns how similarity notion behaves with respect to perceptual stimuli and human judgments. This section is a short survey of psychological theories, where the concept of similarity is formalized and discussed with respect to its properties.

Traditionally, *psychological distance* is associated with the notion of *perceived dissimilarity*, and thus dissimilarity has become an important concept in the domain of similarity analysis. Let $s(S_A, S_B)$ be the perceived similarity of stimulus S_A to S_B, and let $d(S_A, S_B)$ be the perceived dissimilarity of S_A to S_B. For now, we will assume only that similarity and dissimilarity are inversely related.

First of all a distinction between perceived dissimilarity ($d(S_A, S_B)$) and *judged dissimilarity* can be found in [AP88]. The judged dissimilarity $\delta(S_A, S_B)$ of S_A to S_B, is obtained by experiments that require subjects to make judgments of dissimilarity, generally, using a variety of response instructions[1]. The distinction between perceived and judged dissimilarity is necessary because only judged dissimilarity is accessible by experiments and it is assumed to agree ordinally with respect to perceived dissimilarity. For example, a very general model that relates perceived and judged dissimilarity assumes that:

$$\delta(S_A, S_B) = g[d(S_A, S_B)] + \epsilon_{S_A,S_B} \qquad (1.1)$$

where g is a monotonic non-decreasing function that guarantees the ordinal coherence between the pair of stimulus, and ϵ_{S_A,S_B} is a random variable whose distribution depends on the stimuli S_A and S_B. It is commonly assumed, however, that the relation between $\delta(\cdot,\cdot)$ and $d(\cdot,\cdot)$ is deterministic and

[1] The most popular response instructions method is to rate the dissimilarity of a pair of stimuli on an n-point scale, where n is usually fairly small, for instance n = 7

thus the equation (1.1) can be rewritten as:

$$\delta(S_A, S_B) = g[d(S_A, S_B)] \tag{1.2}$$

In the following only the judged dissimilarity defined in equation 1.2 will be considered.

1.2.1 Theories Based on the Geometric Models of Similarity

The most common theories represent the perceptual effects of stimuli as points in a multidimensional metric space and assumes that judgments of the perceived similarity of two stimuli are inversely related to the distance between their perceptual representations [Dav83, Kru64a, Kru64b, She62a, She62b, Tor58]. This class of models, known as the *Geometric Models of Similarity*, is contained within the larger class of *Multidimensional Scaling (MDS) Models*. MDS models assume the same sort of stimulus representation but do not necessarily require the perceptual space to be a metric.

The simplest Euclidean MDS model for a two-dimensional space assumes that:

$$d(S_A, S_B) = [(X_A - X_B)^2 + (Y_A - Y_B)^2]^{1/2} \tag{1.3}$$

where X_A, is the coordinate of stimulus S_A on dimension x of the psychological space. An important assumption of the simple Euclidean MDS model is that the psychological space is the same for all subjects.

A generalization of the equation (1.3) is introduced in [Hor69, CC70]. It is the *Weighted Euclidean Model* (also called the individual differences in orientation scaling, or INDSCAL, model) and it assumes that subjects share the same psychological space but that each individual stresses the dimensions differently. Formally, for subject j the model assumes that:

$$d_j(S_A, S_B) = [W_{x_j}^2(X_A - X_B)^2 + W_{y_j}^2(Y_A - Y_B)^2]^{1/2} \tag{1.4}$$

where $W_i(i = x, y)$ is a weight reflecting the importance that subject j places on dimension i. Note that the simple Euclidean model is a special case of the weighted model in which all $W_i = 1$. An alternative interpretation proposes the weights as measures of relative selective attention. Under this interpretation, each weight measures the degree to which an individual attends to a dimension of a perceptual space.

In both the simple and weighted Euclidean models, the dimensions are assumed to be perceived independently, but in [KT75] a violation of this assumption is discussed. The weighted Euclidean model was generalized to allow for dependencies among perceptual dimensions [Tuc72, CC72, CW74]. The idea was that the degree of perceptual dependence should be related to the angle between dimensions [AT86] and so the resulting model, known as the *General Euclidean Scaling Model*, allows oblique dimensions and defines the perceived dissimilarity of S_A to S_B for the subject j as:

$$\begin{aligned} d_j(S_A, S_B) = [&W_{x_j}^2(X_A - X_B)^2 + W_{y_j}^2(Y_A - Y_B)^2 + \\ &+ 2W_{x_j}W_{y_j}\cos\theta_j(X_A - X_B)(Y_A - Y_B)]^{1/2} \end{aligned} \tag{1.5}$$

where θ_j is the angle between dimensions x and y. When the dimensions are orthogonal, $\cos\theta_j = 0$, and the last term of equation (1.5) drops out. Thus the general Euclidean scaling model contains the weighted Euclidean model of equation (1.4) as a special case.

1.2.2 Theories Based on the Violations of the Distance Axioms

The geometric nature of the dissimilarity notion, described in section 1.2.1, implies the satisfaction of one or more of the following distance axioms, for all the stimuli S_A, S_B and S_C:

$$
\begin{aligned}
d(S_A, S_A) &= d(S_B, S_B) &(1.6)\\
d(S_A, S_B) &\geq d(S_A, S_A) &(1.7)\\
d(S_A, S_B) &= d(S_B, S_A) &(1.8)\\
d(S_A, S_C) &\geq d(S_A, S_B) + d(S_B, S_C) &(1.9)
\end{aligned}
$$

All these axioms arise from the Euclidean approximation of the perceptual space. Some psychologists assumed that this approximation is too strong for human perception of similarity and they empirically questioned the validity of the axioms. Note that the first three axioms are potentially testable because the judged dissimilarity $\delta(\cdot,\cdot)$ is monotonically related to the perceived dissimilarity $d(\cdot,\cdot)$ as described in equation (1.2). More formally let D be the class of monotonically increasing functions from $\Re$ to $\Re$. A logic predicate P over the distance functions d is an *ordinal property* if, for all $g \in D$ the following implication holds:

$$Pd \Rightarrow P(g \circ d) \qquad (1.10)$$

Ordinal properties can therefore be empirically tested by human subjects. For example, symmetry (1.8) is an ordinal property, but the triangle inequality (1.9) is not, as we will see in the following.

Although there may be problems making subjects understand the concept of self-dissimilarity, the axiom (1.6) implies that $\delta(S_A, S_A) = \delta(S_B, S_B)$ for all S_A and S_B. In [Kru78] an empirical evidence against this assumption has been shown. In particular, it has been argued that distinctive or unique stimuli, that is, stimuli having few features in common with other objects in the stimulus domain, have a greater perceived self-similarity and so a smaller perceived self-dissimilarity.

The minimality axiom (1.7) states that two different stimuli are always at least as dissimilar as either stimulus is to itself. This axiom is also potentially testable because it implies $\delta(S_A, S_B) \geq \delta(S_A, S_A)$ for all S_A and S_B. Although this appears to be a weak assumption, in [Tve77], is shown that it may sometimes be inappropriate.

The third axiom (1.8) states that similarity is a symmetric relation and therefore that $\delta(S_A, S_B) = \delta(S_B, S_A)$. A number of investigators have attacked this assumption [Tve77, TG78, Kru78]. In [TG78] Tversky and Gati gave the example that the similarity of North Korea to Red China is judged to be greater than the similarity of Red China to North Korea. The validity of this assumption may depend on the experimenter's assumptions, actually, similarity judgments can be regarded as extensions of similarity statement of the form ”a is like b”. Such a statement is directional; it involve a subject,

a, and a referent, b, and it is not equivalent in general to the converse similarity statement ”b is like a”. In fact the choice of a subject and a referent depends, in part at least, on the relative salience of the objects.

A final important assumption made by geometric similarity models is the triangle inequality axiom (1.9). Empirical testing of this axiom is problematic when perceived and judged dissimilarity are only monotonically related. In this case the fact that the perceived dissimilarities satisfy (or violate) the triangle inequality places no logical constraints on the judged dissimilarities. Even a linear relation between perceived and judged dissimilarity is not enough. For example, if the perceived dissimilarities violate the triangle inequality and

$$\delta(S_A, S_B) = \alpha d(S_A, S_B) + \beta \tag{1.11}$$

for some constants α and β (where α is positive), then it is always possible to find values of β for which the judged dissimilarities satisfy the triangle inequality. In spite of these difficulties, it is widely suspected that perceived dissimilarity may sometimes violate the triangle inequality [Tve77, TG78]. More than a century ago, William James [Jam90] gave an example of what seems a clear violation. A flame is similar to the moon because they both appear luminous, and the moon is similar to a ball because they are both round. However, in contradiction to the triangle inequality, a flame and a ball are very dissimilar.

Because of their questionable empirical validity, it is desirable to investigate theories of perceived similarity not constrained by the distance axioms. Although the simple and the weighted Euclidean MDS models, outlined in section 1.2.1, are based on true distance metrics, the weighted Euclidean model can account for violations of the triangle inequality. For example, when judging the similarity of a flame and the moon, attention is focused on a luminosity dimension, but when judging the similarity of the moon to a ball, attention is switched to a shape dimension. On the other hand, the dissimilarity measure associated with the general Euclidean scaling model is not a true distance metric because it is not constrained by the triangle inequality. Thus unequal self-dissimilarities, or violations of minimality or symmetry, falsify the general Euclidean scaling model (and therefore also the simple and weighted Euclidean models), but violations of the triangle inequality do not.

A modification of the standard geometric similarity model that can account for violations of some distance axioms has been proposed in [Kru78]. The idea was that pairwise similarity should depend not only on the distance between the psychological representations of the two stimuli but also on the spatial density of stimulus representations in the surrounding psychological space. Let $\Phi(S_A, S_B)$ be the distance between the perceptual representations of S_A and S_B, and let $h(S_i)$ be a measure of the spatial density around the representation of stimulus S_i. Thus $h(S_i)$ is greater in ensembles with many stimuli similar to S_i than in ensembles with few stimuli similar to S_i. The perceived dissimilarity measure, $d(S_A, S_B)$, in the distance-density model is defined as:

$$d(S_A, S_B) = \Phi(S_A, S_B) + \alpha h(S_A) + \beta h(S_B) \tag{1.12}$$

where α and β are nonnegative weighting constants. Because the spatial density around S_A may be different from the density around S_B ($h(S_A) \neq h(S_B)$), the distance-density model can account for differences in self-similarity. By allowing $\alpha \neq \beta$, it can also predict violations of symmetry. On the other hand, the model cannot account for violations of the triangle inequality, no matter what the values of α and β (so long as they are nonnegative).

A powerful alternative to MDS models is the *feature-contrast model*, proposed by Amos Tversky in [Tve77, TG78]. In this approach, stimuli are characterized as sets of features, and similarity is based on a feature-matching function that weights common and distinct features of the pair of stimuli. Specifically, Tversky assumed that the perceived similarity of SA to SB is given by:

$$d(S_A, S_B) = \alpha f(A - B) + \beta f(B - A) - \Theta f(A \cap B) \tag{1.13}$$

where A and B are the set of features associated to the stimuli S_A and S_B, Θ, α and β are nonnegative free parameters, and the nonnegative function f is a measure of the salience of a set of features. Thus $f(A \cap B)$ is the salience of the features that A and B have in common and $f(A - B)$ is the salience of features that are contained in A but not in B. In general, we expect the feature-contrast model to predict an increase in similarity with the number of features a pair of stimuli have in common and a decrease with the number of distinct features. However, the model is exceedingly general. Feature salience is an elusive term that may be only weakly related to the number of relevant features. Similarly, the process of identifying features may be problematic. For example in [TG82] the term features has been used ”to describe any property, characteristic, or aspect of objects that are relevant to the task under study”. This kind of catch-all definition makes the task empirically falsifying the contrast model very difficult. More formally the theory of Tversky is based on the following assumptions:

$$\begin{aligned} d(S_A, S_B) &= F(A - B, B - A, A \cap B) && (1.14)\\ d(S_A, S_B) &< d(S_A, S_C) \quad \text{whenever} \\ & \qquad A \cap C \subseteq A \cap B,\ A - B \subseteq A - C,\ B - A \subseteq C - A && (1.15) \end{aligned}$$

A function d, that satisfies the matching (1.14) and monotonicity (1.15) properties is called a matching-function. Let the expression $F(X, Y, Z)$ be defined whenever there are A and B such that X = A ∩ B, Y = A - B and Z = B - A. Define V ≈ W if there exist X, Y, Z such that one or more of the following holds:

$$\begin{aligned} F(V, Y, Z) &= F(W, Y, Z) \\ F(X, V, Z) &= F(X, W, Z) \\ F(X, Y, V) &= F(X, Y, W) \end{aligned} \tag{1.16}$$

The pairs of stimuli (S_A,S_B) and (S_C, S_D) are said to agree on one (two, three) components whenever one (resp. two, three) of the following hold:

$$\begin{aligned} (A \cap B) &\approx (C \cap D) \\ (A - B) &\approx (C - D) \\ (B - A) &\approx (D - C) \end{aligned} \tag{1.17}$$

based on these definition, Tversky postulates a third property of the dissimilarity measure called independence. Supposing that the pairs of stimuli (S_A, S_B) and (S_C, S_D) as well as (S_A’, S_B’) and (S_C’,

S_D'), agree on the same two components while the pairs (S_A, S_B) and (S_A', S_B'), as well as (S_C, S_D) and (S_C', S_D') agree on the remaining (third) component, then:

$$d(S_A, S_B) \leq d(S_A', S_B') \Longleftrightarrow d(S_C, S_D) \leq d(S_C', S_D') \tag{1.18}$$

Figure 1.9 shows an example of the independence property. Each stylized face is characterized by three features: face profile, eyebrow and mouth. In this case, the independence property states that if (a,b) are ”closer” than (c,d), then (a’,b’) are ”closer” than (c’,d’). Let d be a similarity function for which matching (1.14), monotonicity (1.15) and independence (1.18) hold. Then, there are a dissimilarity function D and a nonnegative function f and three constants Θ, α and β such that, for all stimuli S_A, S_B, S_C and S_D:

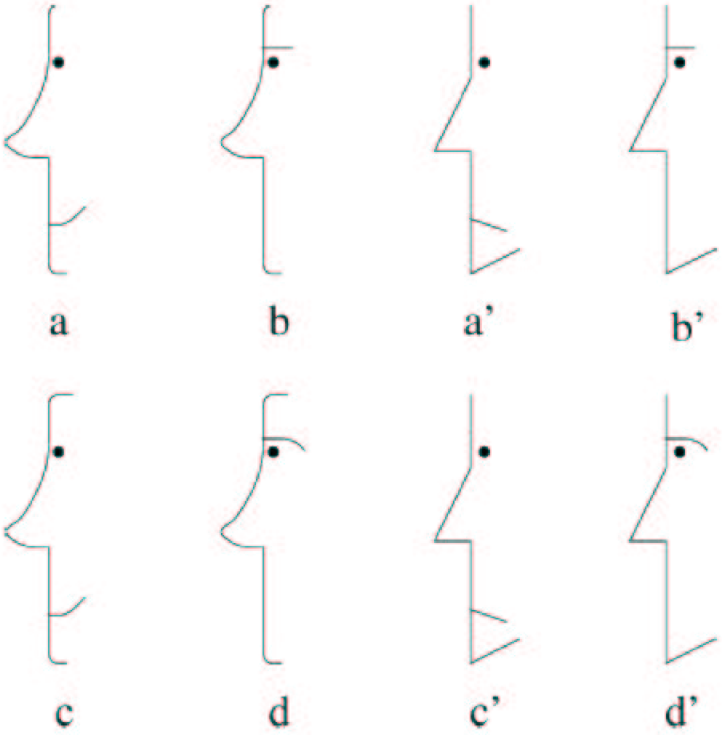


Figure 1.9: *An example of independence. If a and b are considered more similar than a’ and b’, then c and d will appear more similar than c’ and d’. The picture is from [SJ99]*

$$\begin{aligned} D(S_A, S_B) &\leq D(S_C, S_D) \Longleftrightarrow d(S_A, S_B) \leq d(S_C, S_D) \\ D(S_A, S_B) &= \alpha f(A - B) + \beta f(B - A) - \Theta f(A \cap B) \end{aligned} \tag{1.19}$$

This result implies that any dissimilarity ordering that satisfies matching, monotonicity, and independence can be obtained using a linear combination (contrast) of a function of the common features ($A \cap B$) and of the distinctive features ($A - B$) and ($B - A$).

Moreover, in the case in which stimulus features can be identified and experimentally manipulated, Tversky and Gati [TG82] identified three ordinal properties, used to replace the metric axioms, that characterize what they called a *monotone proximity structure*. Let $d(ap, bq)$ be the perceived dissimilarity between a pair of stimuli ap and bq that differ on two stimulus features or components, where the first stimulus has a value a on the first feature and value p on the second, and the second stimulus has value b on the first feature and value q on the second. The first property of a monotone proximity structure is *dominance,* which states:

$$d(ap, bq) \geq max[d(ap, aq), d(aq, bq)] \tag{1.20}$$

for all values a, b, p, and q. In other words, the two-dimensional dissimilarity of a pair of stimuli exceeds both one-dimensional dissimilarities.

The second property is called *consistency* and states that:

$$\begin{array}{l} d(ap, bp) > d(cp, dp) \quad iff \quad d(aq, bq) > d(cq, dq) \\ \text{and} \\ d(ap, aq) > d(ar, as) \quad iff \quad d(bp, bq) > d(br, bs) \end{array} \tag{1.21}$$

for all values a, b, c, d, p, q, r, and s. In other words, the ordinal dissimilarity relation of two pairs of stimuli differing on one dimension does not depend on the level of the other fixed dimension.

The third property characterizing a monotone proximity structure involves an ordering relation on each dimension. If

$$d(ap, cp) > max[d(ap, bp), d(bp, cp)]$$

then b is said to be between a and c, and we write $a|b|c$. The property states that this form of ordering satisfies *transitivity,* that is

$$\text{if } a|b|c \text{ and } b|c|d \text{ then } a|b|d \text{ and } a|c|d \tag{1.22}$$

A similar condition can be derived for the second dimension.

Monotone proximity structures are considerably more general than geometric models of similarity in the sense that geometric models predict dominance, consistency, and transitivity to be true, but not all monotone proximity structures predict the distance axioms to hold [TG82]. Even so, not all MDS models are monotone proximity structures, because the general Euclidean model can account for violations of dominance. On the other hand, it is not difficult to show that the feature-contrast model predicts consistency and transitivity, and if feature saliency is an increasing function of the number of features, so that for example, $f(ap) > f(p)$, then it also predicts dominance. Therefore, a large class of feature-contrast models are monotone proximity structures, and so empirical evidence of violations of dominance, consistency, or transitivity would present serious difficulties for the feature-contrast theory. Dominance is a weak form of the triangle inequality that applies along the coordinate axes, a graphical representation of this property is shown in figure 1.10(a). The property holds if the distance between (a, p) and (b, q) is grater than the distance between (a, p) and (a, q) or the distance between (a, q) and (a, b). Consistency ensures that certain ordinal properties related to the ordering of the features x do not change when y is changed. As shown in figure 1.10(b), if the distance between

(a, p) and $(b.p)$ is greater than the distance between (c, p) and (d, p) then the distance between (a, q) and (b, q) is greater than distance between (c, q) and (d, q) and vice versa. The same is for the features (a, p) and (a, q), and (a, r) and (a, s) as well as (b, p) and (b, q), and (b, r) and (b, s). Transitivity ensures that the ”in between” relation behaves as in the metric model, at least when moving along the axes of the feature space. In figure 1.10(c) is shown that if (b, p) is between (a, p) and (c, p), and that if (c, p) is between (b, p) and (d, p), then both (b, p) and (b, p) are between (a, p) and (d, p). Note that, in the Euclidean model which is isotropic every property holds (or does not hold) for a series of collinear points irrespective of the direction of the line that joins them. In measuring the perceptual distance, the directions of the feature axes have a special status. Moreover note that most of the distance measures proposed in the literature, as well as the feature contrast model, predict that dominance consistency and transitivity hold.

Finally, one other axiom has played an important role in discriminating among alternative theories of similarity. For experiments with stimuli composed of several separate components, in [TG82] has been proposed an ordinal axiom, called the *corner inequality*, that captures the spirit of the triangle inequality. If $a|b|c$ and $p|q|r$, then the corner inequality holds if:

$$\begin{array}{l} d(ap, cp) > d(ap, dq) \quad and \quad d(cp, cr) > d(bq, cr) \\ \text{or if} \\ d(ap, cp) > d(bq, cr) \quad and \quad d(cp, cr) > d(bq, cr) \end{array} \tag{1.23}$$

In other words, the corner inequality holds if both one-dimensional dissimilarities exceed the two-dimensional dissimilarities. In figure 1.10(d) a graphical representation of the corner inequality is shown. Tversky and Gati [TG82] derived this property, not to test the feature-contrast model, which can predict the property but is not constrained to do so, but to test geometric similarity models. They showed that a large and popular class of geometric models (i.e., those possessing Minkowski distance metrics) predict the corner inequality to be true and they presented compelling evidence that under certain stimulus conditions, the axiom fails dramatically.

1.3 Similarity Measures

Although in section 1.2 has been shown that the violation of the distance axioms are statistically significant and experimentally reliable, in the sense that they were observed with different stimuli under different experimental conditions, the effects are relatively small. Consequently, an Euclidean map may provide a very useful description of complex data, even though its underlying assumptions (e .g., symmetry, or the triangle inequality) may be incorrect [TG78]. By this point of view, a distance function having the properties shown by the equations 1.6, 1.7, 1.8 and 1.9, provide a good first approximations for the similarity estimation. The way of thinking of Tversky and Gati, is well summarized by the following analogy: the knowledge that the earth is round does not prevent surveyors from using plane geometry to calculate small distances on the surface of the earth. The fact that such measurements often provide excellent approximations to the data, however, should not be taken as evidence for the flat-earth model.

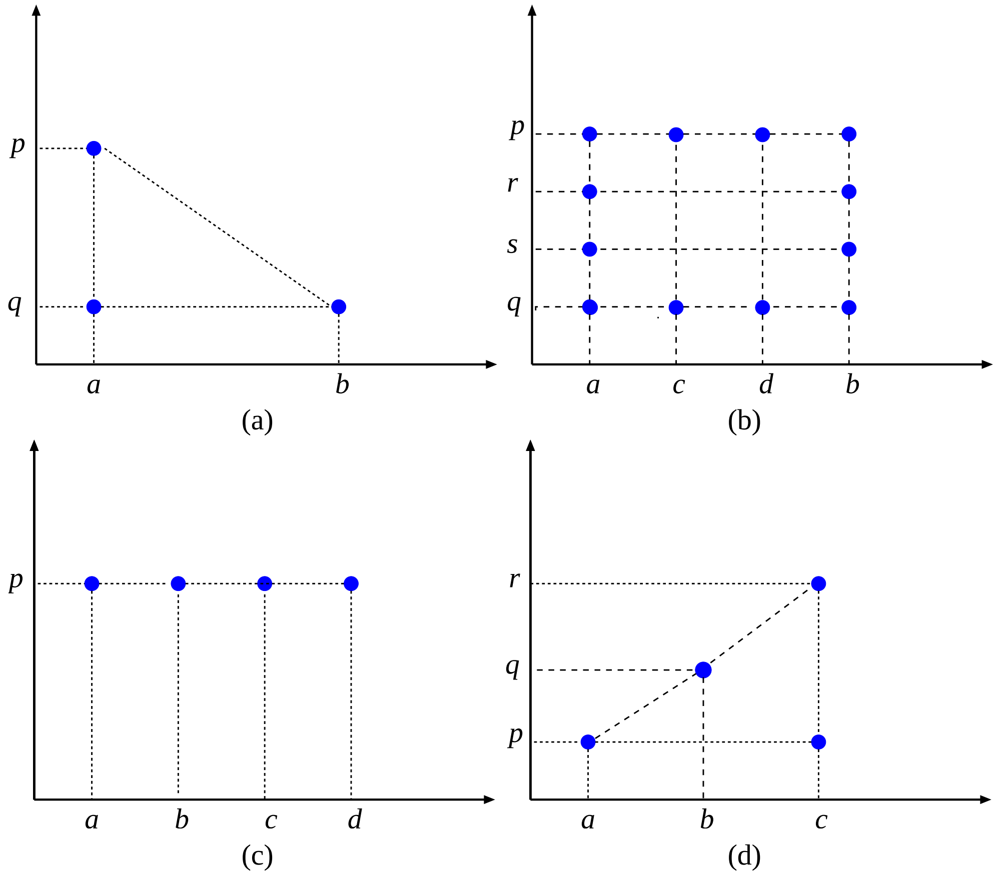


Figure 1.10: *Graphical representation of the monotone proximity structure properties: dominance (a), consistency (b), and transitivity (c). Finally the corner inequality property (d).*

In this section, the properties characterizing the distance function d will be formalized and described with the aim to introduce the concept of distance commonly used in the computer graphic community. Formally, a similarity measure d on a set S is a *n*onnegative valued function $d : S \times S \to \Re$ characterized by the properties given below [Hag00]. A subset of these properties leads to the de notion of semi-metrics, pseudo-metrics, metrics and ultra-metrics.

The self-identity property states that the distance between identical objects is zero:

$$d(x, x) = 0, \text{for all } x \in S \text{ (self-identity)} \tag{1.24}$$

Positivity is the property which states that distinct objects have a nonzero distance:

$$d(x, y) > 0, \text{for all } x \neq y \in S \text{ (positivity)} \tag{1.25}$$

Simmetry states that the order of two elements does not matter for the distance between them:

$$d(x, y) = d(y, x), \text{for all } x, y \in S \text{ (simmetry)} \tag{1.26}$$

The triangle inequality states that the distance between x and z does not exceed the sum of the distance between x and y and the distance between y and z:

$$d(x,z) \leq d(x,y) + d(y,z), \text{for all } x,y,z \in S \text{ (triangle inequality)} \tag{1.27}$$

A relaxed triangle inequality, has been introduced in [FS98a]. For all $x, y, z \in S$, it is formalized as:

$$d(x,z) \leq \alpha(d(x,y) + d(y,z)), \text{ where } \alpha > 1 \text{ (relaxed triangle inequality)} \tag{1.28}$$

while a stronger version of the triangle inequality is the following. For all $x, y, z \in S$:

$$d(x,z) \leq max\{d(x,y), d(y,z)\} \text{ (strong triangle inequality)} \tag{1.29}$$

The utility of these properties depends on the application context on which the distance function is used for. By the other hand, different distance functions, that will be discussed in the following chapters, satisfies only a subset of the previous properties. The properties are usually grouped according to the following definitions [VT03]:

Definition 1.3.1 (Dissimilarity Properties) *The properties previously described can be grouped as in the following:*

- **semi-metric***: a distance d that satisfies the properties 1.24, 1.25 and 1.26;*
- **pseudo-metric***: satisfies the properties 1.24, 1.26 and 1.27;*
- **metric***: a pseudometric that satisfies the property 1.25;*
- **ultra-metric***: a metric satisfying property 1.29.*

A trivial example of a metric is the discrete metric, denoted by c. For an arbitrary set S, this metric is defined as $c(x,x) = 0$ and $c(x,y) = 1$ for $x \neq y$ and $x, y \in S$. The discrete metric is also an example of ultra-metric [Hag00]. An example of pseudometric but not a metric is the distance between the first coordinates of two points, $d(x,y) = |x_1 - y_1|$ on the Euclidean plane [Hag00].

The triangle inequality is very useful for making searching more efficient as shown in [BFM$^+$96]. This is based on the following observation. Consider a database A and let A_i be an element of A. Consider also a query A_q. The problem is to find all the database elements A_i such that $d(A_i, A_q) \leq \delta$. Let $A_r \in A$ be some reference element and suppose that $d(A_r, A_i)$ has been computed, off-line, for each $A_i \in A$. If the triangle inequality holds, $d(A_r, A_q) \leq d(A_r, A_i) + d(A_i, A_q)$, then the elements A_i such that $d(A_i, A_q) \leq \delta$ are the ones satisfying $d(A_r, A_q) - d(A_r, A_i) \leq \delta$. The distance between a database element A_i and a query A_q can be approximated by comparing their distances to a reference shape A_r instead of comparing the query A_q with all the database elements $A_i \in A$. Of course if the database has many elements, only one reference shape A_r is not sufficient, thus in [VV99, VV02] the use of more than one reference elements has been proposed.

Depending on the application context the distance function d, other than the previous properties, should be robust with respect to transformations, deformations and noise [HV99], moreover it should satisfy the proportionality and monotonicity properties [VH00].

In order to describe these properties, topological sapces have to be considered. Let X be a topological space, $\mathcal{P}$ a collection of subset of X and d a metric on $\mathcal{P}$. The structure $(X, \mathcal{P}, d)$ is called a *metric pattern space* and the elements of $\mathcal{P}$ are pattenrs. Let $Hom(X)$ be the class of all the homeomorphism on X. A collection of pattern $\mathcal{P}$ uniquely determines a maximal subgroup T of $Hom(X)$ under wich $\mathcal{P}$ is closed. The trasformatrion group T consists of all $t \in Hom(X)$ such that both $t(A)$ and the inverse image $t^{-1}(A)$ are members of $\mathcal{P}$, for all $A \in \mathcal{P}$. The metric pattern space $(X, \mathcal{P}, d)$ is *invariant for a transformation* $g \in T$ if:

$$d(g(A), g(B)) = d(A, B), \text{ for all } A, B \in \mathcal{P} \text{ (transformation invariance)} \tag{1.30}$$

The *invariance group* G of a metric pattern space is the set of all the transformations in T for wich the space is invariant. In figure 1.11 an example of affine trasformation is shown. The two pattern A and B has been trasformed in $g(A)$ and $g(b)$ respectively. Invariance for affine trasformation makes the distance function independent from the choice of the coordinate system.

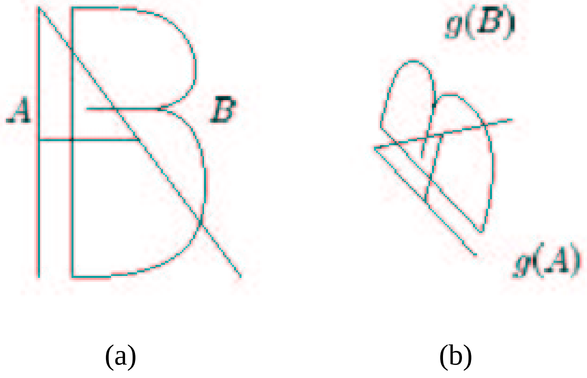


Figure 1.11: *Two patterns (a) and their affine transformation (b). The figure is from [HV99].*

In order to formalize the robustness with respect to small deformations, the group of transformations T should be considered a topological space where transformations are points, and each transformation has an open neighbourhoods that is a set of transformations [HV99]. This is because choosing a transformation $t \in T$ sufficiently close to the identity, makes the deformation arbitrarily small and in a topological space, such closeness, can be described by considering increasingly smaller open neighbourhood of a point. This topology space is called compact-open topology and it is determined by its collection of open sets. A collection of subsets $\mathcal{S}$ of a space S whose union equals S, is called a subbasis for a topology on a space S. The topology generated from a subbasis $\mathcal{S}$ consists of all unions of finite intersections of elements of $\mathcal{S}$. The compact-open topology on the set of transformations T is generated by the subbasis consisting of all sets of the form $S(K,U) = \{t \in T | t(K) \subseteq U\}$, where $K \subseteq X$ is compact, and $U \subseteq X$ is open. The metric pattern space $(X, \mathcal{P}, d)$ is called *deformation robust* if for each $A \in \mathcal{P}$ and $\varepsilon > 0$, an open neighbourhood $I \subseteq T$ of the identity exists such that:

$$d(A, t(A)) < \varepsilon, \text{ for all } t \in I \text{ (deformation invariance)} \tag{1.31}$$

Figure 1.12 shows the image of a pattern under a transformation t, contained in an open neighbourhood of the identity. This neighbourhood is a finite intersection of subbasis element $S(K_i, U_i)$ generated by compact segments K_i of A and open balls U_i containing A_i.

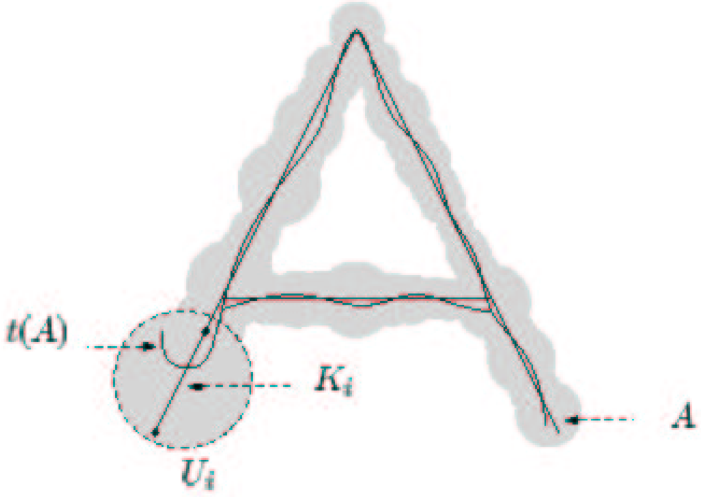


Figure 1.12: *The segment K_i of the image has been deformed through the transformation t. The figure is from [HV99].*

The *noise invariance* property is described by the equation 1.32. It states that changes in patterns do not cause discontinuity in patterns distance, if the changes happen within small regions [HV99]. For each $A \in \mathcal{P}$, $x \in X$ and $\varepsilon > 0$, an open neighbourhood U of x such that $B - U = A - U$ implies:

$$d(A, B) < \varepsilon, \text{ for all } B \in \mathcal{P} \text{ (noise invariance)} \tag{1.32}$$

This property states that the distance between A and B can be made smaller by making U smaller. In figure 1.13 a pattern is shown and a noise on a point x is simulated. The addition of noise $B - A$ within a neighbourhood U of x results in a new pattern B.

Another important property that a distance function should satisfy is the *proportionality*. It states that the change from A to $A \cup B$ is smaller than the change from A to $A \cup C$ if B is smaller than C. For all $A \cap B = \phi$ and $A \cap C = \phi$, if $B \subset C$, then:

$$d(A, A \cup B) < d(A, A \cup C) \text{ (proportionality)} \tag{1.33}$$

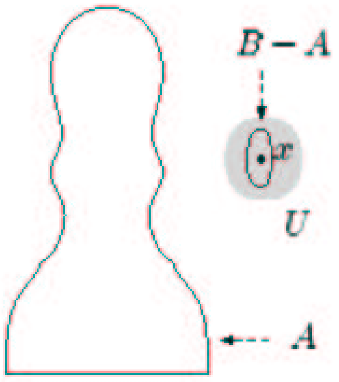


Figure 1.13: *The noise $B - A$ is applied to the pattern A. The figure is from [HV99].*

Finally the distance function is strictly monotone if at least one of the intermediate steps of adding $B - A$ to A, and $C - B$ to B is small than the two steps combined. For all $A \subset B \subset C$:

$$d(A, C) > d(A, B) \text{ or } d(A, C) > d(B, C)(\text{monotonicity}) \tag{1.34}$$

1.4 Shape Descriptors Characterization

Most of the methods used by the computer graphic community, for the estimation of the similarity between three dimensional shapes, compare descriptors of the shape objects instead of the whole objects themselves. There are many reasons for this, and there are also a lot of kind of shape descriptors. Depending on the descriptor, the amount of memory needed to store it, is much less than the one needed to store the whole model. This implies, expecially for applications dealing with shape retrieval tasks involving several thousands of models, a sensible enanchment of the computational performance in terms of temporal complexity. Another advantage of descriptors, is their capacity to capture important features of the shape and to discard the irrelevant ones. This property maximize the information, minimizing the memory needed to store the information, and sometimes, it allow different kinds of reasoning on the shape.

Before to enumerate the properties that a shape descriptor should satify, a distinction have to be highlighted between *representation* and *description* of a shape. As observed by Nackman et al. in [Nac84], an object representation contains enough information from which to reconstruct (an approximation of) the object, while a description only contains enough information to identify an object as a member of some class of objects. In this sense, we might say that a model, in accordance with the definitions given so far, is a representation which is quantitatively and qualitatively similar to an object, while a description is only qualitatively similar. The representation of an object is more detailed and accurate than a description, but it does not necessarily contain in an explicit manner any high-level information on the shape of the object. The description is more concise and communicates an elaborate and composite view of the object class. In the context of terrain modeling, for instance, representations might provide answers to questions such as ”What is at that specific location ?”, while from a description one might get answers to ”Is there a ridge in that area ?”. Being more abstract and elaborate, the description has to be based on shape ”units”, which synthesize, or group, sets of lower-level entities of the model into classes of elements which can be named with a descriptive label. Thinking of the two levels of description and representation, we can immediately say that both levels are used and quickly interchanged also in everyday situations. It is therefore desirable for the computer modeling process to follow the same approach.

Shape descriptors can be both representations or descriptions of the object shape, depending on the information they store. In this sense, a shape descriptor may contain the same information as the object model, but arranged in a way that allows shape reasoning (it represents the object shape), or just the salient information relevant for a specific task (it describes the object shape). In order to understand and make explicit the differences among them, some characteristic properties that shape descriptors should have are enumerated:

Saliency: it is the property of a descriptor to capture the shape features that characterize the object. These characteristic features depend on the geometry, topology and structure of the object. For instance, the salient geometric, structural and topological features of the teapot shown in figure 1.14 should be the handle, the spout, the tip over the teapot cover and the round shaped body. Saliency grants that the descriptor capture all the necessary information to represent or describe the shape of the object.

Relevance: some features are relevant in the context of use of the object, while other features are not. For example, consider an application context where a virtual human have to deduce, from the descriptor, how to grasp the teapot shown in figure 1.14. In this case the relaevant feature is the handle of the teapot, independently by the roundness of the body. Relevance assures that all the information usefull in a specific context are captured by the descriptor and can be used for shape reasoning depending on the application context. Relevant features are a subset of the salient features.

Concisness: is the capability to minimize the memory needed to store the descriptor, maximizing the amount of information represented. For example the teapot model shown in figure 1.14(b) is represented by a mesh of seventeen thousand of triangles, but a shape descriptor should encode the information needed to represent it taking up only few byte. In this case the amount of memory needed to store the descriptor, should be much less than the one needed to store the whole model.

Unicity: Choosen a theoretical methodology to extract the descriptor, choosen an algorithm and finally choosen the parameters needed to run the algorithm, the shape descriptor have to be unique. In some cases this does not happen: some descriptors are obtained through a stochastic process, for example based on a random sample of the surface points of the object model and thus producing different descriptors of the same model depending o the sample. The more the number of samples approximate the number of elements that constitutes the model, the more the difference between two different descriptors decreases. This implies that differences can be reduced only producing space consuming descriptors, thus compromising the cnciseness.

Invariance to the rigid transformations: this is the property of a descriptor to be invariant with respect to rotations, translations and scaling of the coordinate system of the model object. Rigid transformation invariance is usefull in contexts aimed at analyzing the shape independently by its immersion into the space(coordinate system). While, in other contexts it is important to discrimante among the relative position and/or scale among objects. For instance, a scene where a bottle is laying on the table rather than standing on it.

Invariance to the model representation: in some cases theoretical methodologies for the extraction of the shape descriptor impose constraints on the model representation of the object (e.g. polygon mesh, algebraic surfaces, volumetric representations). In some other cases, given the representatoin, the constraints focuses on topological properties as 2-manifoldness (e.g. polygon soups or connected and not self-intersecting polygonal mesh), orientability or genus and number of the object shels. Another interpretation of the invariance with respect to representation, also, means robustness with respect to change in the model resolution, e.g. for a polygon mesh, a slight reduction of the number of the mesh vertexes should not cause a relevant change into the shape descriptor. The triangle mesh shown in figure 1.14(b) has a number of vertexes one hundred seventy times bigger than the triangle mesh shown in figure 1.14(c). A shape descriptor is invariant with respect to the model representation if it is able to capture the salient and relevant features of the model in spite of this difference.

Scalability: it is the capability to select the relevant information contained in the descriptor and to discard the non relvant one. Some shape descriptors allow a multi-reslolution representation/description of the object. In this case the relevant information stored in the descriptor can be selected through a process based on geometric reasoning. For example, an high resolution representation/description can be usefull by comparing objects having small geometric features, on the other hand low resolution can be used to discard small features, but improving the time consuming performance through a rough comparison of objects. Another interpretation of scalability deals with multi-scale representatoin/description of objects. It is the capability to reduce the size of the descriptor selecting relevant features and neglecting the others. For example, handles and protrusions can be processed independently by their geometric characterization. In this case, retrieval of objects that can be grasped by virtual humans can be improved by selecting the information that describe the handles of the object and by discarding the other information.

Meaning of the informatoin stored: salient and relevant features captured by the descriptor can be used in several application domains. Each of the following tasks need different information in order to be accomplished:

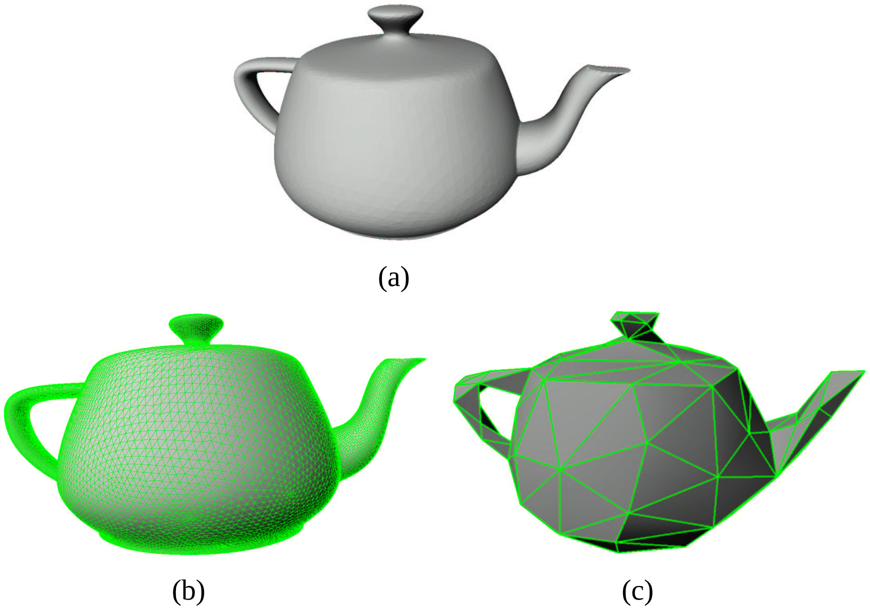
- *shape retrieval*: a huge number of models belonging to a databese need to be repesented/described by concise descriptors in order to perform fast queries
- *shape recognition/classification*: given a model and a scene made of several objects, the recognition task consits in finding the model in the scene. Concerning the classification task a model have to be related with a specific class of objects. In both the cases, both relevant and salient features have to be stored.
- *shape reconstruction*: the descriptor has to provide the information usefull to an approximate reconstruction of the object, it has to represent the shape of the obejct.
- *shape composition/editing*: in this case salient features have to be described/represented in order to drive the editing of the object.
- *animation*: many applications dealing with animation need to reason on relevant features in order to be able to deform them.

Into the chapters 2 and 3, the most used shape descriptors will be shown and each of them will be discussed with respect the previous properties and to the similarity measures used for the comparison.

1.5 Discussion

In this chapter, the literature on shape representations, similarity theories addressed by psychologists and similarity measures commonly used in the computer graphics community have been surveyed in order to provide a wide view of the problem statement concerning the similarity computation. Finally some properties of the shape descriptors has been enumerated.

The two methodologies proposed in section 1.1.1 and 1.1.2 are really different way to represent shape. The superquadrics proposed by Pentland and the geons proposed by Biederman are shape representations in the sense described in section 1.4. They provide the information needed to approximatively reconstruct the object they represent, moreover the geons approach, that subdivide the object into parts and explicit adjacency relations among them, leads to the notion of *structural description of the object*. These methods, have been used mainly for object recognition, classification and retrieval tasks, but due to their capability to syntesize salient and relevant features (figure 1.4(b)), they should be suitable for approximation, authomatic or semi-authomatic editing and finally animations. The



(a)

(b)

(c)

Figure 1.14: *A 2-manifold model of a teapot (a). The same model represented as triangle mesh with more than seventeen thousand vertexes (b) and only one hundred vertexes (c). In the three models the salient and relevant features are the same.*

second order isomorphism representation of shape propsed by Sheppard, does not focus on the explicit representation/description of the object or of its subparts. The method aims at representing the similarity relations among an object with respect to the other objects of the ”universe”. Also in this case, this methodology has been used for object recognition, classification and retrieval, but differntly from the geons and superquadrics approachs, the salients and relevants features are not necessarily represented. This chracteristic makes the second order isomorphism representation not suitable for animaitons, editing and reconstruction application tasks.

Independently on the shape representation, psychologists studied how humans perform similarity (dissimilarity) judgements on perceptual stimuli. The experiments executed on shapes, but also on more general stimuli, pointed out a strange behaviour at first glance. The dissimilarity judgement between the stimuli A and B might be different, in some contexts, from the one between B and A, moreover the dissimilarity between the stimulus A and itself might be different from the dissimilarity between B and itself, depending on the features that characterize the two stimuli. Another important property, that by the computational point of view should be staisfied by the dissimilarity function, is the triangle inequality. Unfortunatelly this property, as explained in section 1.2.2, can not be investigated by empirical experiments, because the monotonic relationship between judged and percepted dissimilarity. This led the psychologists to formalize theories able to capture the behaviour of judged similarity, and not satisfying the properties characterizing the Euclidean properties of the distance function. In spite of the empirical results on judged dissimilarity, only few cases using dissimilarity functions based on not Euclidean axioms are prsent into the literature dealing with the computer graphic community. This is because psychologists agree on the use of a distance function based on the Euclidean axioms, in order to approximate the intuitive notion of dissimlarity.

For this reason, section 1.3 provides a short description of the properties that a distance function should adopt in the application contexts dealing with shape similarity. The properties has been grouped according to the notion of semi-metric, pseudo-metric, metric and ultra-metric. Other than the Euclidean properties, a distance function should also satisfy properties dealing with invariance with respect ot affine transformation and robustness with respect to deformations and noise. All these properties, together with the properties highlighted, in section 1.4, makes the shape descriptor suitable for specific application tasks.

Chapter 2

Evaluation of Shape Similarity Based on Geometric Shape Descriptors

The notion of shape descriptor given in section 1.4 is the amount of information needed to produce an approximation of the object or to identify the object as member of a specific class. This chapter will show the most used shape descriptors obtained analyzing the shape of the object by the geometric point of view, that is involving measurements of angles, curvature, area, volume, distances, volume and normals of the object to be described. Even if this descriptors are mainly used for shape comparison among objects, they are low dimensional descriptors that do not resemble the object itself, in some cases they are point cloud sets, in some other cases vectors or matrices.

In this chapter the geometric shape descriptors are roughly splitted in two main categories: statistical and exact descriptors.

2.1 Survey on Statistical Shape Descriptors

Statistical shape descriptors are usually based on a random sampling of the object surface. The characteristics of the descriptor depend mainly on the size and the uniformity of the sampling, it can be concise or independent by the coordinate system of the object, it can capture salient or relevant shape features, but due to its statistical nature, it can not be unique.

2.1.1 Shape Distributions

The approach proposed in [OFCD01, OFCD02], describes the overall shape of a 3D object, through a shape signature representing the object spatial distribution. The signature is a statistical descriptor obtained by a function defined on a random sampling of the model surface, that measures the geometric properties of the object. The authors, propose the following five different measuring functions:

- **A3**: measures the angle between three random points on the surface of a 3D model.
- **D1**: measures the distance between a fixed point and one random point on the surface, where the fixed point is the centroid of the model.
- **D2**: measures the distance between two random points on the surface.
- **D3**: measures the square root of the area of the triangle between three random points on the surface.
- **D4**: measures the cube root of the volume of the tetrahedron between four random points on the surface.

Examples of the five functions are shown in figure 2.1. Since A3 deals with angles, it is invariant with respect to the scale of the object, but D1, D2, D3 and D4 do not. All the measuring functions are invariant with respect to coordinate system of the object.

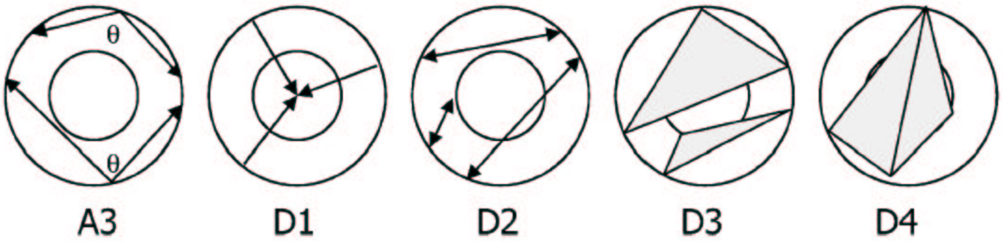


Figure 2.1: *Five functions measuring geometric characteristic of 3D objects. The figure is from [OFCD02].*

The shape descriptor construction is based on a stochastic method. Specifically, N samples from the measuring function are computed and an histogram is produced by counting how many samples fall into each of B fixed sized bins, then a piecewise linear function with V ($\leq$ B) equally spaced vertexes is obtained from the histogram. The linear function corresponding to a sequence of V integers represents the shape descriptor. The points used to evaluate the measuring function are sampled from the model surface of the object and not from the set of vertexes belonging to the mesh representing the object[1]. In this way the descriptor is independent from the resolution of the mesh. A process based on the area of the triangles belonging to the model mesh is used to obtain a uniform random sampling of the object surface. Figure 2.2 shows some examples of shape descriptors based on the measuring function D2.

The shape descriptors of simple 2D and 3D objects, like a line segment, a circumference, a triangle, a cube, a sphere, and a cylinder, are shown in figure 2.2(a-f). These examples show how the

[1] The authors experimented their shape descriptor on triangle meshes.

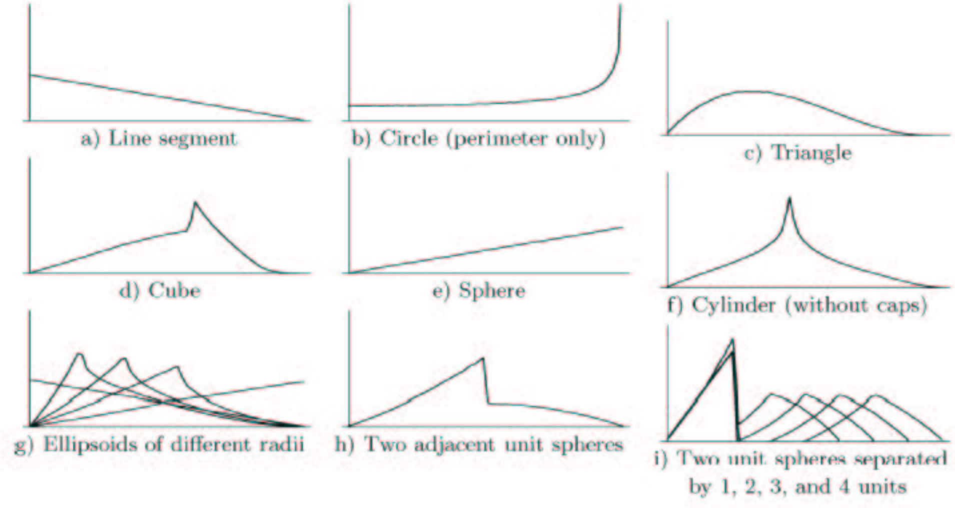


Figure 2.2: *The D2 measuring function applied to simple objects. The figure is from [OFCD02].*

distance distribution is captured and represented by the descriptor and highlight the shape discriminating capability for basic shapes. Figure 2.2(g) shows the distance distributions for ellipsoids of different semi-axis lengths (a, b, c) overlaid on the same plot. The left-most curve represents the D2 distribution for a line segment ellipsoid $(0, 0, 1)$; the right-most curve represents the D2 distribution for a sphere ellipsoid $(1, 1, 1)$; and, the remaining curves show the D2 distribution for ellipsoids in between ellipsoid $(r, r, 1)$ with $0 < r < 1$. The figure that the change from sphere to line segment is continuous. Similarly, Figure 2.2(h-i) show the D2 distributions of two unit spheres as they move 0, 1, 2, 3, and 4 units apart. In each distribution, the first hump resembles the linear distribution of a sphere, while the second hump represents the distance between the two spheres. Also in this case as the spheres move farther apart, the D2 distribution changes continuously.

Moreover, in figure 2.3 the descriptors of five tanks (gray) and six cars (black) are shown for the measuring function D2. Even if the two classes of objects are similar, the eleven descriptors are clustered in two separated families of curves, where the models belonging to the same object class have similar curves.

This behavior is not the same for all the measuring functions A3, D1, D2, D3 and D4. In figure 2.4 the descriptors of different classes of objects are shown with respect to the five different measuring functions. As can be argued from the figure, the measuring distance D2 classifies objects better than the other four functions. In [OFCD02] has been discussed why the others functions do not perform as well as D2. First, the D1 distributions tend to be sensitive to bumps on the surface of the object (e.g. the spikes in the D1 distributions for missiles are caused by fins protuding from the missiles bodies). Second, D1 distributions seem to be sensitive with respect to the object distribution around the centre of mass (e.g. the different size of the handles in the mug models). Third, the A3, D3 and

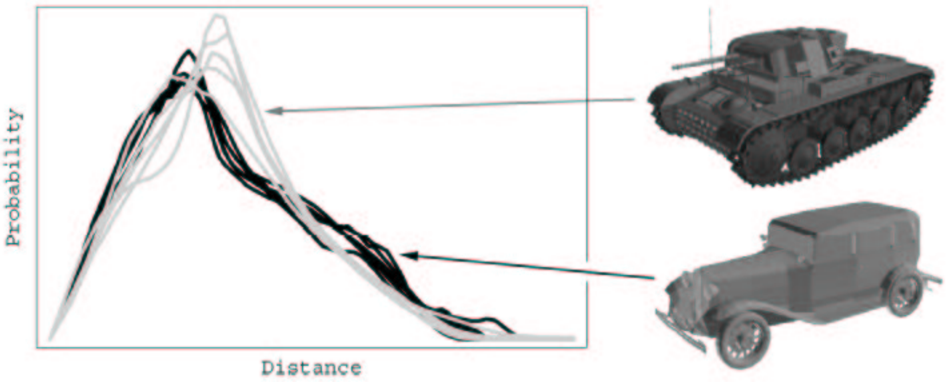


Figure 2.3: *Experiments with several versions of tanks and cars models. The figure is from [OFCD02].*

D4 distributions produce distributions that are similar in shape for diverse object classes. Although the intra-class diversity for D3 and D4 is smaller than for the other shape functions, the inter-class diversity is also smaller, which results in a less discriminating classifier. Overall, since the D2 shape function produces distributions with the best combination of distinctiveness and stability, it is the best object classifier.

2.1.2 Shape described by Oriented Points

The method proposed in [Joh97, JH97, JH99] samples the object surface in a set of oriented points, that is 3D points with surface normal (see figure 2.5(a)), and associates to each sampled point a description of the surface around it.

The description generated by an oriented point corresponds to an image (spin-image) obtained from a local base associated to the point. The whole object shape descriptor is the set of the sampled points, the normals and the set of images related to the points, see figure 2.5(b). The positions of the points lying on the surface, with respect to a sampled point, can be described by two parameters: the radial coordinate α, defined as the perpendicular distance to the line through the surface normal, and the elevation coordinate β, defined as the signed perpendicular distance to the tangent plane defined by vertex normal and position. Each sampled point, together with its surface normal can be considered as a local coordinate system. By coding these parameters in a 2D array, a descriptive image associated with the point is created. More formally an oriented point $\mathcal{O}$ defines a 2D local basis (p, n) using the tangent plane $\mathcal{P}$ through p oriented perpendicularly to the surface normal n and the line $\mathcal{L}$ through p parallel to n (see figure 2.6(a)). The two coordinates of the basis are α, the perpendicular distance to the line $\mathcal{L}$, and β the signed perpendicular distance to the plane $\mathcal{P}$. A spin-map $S_{\mathcal{O}}$ is the function that

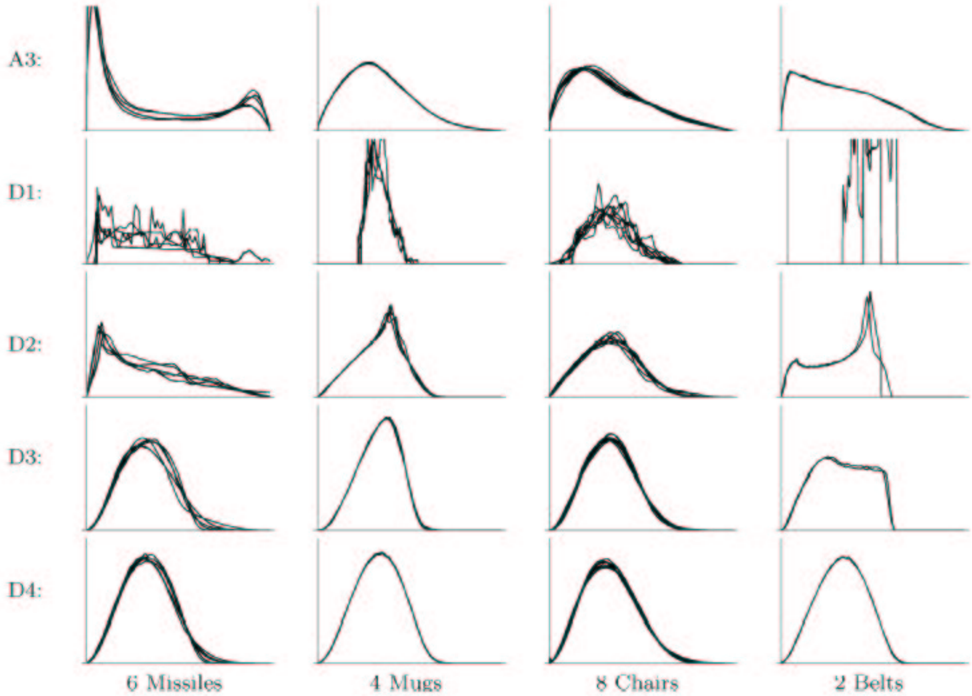


Figure 2.4: *Four families of 3d objects (missiles, mugs, chairs and belts) are described by the five measuring functions A3, D1, D2, D3 and D4. The figure is from [OFCD02].*

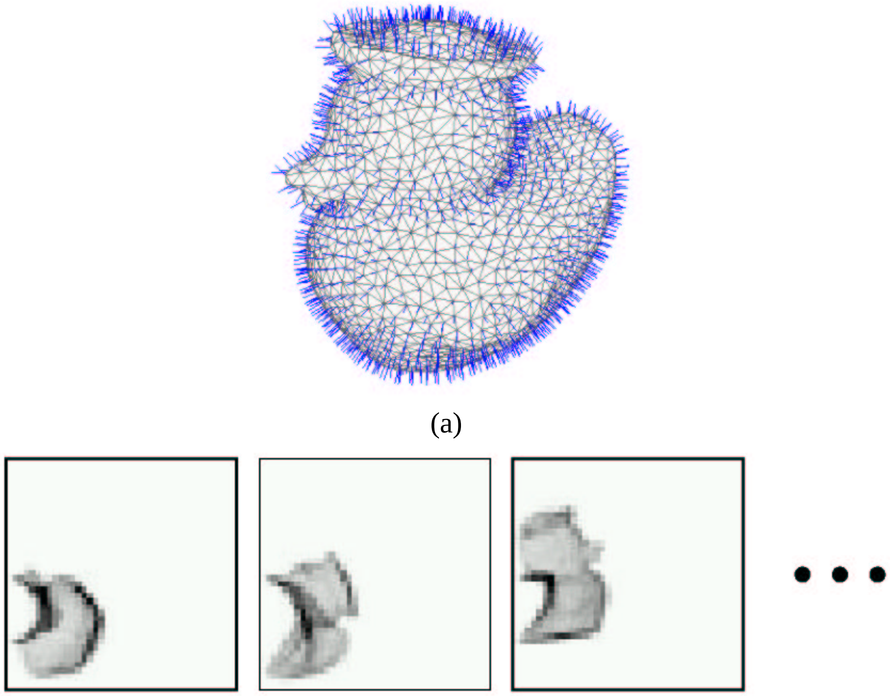
maps 3D points x to the 2D coordinates of a particular basis (p, n) corresponding to oriented point $\mathcal{O}$:

$$S_{\mathcal{O}}(x) = (\sqrt{\|x-p\|^2 - (n \cdot (x-p))^2}, n \cdot (x-p)) = (\alpha, \beta) \qquad (2.1)$$

Starting from the coordinates α and β of the points lying on the object surface, a 2D accumulator is created to store the image. The coordinates (α, β) are computed for each vertex in the surface mesh that is within the support of the spin-image. The bin indexed by (α, β) in the accumulator represent the spin-image: dark areas in the image correspond to bins that contain many projected points.As long as the size of the bins in the accumulator is greater than the median distance between vertexes in the mesh (the definition of mesh resolution), the position of individual vertexes will be averaged out during spin-image generation. Figure 2.6(b) shows the projected (α, β) 2D coordinates and spin-images for three oriented points on a duck model. Spin-images are constructed for every vertex in the surface mesh.

The generation of the spin-image depends on three parameters:

- **bin size:** is the geometric width of the bins in the spin-image, it determines the storage size of the image and the averaging that reduces the effect of individual point positions. Usually



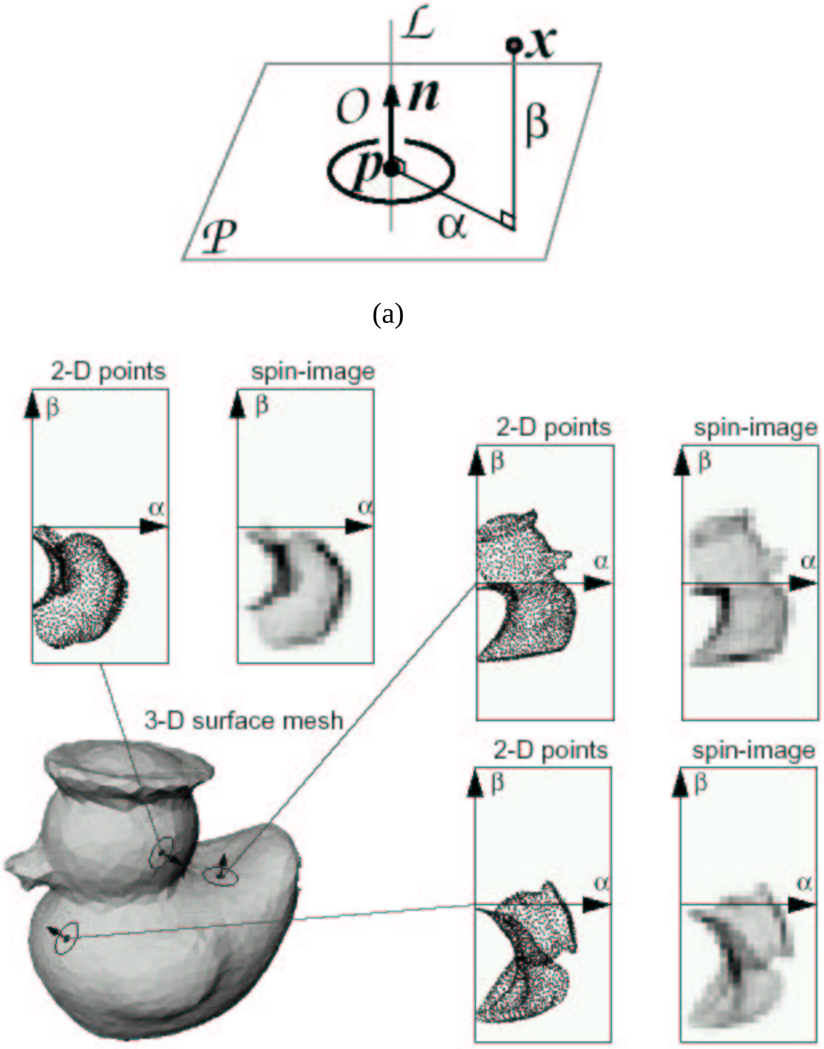
(a)

(b)

Figure 2.5: *Oriented point of a triangle mesh (a). The shape descriptor is represented as a set of images (spin-images), each one associated to the sampled oriented point. The figure is from [JH99].*

the bin size is set as a multiple of the resolution of the surface mesh in order to eliminate the dependence of setting bin size on object scale and resolution, this is reasonable because because mesh resolution is related to the size of shape features on an object and the density of points in the surface mesh. In figure 2.7 three different bin size has been tested with respect to the mesh resolution of the duck model. The spin-image generated for a bin size of four times the model resolution is not very descriptive of the global shape of the model. The spin-image generated with a bin size of one quarter the mesh resolution does not have enough averaging to eliminate the effect of surface sampling. The spin-image generated with a bin size equal to the mesh resolution has the proper balance between encoding global shape and averaging of point positions. In [JH99], has been proved that the image is representative of the model when the bin size is close to the mesh resolution.

- **image width:** it is the number of rows and columns of a spin-image. Usually the spin-image is squared, that is, it has the same number of rows and columns. For a fixed bin size, decreasing image width will decrease the descriptiveness of the spin-image because the amount of global shape included in the image will be reduced. Figure 2.8 shows spin-images for a single oriented



(a)

(b)

Figure 2.6: *The local coordinate system of an oriented point (a). Examples of spin-images extracted from oriented points (b). The figures are from [JH97, JH99].*

point on the duck model, where the image width is decreased. This figure shows that as image width decreases, the descriptiveness of the images decreases.

- **support angle:** it is the maximum angle between the direction of the oriented point basis of a spin-image and the surface normal of points that are allowed to contribute to the spin-image. Figure 2.9 shows the spin-image generated for three different support angles along with the vertexes on the model that are mapped in the spin-image. Also in this case decreasing of the support angle has the effect to decrease the descriptiveness of spin-images

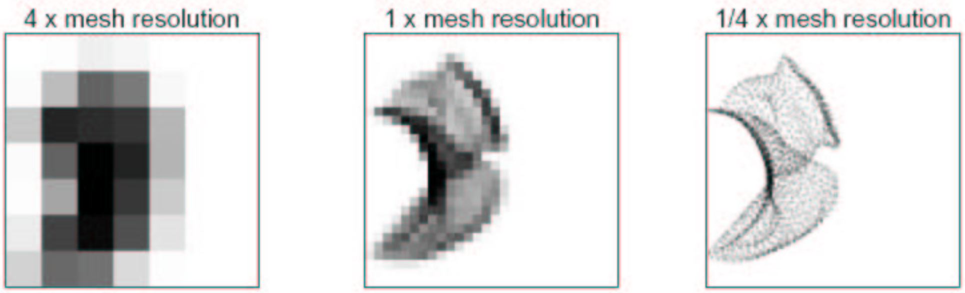


Figure 2.7: *Three examples of bin size for the duck model. The figure is from [JH99].*

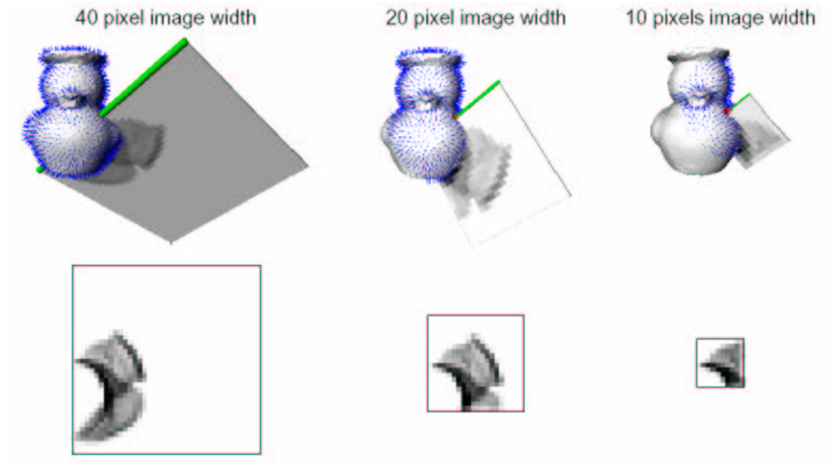


Figure 2.8: *Different image width compromise the spin-image descriptiveness capabilities. The figure is from [JH99].*

Spin-images have been defined for each sampled point. Even if the set of all the spin-images associated to the object defines a global shape descriptor, each of them provide a local description of the shape object. Since neighbor sampled points have similar spin-images, in [RCSM03] has been proposed to cluster similar image signature in order to improve the local descriptive capability of spin-images, defining a shape-class based descriptor. A shape class component is a group of connected surface mesh points whose image signatures are similar as defined by a clustering algorithm. The different components of a class can be represented on a labeled surface mesh (Figure 2.10(a)), where each vertex of the mesh has an associated symbolic label referencing the component in which it lies.

From the symbolic label a symbolic surface signature can be computed considering a point P on the surface mesh and a set of contributing points Q, which are defined in terms of distance from P (image width) and support angle, figure 2.10(b). The symbolic surface signature construction is obtained

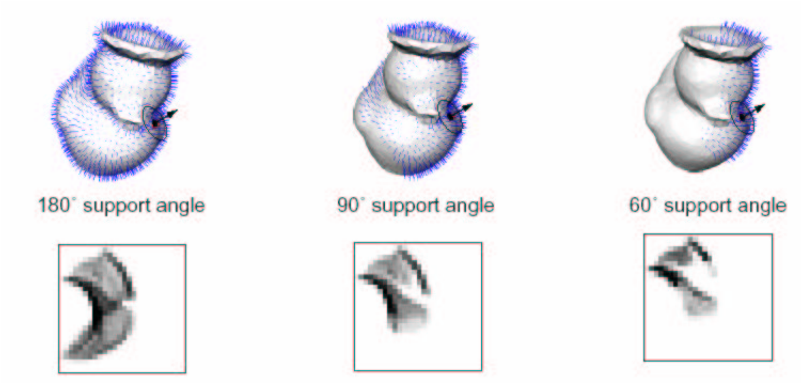


Figure 2.9: *Also the support angle influences the spin-image descriptiveness power. The figure is from [JH99].*

projecting the vector $\overline{PQ}$ on the tangent plane at P where the orthogonal axes γ and δ have been defined. As for the spin-images the discretized version of the γ and δ coordinates of $\overline{PQ}$ are used to index a 2D array, but in this case, the indexed position of the array is set to the component label of Q. Since it is possible that multiple points Q that have different labels project in the same bin in the symbolic surface signature, the label that appeared most frequently is assigned to the bin. The resultant array is the symbolic surface signature at point P. This signature captures the relationships among the labeled regions on the mesh as it is shown as a labeled color image in figure 2.10(c).

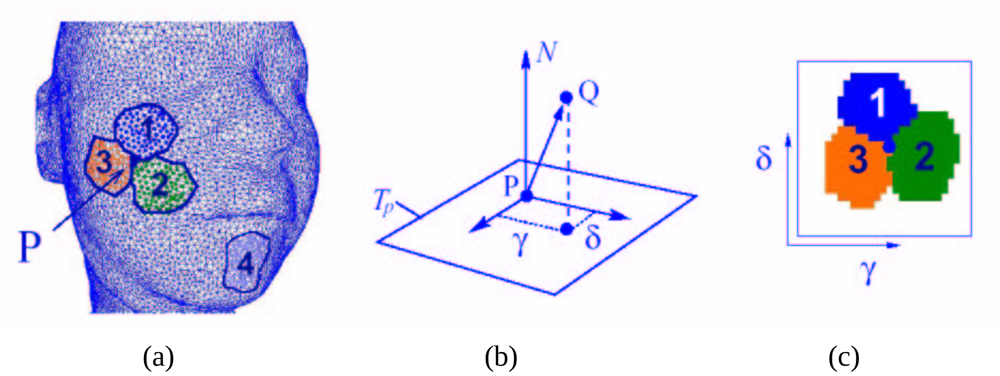


Figure 2.10: *Some example of labeled surface mesh (a). The local coordinate system for the construction of the symbolic surface signature (b), and finally some examples of symbolic surface signature . The figure is from [RCSM03].*

2.2 Survey on Exact Shape Descriptors

Differently from the statistical shape descriptors, exact methodologies uses all the information contained in the model, nevertheless the scalability of the descriptors (if any) can adapt the amount of information coded, to the specific application task. Moreover, given a methodology, an algorithm and the settings needed to apply the algorithm to the object model, the shape descriptor extracted, satisfies the uniqueness property described in section 1.4.

2.2.1 Coarse Shape Description

The correct granularity of the object features captured by the shape descriptor depends, mainly, on the application contexts on which it is used for. Some application tasks, especially dealing with shape retrieval, do not need a refined object description, but just a coarse analysis aimed at provide the smallest information amount used to roughly discriminate between objects. This is the case of the work proposed in [CRC$^+$02].

One of the contribution of the work, with respect to the shape description, is the definition of shape-metrics based on the geometric properties of an object and its convex hull (figure 2.11).

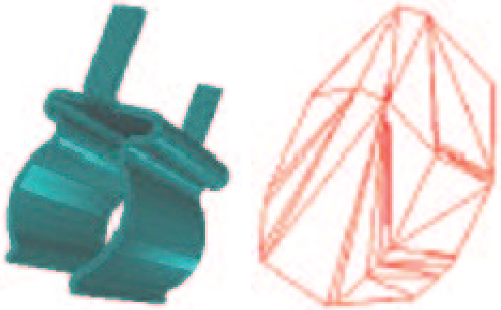


Figure 2.11: *A mechanical part and its convex hull. The figure is from [CRC$^+$02].*

The following shape-metrics involve the bounding box, the volume and the surface area of the object and the convex hull.

- **Bounding-box aspect ratio** is determined by identifying the objects axis-aligned bounding box. The ratio of the longest to the shortest edge of the box, is used, as a non-dimensional metric.
- **Hull crumpliness** is the ratio of the object's surface area to the surface area of its convex hull:

$$Hcp = \frac{A_{mod}}{A_{Hull}}$$

- **Hull packing** is the percent of the convex hull volume not occupied by the original object:

$$Hp = 1 - \frac{V_{mod}}{V_{Hull}}$$

- **Hull compactness** is the non-dimensional ratio of the convex hull's surface area cubed over the volume of the convex hull squared:

$$Hc = \frac{{A_{Hull}}^3}{{V_{Hull}}^2}$$

In figure 2.12 an example of the previous metrics is shown on a small set of models obtained through deforming operations on two different mechanical parts.

The models shown in the figures has been obtained through the following model deformations: tapering along an axis orthogonal to maximum and minimum principal axes; removing singly connected features, such as depressions and protrusions emanating from a single face; nonuniform scaling in order to fit the model into a $100 \times 100 \times 100$ cube, producing nonuniform deformations of its parts; and finally twisting the model along an axis orthogonal to maximum and minimum principal axes, producing parts that differs from the original model with respect to the underlying geometry, and thus in size and distribution of facets.

As shown in the figure, the axis-aligned bounding box ratio is sensitive to the feature removal, since a change in the model mass causes a modification of principal components. Hull crumpliness appears relatively insensitive to the effects of the twisting and tapering deformations. However, feature removal and nonuniform scaling cause large changes in surface area and create sensible variations of the metric. Also hull packing is sensitive to features removal, by the other hand it is almost insensitive to any other transformation. Finally, hull compactness varies with respect to the hull dimensions, and although its sensitivity to tapering is small, both twisting and nonuniform scaling cause large changes in its value. However, in cases where the hull remains unchanged (as in feature removal), the value is unchanged.

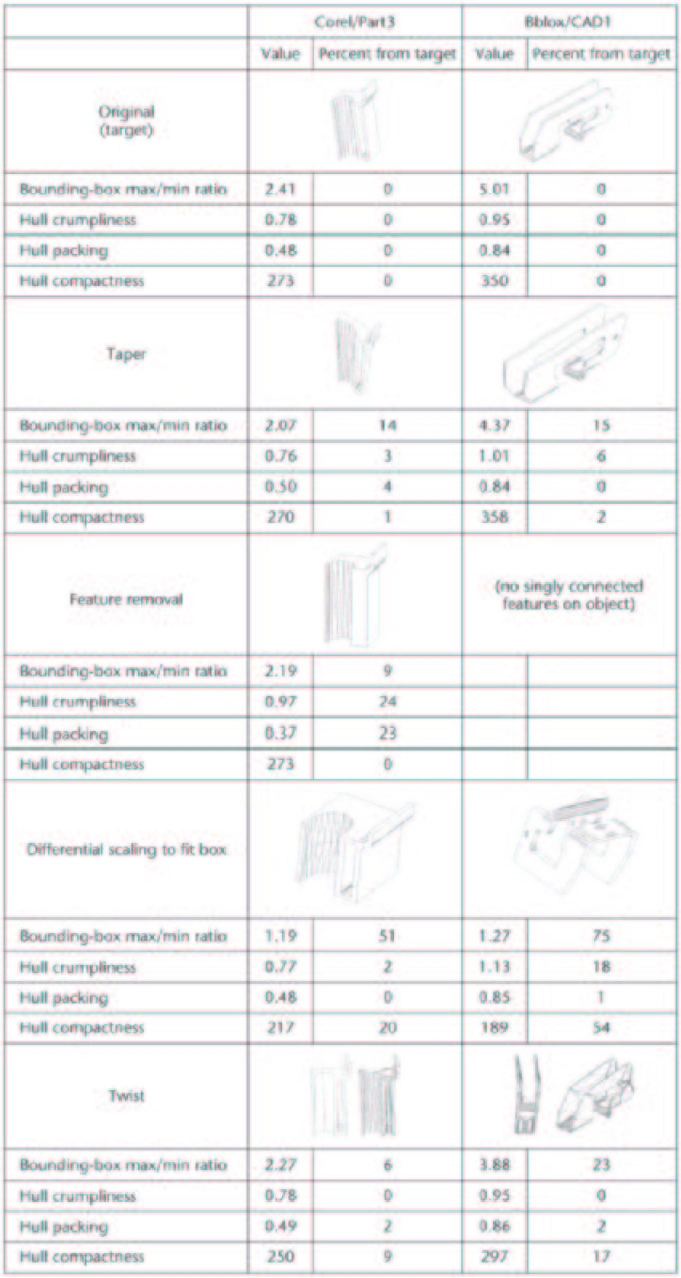
2.2.2 Spherical Harmonics Analysis

Given a model object and its centre of mass, the shape of the object can be described by the distribution of the object extent with respect to the directions starting from the centre of mass (ray-based method).

In [VSR01, Vra04], the object extent has been formalized as a function defined on the unit sphere $S^2 \subseteq \Re^3$:

$$\begin{aligned} f \quad &: \quad S^2 \rightarrow [0, +\infty) \\ & \quad u \ \ \rightarrow max\{r \geq 0 | ru \in I \cup \{O\}\} \end{aligned} \tag{2.2}$$

where I is the model, O is its centre of mass and $u = (\theta, \varphi) = (\cos\varphi \sin\theta, \sin\varphi sin\theta, \cos\theta)$ is the direction between O and a point on the sphere S^2 $(0 \leq \theta \leq \pi, 0 \leq \varphi \leq 2\pi)$, the angle θ is measured



	Corel/Part3		Bblox/CAD1	
	Value	Percent from target	Value	Percent from target
Original (target)				
Bounding-box max/min ratio	2.41	0	5.01	0
Hull crumpliness	0.78	0	0.95	0
Hull packing	0.48	0	0.84	0
Hull compactness	273	0	350	0
Taper				
Bounding-box max/min ratio	2.07	14	4.37	15
Hull crumpliness	0.76	3	1.01	6
Hull packing	0.50	4	0.84	0
Hull compactness	270	1	358	2
Feature removal			(no singly connected features on object)	
Bounding-box max/min ratio	2.19	9		
Hull crumpliness	0.97	24		
Hull packing	0.37	23		
Hull compactness	273	0		
Differential scaling to fit box				
Bounding-box max/min ratio	1.19	51	1.27	75
Hull crumpliness	0.77	2	1.13	18
Hull packing	0.48	0	0.85	1
Hull compactness	217	20	189	54
Twist				
Bounding-box max/min ratio	2.27	6	3.88	23
Hull crumpliness	0.78	0	0.95	0
Hull packing	0.49	2	0.86	2
Hull compactness	250	9	297	17

Figure 2.12: *Two mechanical parts (Corel/Part3 and Bblox/CAD1) has been deformed (taper, features removal, scaling and twisting) and the resulting metrics has been computed. The figure is from [CRC$^+$02].*

down from the z-axis, while φ is measured counterclockwise off the x-axis in the plane xOy. The real value r represents the furthest model intersection of the ray emanated from O having direction u.

In figure 2.13 an example of shape extent with respect to the object centre of mass is shown.

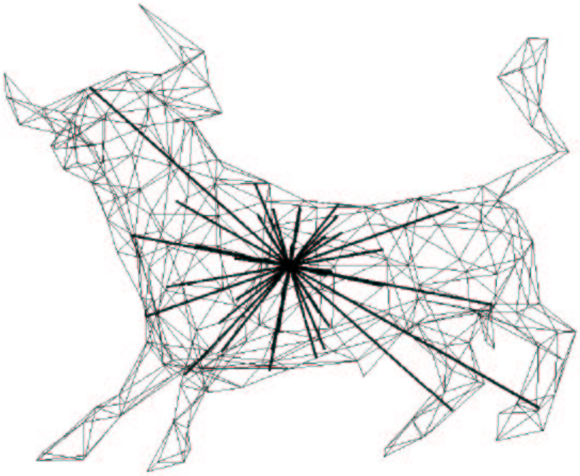


Figure 2.13: *A model of a bull and a schematic representation of its ray-based shape extent. The figure is from [Vra04].*

The descriptor associated to this shape representation is the vector of real coefficients obtained by the following steps (For details see [Vra04]):

1. pose estimation through principal component analysis [VSR01, Vra04] in order to achieve the invariance of the descriptor with respect to rotation, scaling, translation and reflection;
2. sampling the function f into $4B^2$ points;
3. performing the Spherical Fast Fourier Transform (SFFT) on the samplings in order to obtain B^2 complex coefficients;
4. considering the magnitudes of the coefficients as real elements of the vector (shape descriptor).

The spherical harmonic coefficients can be used to reconstruct an approximation of the underlying object[2] at different levels of detail as shown in figure 2.14.

Also in [KFR03, FMK$^+$03] a shape descriptor based on spherical harmonics has been proposed. It decomposes a 3D model into a collection of functions defined on concentric spheres, where for each function spherical harmonics decomposition is used to produce a 1D descriptor. Combining the 1D descriptors, by analyzing spheres at different radii, a 2D descriptor is obtained. From the properties of the spherical harmonics decomposition this shape descriptor is independent by the orientation of the coordinate system. An example of shape descriptor obtained by an aeroplane is shown in figure 2.15.

[2] Since the distribution of the object extent is based on the ray-based method, only star-shaped objects can be reconstructed.

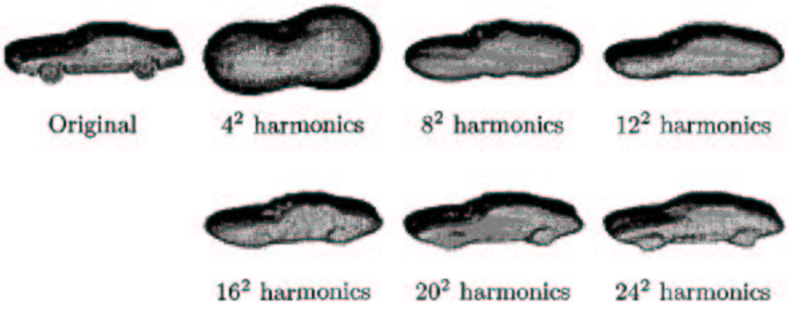


Figure 2.14: *Car model obtained by a multi resolution representation of the function measuring the shape extent (equation 2.2). The figure is from [VSR01].*

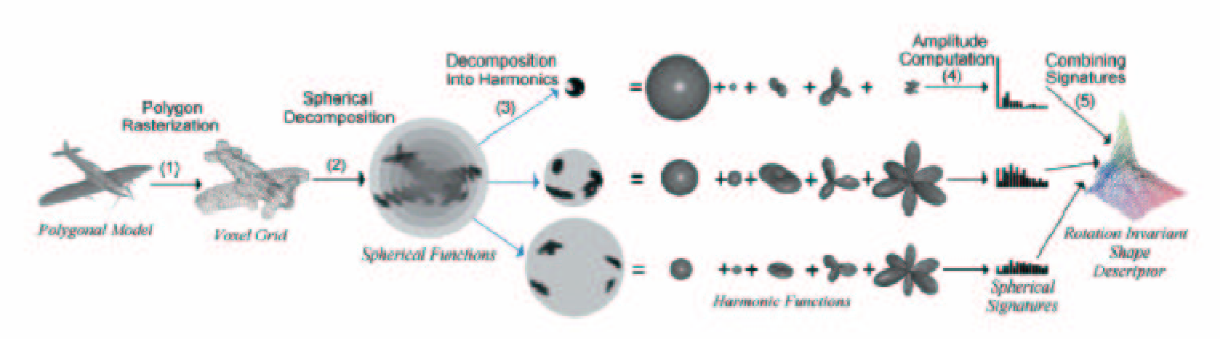


Figure 2.15: *The model of an aeroplane is voxelized, then its shape-descriptor is obtained through the spherical harmonic analysis of the function defined on the voxelization. The figure is from [FMK$^+$03].*

The method to obtain the descriptor is summarized through the following steps:

1. The centre of mass of the model object is translated into the origin of the coordinate system. Moreover the model is scaled by dividing it by the average distance of a point on the model surface to the centre of mass. This normalization approach is less sensitive to outliers than the one based on the centre and radius of the bounding sphere;

2. The model surface is rasterized in order to obtain a $2R \times 2R \times 2R$ binary voxel grid, where the grid is defined as:

$$\eta_{abc} = \left\{ (x,y,z) \,\middle|\, a = \left\lfloor \frac{x+2}{2} R \right\rfloor + 1,\ b = \left\lfloor \frac{y+2}{2} R \right\rfloor + 1,\ c = \left\lfloor \frac{z+2}{2} R \right\rfloor + 1 \right\} \quad (2.3)$$

   where $a, b, c \in \{(1, \ldots, 2R)\}$ and $-2 \leq x, y, z \leq 2$;

3. The value 1 is assigned to each voxel that intersects the object surface, and 0 otherwise:

$$v_{abc} = \begin{cases} 0 & \eta_{abc} \cap I = \phi \\ 1 & \eta_{abc} \cap I \neq \phi \end{cases} \tag{2.4}$$

   where I is the normalized model;

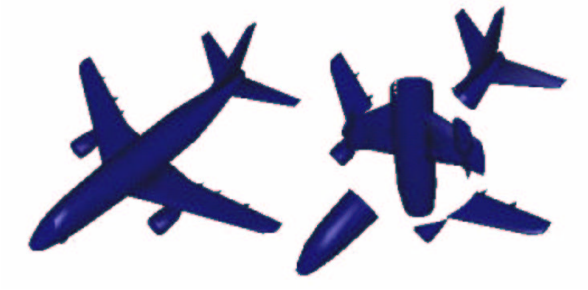
4. A binary function on the sphere of radius r is defined as:

$$f_r(\theta, \varphi) = v_{abc}$$

   where $\theta = [0, \pi]$, $\varphi = [0, 2\pi]$ and $(r \sin\theta \cos\varphi, r \cos\varphi, r \sin\theta \sin\varphi) \in \eta_{abc}$ and $r = (1, \ldots, R)$;

5. Using spherical harmonics, each function f_r can be expressed as a sum of its different frequencies [FMK$^+$03, KFR03, Vra04]. The 1D shape descriptor is the vector whose values are the magnitudes of the coefficients obtained from the spherical harmonics analysis and the 2D descriptor is the combination of the 1D descriptors. See figure 2.15.

The spherical harmonic representation is invariant to rotation of the model object, but it is also invariant to independent rotations of the different spherical functions. For example, the plane in figure 2.16(b) is obtained from the one in figure 2.16(a) by applying a rotation to the interior part of the model. The two models are not rotations of each other, but the descriptors obtained are the same.



(a) (b)

Figure 2.16: *A model of an aeroplane (a), and a model of the same aeroplane obtained through rotation of its interior part (b). The figure is from [KFR03].*

2.2.3 Weighted Point Sets

The key idea proposed in [VT03] is to represent a polyhedral model object as a weighted point set describing the salient characteristics of the object. Since the spatial distribution of the weighted point set depends on the coordinate system, the same object embedded in two different coordinate systems produces two different spatial distribution of the point set, thus two different shape descriptors. In order to satisfy the uniqueness property of the descriptor (section 1.4), the model object has to be put in a canonical coordinate system (standard pose). The method used in [VT03] to bring the model object into the standard pose is the principal component analysis defined by the principal axes of inertia [VSR01, Vra04].

Once the model has been put into the standard pose, it is scaled and enclosed by a 3D grid as shown by the equations 2.3 and 2.4. For each non-empty cell ($v_{abc} = 1$) of the grid, a weighted point is generated in order to describe the part of surface enclosed in the cell. Three methods has been proposed to generate the weighted points:

- **Gaussian Curvature**: it is denoted as $c(v)$ and can be computed as in the following [Cal85]:

  $$c(v) = \frac{d(v)}{a(v)}$$

  where $d(v)$ is the angular defect at v and it is defined for interior vertexes as 2π minus the sum of the interior angles of the polygons meeting at v. For vertexes at the boundary of a gap the it is defined as π minus the sum of the interior angles of the facets meeting at v. The scalar $a(v)$ denotes the area associated with vertex v. It is the sum of the areas of the polygons meeting in v divided by the number of their vertexes. The flatter the surface, the smaller will be. The weighted point describing the part of surface belonging to the non-empty cell is $(v, \overline{c(v)})$, where v is the vertex in the cell having the maximum Gaussian curvature, and $\overline{c(v)}$ is the value of its curvature normalized to the range $[0, 1]$;

- **Normal Variation**: for each grid cell the area-wighted normal mean of the polygons having the vertexes belonging to the grid cell is computed as [BW00]:

  $$\overrightarrow{mn} = \sum_{i=1}^{N} a_i \overrightarrow{n_i}$$

  where N is the number of polygons in the grid cell, $\overrightarrow{n_i}$ and a_i are the normal and the area of the i^{th} polygon. $\overrightarrow{mn}$ is the area-wighted mean of the normals. The flatter the surface, the larger the magnitude of $\overrightarrow{mn}$ will be. If all the polygons are coplanar, the magnitude of $\overrightarrow{mn}$ will equal the area of the model surface in to the cell, thus its normalization to the range $[0, 1]$ is defined as $cp = \|\overrightarrow{mn}\| \sum_{i=1}^{N} a_i$. The weighted point associated to the grid cell is thus the pair $(p(c), 1 - cp)$, where $p(c)$ is the area-wighted mean of the vertexes belonging to the cell, and it is defined as:

  $$p(c) = \left(\sum_{j=1}^{M} w_j v_j \right) / \sum_{j=1}^{M} w_j$$

  where M is the number of the vertexes in the cell, v_j and w_j are the j^{th} vertex of the cell and its related area defined as the sum of the area of the polygons incident in v_j, each one divided by number of its vertexes.

- **Midpoint**: it is simply the average of all the vertexes belonging to the grid cell with unit weight. This simple representant of the surface enclosed in the cell is useful to extract shape descriptor of, models where polygons are wrongly oriented (e.g. ”polygon soups”).

In figure 2.17 an example of a model representing an aeroplane and its shape descriptor based on the midpoint method is shown.

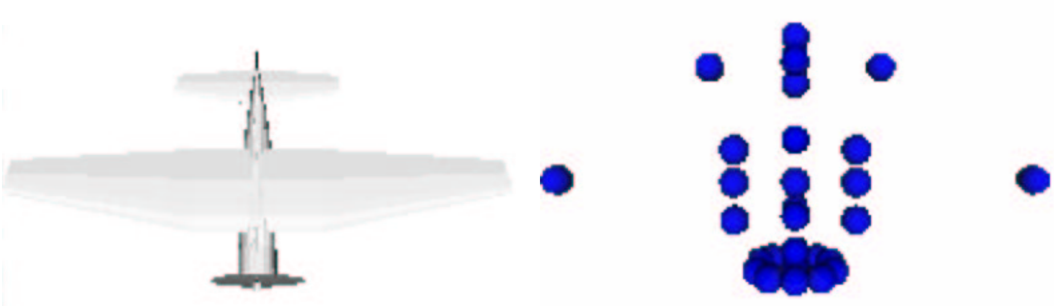


Figure 2.17: *A shape model and the corresponding weighted point set obtained through the midpoint method. The figure is from [VT03].*

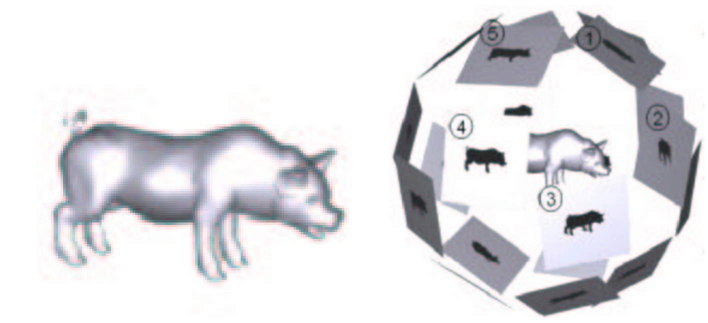
2.2.4 Shape Described by Set of Views

The methodology proposed in [COT$^+$03] describes a 3D object as a collection of 2D images, each one obtained as if a set of cameras was uniformly distributed on the surface of a sphere containing the 3D object (light fields [3]). An example of how a camera can be positioned around a 3D model is shown in figure 2.18(b). In order to reduce the size of the shape descriptor, only the silhouettes of the 2D images are considered and the light field cameras has been put on twenty vertexes of a regular dodecahedron. That is, there are twenty different views, which are distributed uniformly, over a 3D model. Since the silhouettes projected from two opposite vertexes on the dodecahedron are identical, only ten different silhouettes are produced for a 3D model.

Once a silhouette of a 2D view has been found, it is processed in order to carry out its shape descriptor. The method is based on the centroid distance function expressed by the distance to the boundary points from the centroid of the silhouette [ZL02]. Figure 2.19 shows a typical example of the centroid distance. Figure 2.19(a) shows a 2D shape rendered from a viewpoint of a 3D model, and the contour tracing result is shown in Figure 2.19(b). Figure 2.19(c) shows the centroid distance of (a).

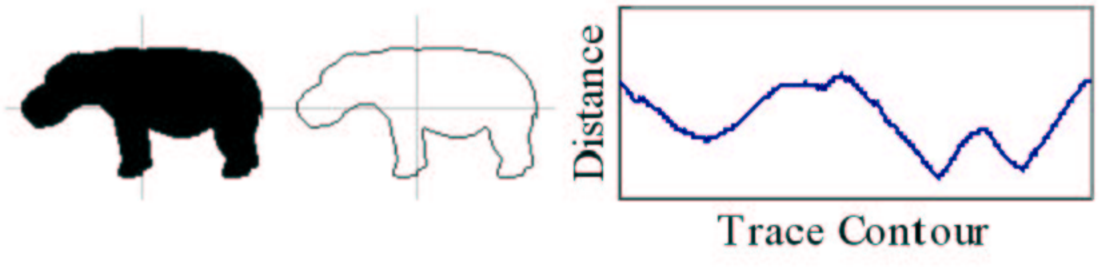
The process that extracts the light field shape descriptor can be summarized as in the following steps:

[3] The phrase light field describes the radiometric properties of light in a space. A light field (or plenoptic function) is traditionally used in image-based rendering [Ger39] and is defined as a five dimensional function that represents the radiance at a given 3D point in a given direction. For a 3D model, the representation is the same along a ray, so the dimension of the light field around an object can be reduced to 4D [LH96].



(a) (b)

Figure 2.18: *The model of a pig (a) and twenty different views of its uniformly distributed around it (b). The figure is from [COT+ 03].*



(a) (b) (c)

Figure 2.19: *A 2D view of a 3D object (a), its silhouette (b) and the centroid distance with respect to the centroid of the silhouette (c). The figure is from [COT+ 03].*

1. Translation and scaling are applied and the input 3D model is translated to the origin of the coordinate system. The axis is then scaled such that the maximum length is 1.

2. For a Light Field Descriptor, 10 images are represented for 20 viewpoints in a pre-defined order for storage. For a 3D model, 10 descriptors are created in order to make the shape descriptor robust with respect to rotations. Totally 100 images should be rendered.

3. Descriptors for a 3D model are extracted from the 100 images.

Also in [CK01] a 3D model object is described by a set of 2D views. In this case a similarity metric on the 2D view is defined to group similar views into clusters, named aspects[4], where each aspect is

[4] The term aspect comes from the aspect graph defined in [KvD76, KvD79]. The aspect graph is defined as the set of regions of the viewing sphere where equivalent views and neighborhood relations on the viewing sphere generate a graphical structure of views. Each node of the aspect graph represents a general view, or aspect, of the 3D object and represents a maximally connected region on the viewing sphere. Each link represents some visual event where transitions occur between

represented with a prototypical view (see figure 2.20). Views where similarity varies rather slowly should generate larger size aspects, for example side views of four-legged animals, while regions of rapid change in similarity should lead to smaller aspects, for instance frontal views of four-legged animals.

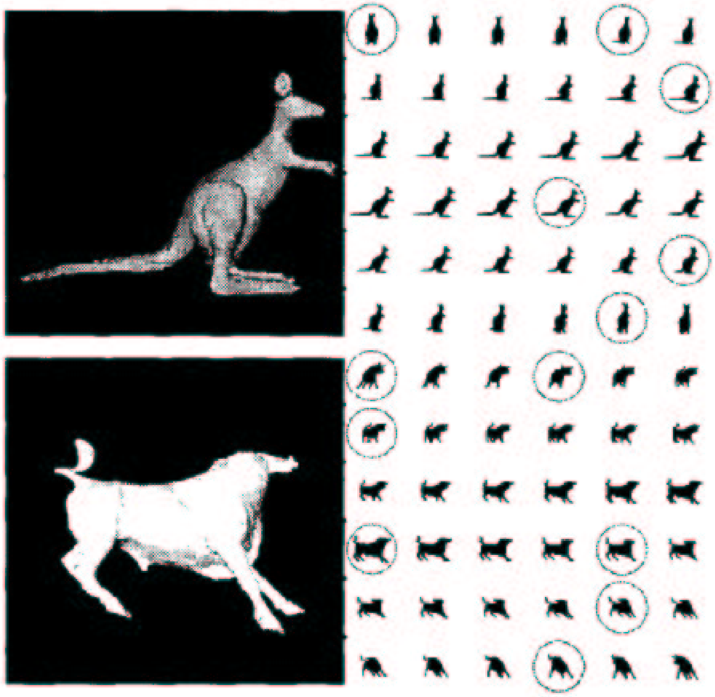


Figure 2.20: *Two 3D models (on the left) and their set of views (on the right). The circled views represent the prototypical views. The figure is from [CK01].*

Given a set of N objects $\{O_1, \ldots, O_N\}$ and the set of M views $\{V_n^1, V_n^2, \ldots\ V_n^M\}$ of the object n, the aspect defined by a contiguous collection of views around a prototype V_n^m is defined as:

$$A_n^m = \{V_n^{m-l^-}, \ldots, V_n^m, \ldots V_n^{m+l^+}\}$$

where l^+ and $-l^-$ are defined as the left and right radius of the aspect and which depend on the object n and the prototypical view V_n^m. In [CK01] has been defined two criteria in order to carry out the elements of the aspect and the prototype view. They are based on the similarity metric between two silhouettes V_n^m and V_i^j:

$$d(V_n^m, V_i^j) \tag{2.5}$$

In [CK01] has been experimented two image metrics: one based on the curve matching and the other based on the shock graph matching. The first criterion is the *l*ocal monotonicity. It assures that the

two neighboring general views, namely, the accidental views.

shape similarity metric should be increasing as the viewing angle between two shapes is increased for some local neighborhood. For each view V_n^m there exists $\delta > 0$ such that:

$$d(V_n^m, V_n^{m+i}) < d(V_n^m, V_n^{m+j}) \tag{2.6}$$

where $|i| < |j| < \delta$. Equation 2.6 limits the size of an aspect centered around a candidate prototype. The second criterion is the *o*bject-specific distinctiveness. It states that the metric of similarity should also be able to differentiate between aspect and non-aspect views. For each aspect A_n^m with prototypical view V_n^m, the following property have to be satisfied for any $V_n^i \in A_n^m$ and for any $V_n^j \notin A_n^m$:

$$d(V_n^i, V_n^m) < d(V_n^j, V_n^{m+j}) \tag{2.7}$$

It delimits the possible aspect boundaries for each candidate prototype views, since:

$$\max_{V_n^i \in A_n^m} d(V_n^i, V_n^m) < \min_{V_n^j \notin A_n^m} d(V_n^j, V_n^{m+j})$$

The aspects and prototypes are then generated by seeking a maximal region where equations 2.6 and 2.7 hold. Figure 2.21 shows the model of a Kangaroo and its prototypes. The et of views from which the prototypes have been extracted (see right part of the figure2.20) have been obtained sampling the ”ground plane” of the object with a frequency of 5 degrees.

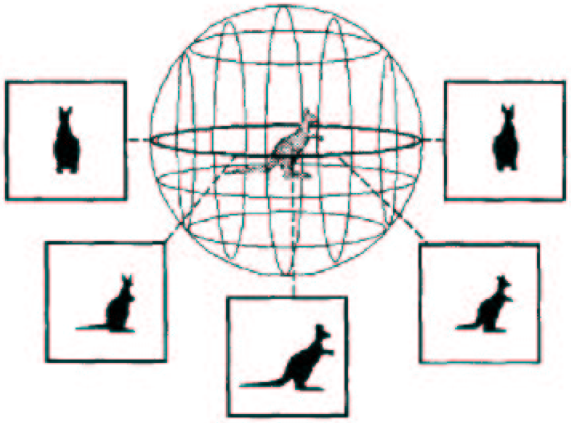


Figure 2.21: *The 3D models of a kangaroo and its prototypical views. The figure is from [CK01].*

2.3 Methods for Geometrical Shape Similarity

In sections 2.1 and 2.2 some of the most important geometric shape descriptors have been described. In this section are shown the methodologies used to compare them, in order to estimate the similarity (or dissimilarity) of shape among 3D model objects. Even if geometric descriptors capture shape characteristics through different methodologies, often they share the way to store the information coded. When the shape is coded as vector or matrix, Minkowski norm $l_n, n = 1, 2, \infty$ the best suitable way to evaluate the distance between descriptors. In other cases the evaluation process goes through optimization techniques.

All the shape descriptors discussed in the sections 2.1.1, 2.2.1 and 2.2.2 store the information captured from the object as vectors or matrices of real values, therefore similar 3D objects are close in the vector/matrix space and thus the l_n norm is suitable to compute the nearest neighbors of a given shape descriptor. In particular the shape descriptor proposed in [OFCD02, CRC$^+$02, Vra04] are real valued vectors, instead the one proposed in [KFR03] is a real valued matrix.

The methodology proposed in [COT$^+$03] and described in section 2.2.4, describes a 3D object by a set of 2D images (light field descriptors) corresponding to the set of views sampled on the vertexes of a regular dodecahedron containing the 3D model. To make the shape descriptor robust to rotation, a set of N light field descriptors is applied to the 3D model. The dissimilarity distance D_B, between two 3D models is then defined as:

$$D_B = \min\ D_A(L_j, L_k), \quad j, k = 1, \dots, N$$

where L_j and L_k are the light field descriptors of the two models, respectively and the function D_A is the similarity between two light field descriptors, and is defined as:

$$D_A = \min_i \sum_{k=1}^{10} d(I_{1k}, I_{2k}), \quad i = 1, \dots, 60$$

where d denotes the dissimilarity between two images, and i denotes different rotations between camera positions of two 3D models. For a regular dodecahedron, each of the twenty vertexes is connected by three edges, which results in 60 different rotations for one camera system. I_{1k} and I_{2k} are corresponding images under i^{th} rotation. Also the shape descriptor proposed in [CK01] and discussed in section 2.2.4 stores a small set of prototypical views of a 3D object. In this case the comparison between two 3D object is simply obtained performing the matching between all the prototypical views related to each object using the distance function defined in equation 2.5.

The method described in section 2.1.2 compares spin-images from points on a given surface, to spin-images from points of another surface. Differently from the other shape descriptors the target of the spin-images is point correspondence between two similar surfaces through image comparison.

The standard way of comparing spin-images is the normalized linear correlation coefficient, used to estimate the distance between the data and the best least squares fit line to the data. Given two spin-images P and Q with N bins each, the linear correlation coefficient $R(P, Q)$ is [Joh97]:

$$R(P, Q) = \frac{N \sum p_i q_i \ - \sum p_i \ \sum q_i}{\sqrt{N \sum p_i^2 \ (\sum p_i)^2 \ (N \sum q_i^2 - (\sum q_i)^2)}} \tag{2.8}$$

R is between -1 and 1 for anti-correlation and complete correlation, respectively. The coefficient R provides a method for the comparison of two spin-images: when R is high, the images are similar; when R is low the images, are not similar. The correlation coefficient imposes an ordering on point correspondences, so good and bad correspondences can be differentiated.

In order to make more robust the comparison method between two spin-image the data used to compute the correlation coefficient is taken only from the region of overlap between the two spin images. Since the linear correlation coefficient is a function of the number of pixels used to compute it, the amount of overlap between spin-images effects the correlation coefficients obtained. The more pixels used to compute a correlation coefficient, the more confidence there is in its value, and the confidence in the correlation coefficient can be measured by its variance. The similarity measure C combines the correlation coefficient R and its variance into a single function [Joh97]:

$$C(P,Q) = (\mathrm{atanh}(R(P,Q)))^2 \; - \lambda\left(\frac{1}{N-3}\right) \tag{2.9}$$

Where P and Q are spin-images, N is the number of overlapping pixels used in the computation of R defined in equation 2.8 and λ weights the variance against the expected value of the correlation coefficient. C returns a high value for two images that are highly correlated and have a large number of overlapping bins.

In [Joh97] has been proposed a method for the establishment of point correspondences between oriented points of two surface meshes (the model and scene surface meshes respectively) that uses the similarity estimation defined in equation 2.9. The point correspondence algorithm is summarized by the following steps:

1. Spin-images are generated for all points on the model surface mesh, and then these images are stored in a spin-image stack;
2. a scene point is randomly selected from the scene surface mesh and its spin-image is generated;
3. the similarity measures defined in equation 2.9 is computed between the spin-image of the scene point and each spin-image belonging to the stack;
4. These similarity measures are stored in a similarity measure histogram, where upper outliers correspond to model/scene pairs with similarity measures that are significantly higher than the rest of the model/scene pairs. These outliers correspond to plausible model/scene point correspondences.
5. This procedure to establish point correspondences is then repeated for a fixed number of randomly selected scene points.

The shape descriptor discussed in section 2.2.3 describes a 3D object as a set of wighted three dimensional points. In [VT03] has been proposed a methodology called Proportional Transportation Distance (PTD), for weighted point cloud matching, inspired by the Earth Mover's Distance (EMD)

[CG99]. The EMD between two weighted point sets measures the minimum amount of work needed to transport from a supplier set of weights to a demander set of weights. This distance has some drawbacks that make it not suitable in some application tasks: it does not satisfies the positivity property 1.25 and the triangle inequality property 1.27 discussed in section 1.3, finally, there are cases where it does not distinguish between two non-identical sets. Let N be a weighted point set and $A, B \in N$. When measuring the distance from A to B, rather than taking A as the supplier and B as the demander, the PTD moves the total weight of A to the positions of the points in B. What the PTD measures is the minimum amount of work needed to transform A to a new set A' that resembles B. In particular, it redistribute the total weight of A from the position of its points, to the positions of the points of B, leaving the old percentages of weights in B the same.

Let $A = \{a_1, \ldots, a_m\}$ a set of weighted points, where $a_i = (x_i, w_i)$ with $x_i \in \Re^k$ the point and $w_i \in \Re$ the weight. Let also $W = \sum_{i=1}^{m} w_i$ the total weight of A. Similar definitions hold for the set of weighted points B and for U, the tonal weight of B. Finally let d the Euclidean distance between two points and f_{ij} the elementary flow from x_i to y_j over the distance d_{ij}. The PTD distance is defined as a linear problem with the following constraints:

$$\begin{array}{lcll} f_{ij} & \geq & 0 & i = 1, \ldots, m, \ j = 1, \ldots, n \\ \sum_{j=1}^{n} f_{ij} & = & w_i & i = 1, \ldots, m \\ \sum_{i=1}^{m} f_{ij} & = & \frac{u_j W}{U} & j = 1, \ldots, n \\ \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} & = & W & \end{array}$$

and the following objective function:

$$PTD(A, B) = \frac{\min_{F \in \mathcal{F}} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{W}$$

where $\mathcal{F} = [f_{ij}]$ is the set of all the feasible flows from A to B. The PTD distance is a pseudo-metric (see section 1.3) since it satisfy the identity and the triangle inequality properties but it does not obey the positivity property.

2.4 Discussion

In section 1.4 some properties characterizing shape descriptors have been stated. The subset of properties making the descriptor suitable for a specific application depends on the task for which it is used for. Therefore it would not make sense to expect that a shape descriptor satisfies all the mentioned properties.

The **saliency** of a shape descriptor assures to capture all the necessary information to represent or describe the shape of the object. No shape descriptors discussed in this chapter have the capability to *e*xplicitly code topological or structural features of the object. Nevertheless the existence of topological features like handles, structural features like protrusions and geometric salient features like the rondness or sharpness of the object influence, in a general fashion, the information encoded in the geometric descriptor. For example missiles and mugs of figure 2.4 have a different shape distribution with respect to the measuring function $D2$, by the other hand, the four different mugs belonging to the same family have similar $D2$ shape distribution due to their common salient features (see section 2.1.1). This is because the descriptor *i*mplicitly incorporate the salient features of the object. Shape distribution described in section 2.1.1 have a good capability to discriminate among different shapes because it store, in some way, salient features of the object. However it is not easy to understand in which way the salient features influence the form of the descriptor with respect to the overall shape of the object. The same considerations hold for the other geometric shape descriptors described in this chapter. Since shape distribution (section 2.1.1) and spin-images (section 2.1.2) are extracted trough a stochastic-based process, the capability to encode salient features mainly depends on the frequency and the uniformity of the samplling, e.g. coarse sampling will provide a coarse description of the object. While coarse filters described in section 2.2.1 provide an implicit and very small amount of information about salient features. The descriptors based on the spherical harmonic analysis (section 2.2.2) can capture salient features increasing the resolution of the spherical harmonic representation. Similar considerations hold for the descriptor discussed in section 2.2.3. Increasing the number of cell from which the weighted point set is extracted increases the capability of capturing salient features. Also descriptors based on the object views (section 2.2.4) implicitly encode salient features of the object and also in this case the saliency of the feature coded, depends on the number of views captured.

Relevant features are the subset of salient features that are useful in a specific context. From this point of view, the capability of a descriptor to encode relevant features mainly depends on: 1) the capability to capture salient features; 2) the capability to explicitly represent/describe relevant features. With respec to the example of a virtual human that need to know how to grasp an object (section 1.4, concerning the relevance property) it is quite impossible to understand which is the part of the object that can be grasped just analyzing a geometrical shape descriptor[5].

One of the properties that makes the geometric descriptors very useful is the **cncisness**. It is the capability of a descriptor to minimize the memory occupied maximizing the information encoded. Geometric descriptors that codifies the shape as vectors or matrices (e.g. sections 2.1.1, 2.2.1 and 2.2.2) or weighted point clouds (section 2.2.3) have good performances with respect to concisness since the number of elements of the vector/matrix is usually much smaller than the number of elements of the object model (e.g. the number of polygons in polygonal meshes). Descriptors based on spin-images (section 2.1.2) and set of views (section 2.2.4) take much more memory due to the number of the 2D images collected.

The **uniqueness** property is given by the methodology and the algorithm used to extract the shape descriptor, and also to the parameter tuning necessary to run the algorithm on a model input. Descriptors extracted through statistical methods (sections 2.1.1 and 2.1.2) are not unique due to the random sampling performed on the model information. Also the method proposed in [COT$^+$03] (section 2.2.4) is not unique. In that case the views captured by the algorithm depend on the coordinate

[5] Perhaps, relevant features can be deduced from structural analysis of spin-images (section 2.1.2) and from the collection of views (section 2.2.4), but no works have been proposed on this topic.

system of the model. The authors, instead of posing the object into a canonical coordinate system, extract a large number of view to make the descriptor more robust against rotations. On the other hand the method proposed in [CK01] (section 2.2.4) captures the object views with respect to the ground plane of the object. In this last case is intention of the authors to discriminate among the same object lying in different positions. The similarity measures defined on shape descriptors that do not obey to the uniquenes do not satisfy the self-identity property 1.24 shown in section 1.3. Besides, fixed the algorithm parameters the other exact descriptors discussed in section 2.2 are unique.

A shape descriptor is **invariant to rigid transformations** if it is invariant to rotation, translation and scale transformation of the coordinate system of the object. This is the case of the measuring functin $A3$ of the descriptor proposed in section 2.1.1. The other measuring functions ($D1, D2, D3, D4$) have to be normalized to enable object comparisons. The descriptors proposed in [Vra04] (section 2.2.4) and [VT03] (section 2.2.3) need to be put in a canonical coordinate system in order to be used for shape comparison, otherwise different rotations or scaling of the same object would result in different shape descriptrs. Descriptors based on collection of views discussed in section 2.2.4 are not invariant to rotatoins. Finally spin-images and coarse filters, sections 2.1.2 and 2.2.1 respectively are invariant to rigid transformations.

The methods for the extraction of geometric descriptors proposed in section 2.1.2 and 2.2.3 are not **invariant to the model representation**. Spin-images are computed through the analysis of oriented points, that is points endowed with normal vectors, and the methods used to generate the weights of the point-set proposed in [VT03] need a polygonal mesh to be computed. The methods based on shape distribution (section 2.1.1), coarse filters (sectin 2.2.1), spherical harmonic analysis (section 2.2.2) and set of views (section 2.2.4) have been proposed for polygon sup even if in theory they coulc be defined for other model representations.

Scalability is the capability to select the relevant information contained in the descriptor and to discard the non relvant one. Since geometric descriptors does not explicitly encode topological/structural information, only multi-resolution approaches can be adopted. For statistical based descriptors (section 2.1) this is achieved by selecting different sampling of the model information, if the sampling is unifom, the coarser the sampling is, the coarser and concise the shape reperesentation is. Multi-resolution approaches of descriptors based on spherical harmonic analysis (section 2.2.2) can be obtained varying the resolution of the of the set of radii of the spheres used to analyze the shape object. Similar considerations hold for the set of views (section 2.2.4) and for the weighted point set (section 2.2.3). The former can vary the number of views captured from the object and the latter can vary the resolution of the grid used to analyze the object. Among the shape descriptors discussed in this chapter only coarse filter (section 2.2.1) does not allow multi-resolution.

Due to the **meaning of the informatin stored**, many methods discussed in this chapter allow the definition of similarity distance able to return as output a positive real number belonging to the range $[0, 1]$ measuring how much an object resembles to another one (sections 2.1.1, 2.2.1, 2.2.2, 2.2.3 and 2.2.4). All these descriptors are suitable for recognition and classification and apart the method involving spin-images, that generate a correspondence among oriented point between two different object models and not an estimation of the similarity between objects, they are also suitable for shape retrieval. Since the geometric descriptors do not axplicitly encode salient and relevant feature they are not suitable for tasks dealing with animation.

Chapter 3

Evaluation of Shape Similarity Based on Structural Shape Descriptors

For the generation of high-level descriptions it is useful to consider the shape as a combination of geometric and structural information [FS98b].

We can perceive the differences between geometry and structure by thinking about the object shape from an intuitive point of view. In this sense, geometric aspects of the object can be described by adjectives such as long, short, flat, rounded, squared, conic, cylindrical, spherical, etc. When the object is complex, other than geometry, it make sense to describe it in terms of its structural component as holes, protrusions and concavities. The concept of geometry is understood as a local property of the object, while structure deals with the whole object and concerns its global features and how they are related to each other. In accordance with the definition of structural descriptor given by Biederman [Bie95] and discussed in section 1.1.1, an object can be partitioned into protrusions, holes and can be efficiently represented as a collection of its structural features with a set of adjacency relations between them. These facts raise the idea that structural shape descriptors, maybe integrated with geometric information, are suitable for representing, manipulating and comparing shapes.

Usually the structural descriptor encode the shape of an object as a *s*keletal structure able to describe its salient features, in other cases describes the object as a graph. Generally such skeleton is centered in the object and summarizes its overall shape: skeletons may consist of a set of geometric primitives such as points, curves and polygons, where branches are associated to object protrusions and loops correspond to object holes.

The word “skeleton” recalls a support structure (i.e. the skeleton of a ship), or the scheme of something (the skeleton of an opera). Translating this concept into the digital context is not straightforward: in fact no general definition of skeleton exists in the computer graphics literature. In particular, this concept is naturally related with the structural properties and the application context of the object. Typical skeletons used in shape similarity context are one-dimensional structures represented as graphs (possibly direct and a-cyclic) whose nodes represent the shape features and edges the connection among them.

Since in the geometric modeling context there is not a general meaning of the notion of skeleton, its definition and representation depends on the choice of the application context. For example skeletal structures may provide an approximation of the input object able to produce a sketch of the original surface whose quality depends on the stored information and on the underlying recovering algorithm (i.e. implicit surfaces based on convolution [BL99, BBB^{+}97] or radial basis functions [TO02], direct methods [AC02]). In implicit modeling, skeletons are considered as a collection of elements with associated implicit primitives and provide a compact description that is useful in defining both motion and deformation.

In this chapter a short description of the most important structural descriptors is provided, especially focused on application contexts dealing with the evaluation of similarity among 2D and mainly 3D objects. A detailed survey on the mathematical background of structural shape descriptors can be found in [Bia04].

3.1 Survey on Structural Descriptors for 2D Shapes

In the past, two-dimensional objects (i.e. digital photographs or 2D silhouettes) has been the digital models mainly addressed by computer graphic. Lot of tools has been developed for application context dealing with shape morphing, recognition and similarity. Among these tools structural shape descriptors and methodologies for their comparison has been investigated.

In this section two of the most used structural descriptors for two-dimensional objects are shown and discussed.

3.1.1 Medial Axis 2D

Among the existing techniques for extracting the structure of a two-dimensional shape, the Medial Axis (MA) is generally considered the most elegant and effective one. An intuitive definition of the skeleton in the continuum was given by Blum [Blu67], who described the skeleton extraction as a fire front which starts at the boundary of the 2D model and propagates isotropically toward the interior. The medial axis is defined by the locations at which the fire fronts collide.

More formally, the medial axis of a shape S in R^n,$n \geq 2$, is the locus of centres of all maximal discs of S, that is, those discs contained in S which are not contained in any other disc in S (see figure 3.1(a)). Equivalently, if $B(S)$ is the boundary of S, then the medial axis of $B(S)$ is the set of points in S having at least two nearest neighbors on $B(S)$. In practice, we can associate to every point x of the interior of S its distance from the set $B(S)$, $d(x, B(S))$, defined as:

$$d(x, B(S)) = inf\{d(x, y) \mid y \in B(S)\}$$

However, there are some points where the distance is not achieved uniquely; for such points x, at least two boundary points (y and z) can be found such that $d(x, B(S)) = d(x, y) = d(x, z)$. These singularity points x define the nodes of the graph which correspond either to areas where the shape branches or end parts of protrusion-like structures (figure 3.1(b)).

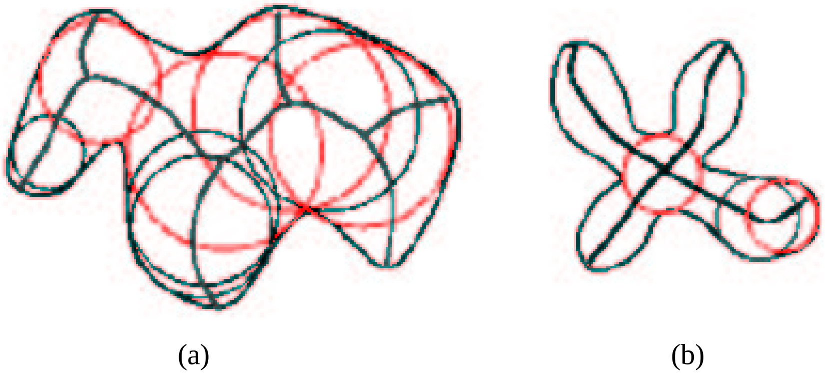


Figure 3.1: *Medial axis transform for 2D objects. The figure is from [BMMP03].*

The medial axis, together with the radius function, i.e. the distance from each point on the axis to the nearest point on the boundary, defines the *M*edial Axis Transform (MAT). The power of this representation is that the shape boundary and its MAT are equivalent and the one can be computed from the other [BS91] (the original shape can be recovered from its medial axis using a simple distance transform);

A two-dimensional object is effectively compressed into a one-dimensional graph-like structure. If the shape is a polygon, the MAT is a graph-like planar graph whose edges are composed by straight-line segments; several algorithms have been defined to compute the MAT from the Voronoi diagram of the polygon elements in $O(n \log n)$ operations, where n is the number of edges in the polygon [SPB96]. Finally, if the shape is simply-connected, then its medial axis is a graph, while cycles appear in the MAT around each hole in the shape.

It is known that this representation is independent on the object position in space (invariance), but it has the negative side that tiny perturbations of the boundary produce an extra edge in the graph with no distinction between main and secondary features (see figure 3.2(a)(b)). This fact raises the problem of graph pruning and makes the medial axis hard to be used in shape matching and recognition contexts.

Furthermore, the medial axis of a 3D shape is more complex and contains not only lines but also surface elements (see figure 3.3) and it is more expensive to compute, $(O(n2 \log n))$.

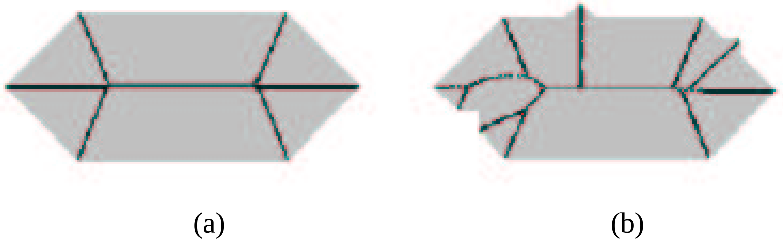


Figure 3.2: *A polygon and its medial axis shape descriptor (a). Perturbation of the boundary produce extra-edges (b). The figure is from [BMMP03].*

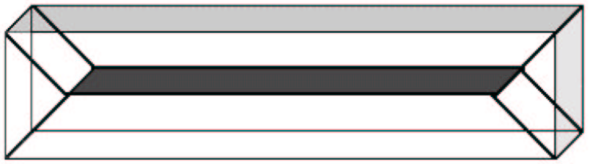


Figure 3.3: *. The figure is from [BMMP03].*

3.1.2 Shock graphs

Another medial-like representation of a shape is provided by the shock graphs [KTZ95, SSDZ98], where the *shock set* of a shape is a dynamic view of the MA which associates a direction and a speed of flow to the fire front propagation [GK03]. In [KTZ95] the consequence of slightly deforming of simple closed curves in the plane has been studied in order to obtain a coordinate system independent description of the 2D shape.

Shocks are generated in the course of the evolution of shape as singularities of the deformation process. Therefore they differ for the interpretation of the structure entities rather than their geometric representation. In fact, if the shape is a curve, the shock graph structure associates to each edge the growing direction of the radius of the bi-tangent spheres: so the resulting graph differs from the MA by edge orientations, see figure 3.4(b).

Four types of shocks may happen:

- **First-order shock**: is a discontinuity in orientation of the shape boundary;
- **Second-order shock**: is formed when two distinct non-neighboring boundary points join and not all the other neighboring boundary points have collapsed together;
- **Third-order shock**: is originated when two distinct non-neighboring boundary points join such that neighboring boundaries of each point also collapse together;

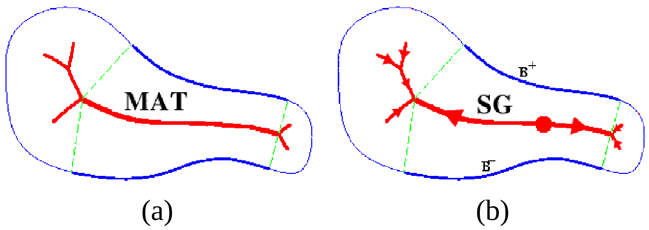


Figure 3.4: *The medial axis (a) and the shock graph (b) of two simple curves.*

- **Fourth-order shock**: is formed when a closed boundary collapses to a single point.

It has been shown that second-order and fourth-order shocks are isolated points while first-order and third-order shocks are neighbored by other shocks of the same type [KTZ95].

To build the shock graph, shocks of the same type that form a connected component are grouped and classified according to the behavior of the radius function around shock points. Then, shock points may be classified according to the number of contact points and the flow direction, as shown in [GK00]. Shock points fall in the following classes:

1. end points corresponding to a single contact point having inward flow;
2. interior points with two contact points and outward flow;
3. junction points corresponding to branches with only one outward flowing branch;
4. interior points with two contact points and inward flow;
5. junction points corresponding to branches all of which are inward flowing;
6. interior points with two contact points having both an outward and an inward flowing direction.

These types of shock points form a directed, planar graph, which is referred as the *shock graph*. In particular, shock points in 1, 2 and 3 are classified *sources* points, while these in 4 and 5 are *sink* points. Sources and sinks determine the nodes of the graph. Finally, points in 6 connect source points to sink ones and define the *edges* of the graph. In addition to these points, attributes are associated to the shock graph representation to store both the intrinsic geometry of the portion of shape corresponding to an edge and the radius and the flow direction of each node. Analogously to the MA, the shock graph structure and the corresponding point classification have been extended to $3D$ shapes [GK03]. However, in this case the shock graph structure contains non homogeneous data and it is not a planar graph.

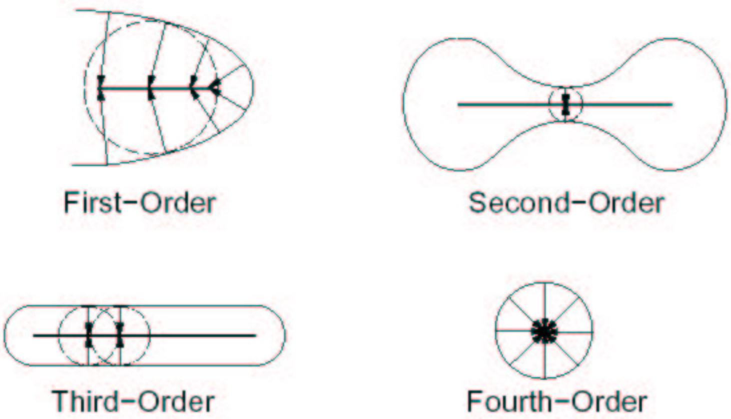


Figure 3.5: *Graphical representation of the different types of shocks. The figure is from [SSDZ98].*

3.2 Survey on Structural Descriptors for 3D Shapes

This section is a survey of the most meaningful 3D structural shape descriptors. The survey is not exhaustive, but the presented methodologies are representative of a large set of structural descriptors.

3.2.1 Volumetric Thinning

The method proposed in [GS99, SSGD03] utilize a parameter-based thinning algorithm from a volumetric representation of the 3D object, in order to extract its skeletal structure. The thinning method computes the distance transform at an object voxel as the minimum distance from the voxel to the boundary of the volumetric object. The distance transform value (DT) of a voxel can be interpreted as the radius of the sphere tangential to the boundary of the object and centered at that voxel. Fill in the sphere, the part of the object touching the boundary can be reconstructed.

The proposed method uses a heuristic which compares the distance transform at a voxel with that of its 26-neighbors. If the distance transform at a voxel is much greater than those of its 26-neighbors, the sphere centered at that voxel is likely to include most or all of the spheres centered at its neighboring voxels, this difference represents the importance of the voxel for the boundary coverage. A thinness parameter determines how much larger such a sphere should be for the centre voxel to be considered important for boundary coverage. For every voxel p, the mean of the distance transform of its 26-neighbors is computed ($MNTp$). The thinness parameter TP is then the difference of the distance transform at voxel p (DTp), and $MNTp$. Thus, for every voxel p in the object the thinness parameter

TP is defined as:

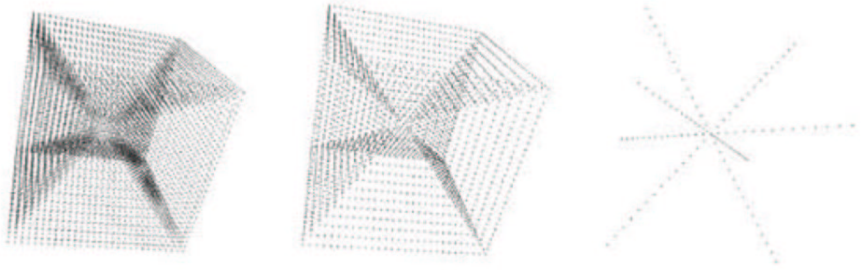
$$TP = DTp - MNTp \tag{3.1}$$

where:

$$MNTp = \frac{\sum_{i=1}^{26} DT_{qi}}{26}$$

with qi a voxel belonging to the 26-neighbors of p.

A low value of TP indicates that p is important for boundary coverage, since as TP decreases, the number of spheres covering the object increase and thus the representation of the object as set of spheres in more accurate. On the other hand, a higher thinness parameter TP provides a larger bounding volume but fewer spheres. The thinness parameter allows the volumetric object representation at several levels of detail, sorting the voxels in decreasing order with respect to TP. Object representation at different level of detail can be achieved extracting the voxels from the sorted list. When a low TP is used, that is those voxels that are not much greater than their neighbors, the points lie on the medial surface planes. As the TP is increased, the points left are those that lie along the medial axis of the individual planes. See figure 3.6



(a) (b) (c)

Figure 3.6: *The volumetric representation of a cube (a). The cube model is thinned into a set of points lying on the medial-axes planes (b). The one-dimensional skeleton along the medial-axes planes of the cube (c). The figure is from [SSGD03].*

In order to make the shape descriptor robust with respect to small perturbation of the object surface, a clustering process among the thinned voxel has been proposed. Let C_a be a cluster of thinned points, a point $p(x, y, z)$ is added to the cluster if the following condition is satisfied:

$$d(p(x, y, z), p(i, j, k)) < D_{th} \wedge NP(p(x, y, z), p(i, j, k))$$

where d is the distance between two points, D_{th} is a threshold, $p(i, j, k) \in C_a$ and NP assures that cluster across surface boundaries does not happens.

After thinning and clustering, the skeletal points are unconnected and an algorithm based on the minimum spanning tree with all the edges weighted proportional to their lengths, is used to transform the skeleton in a directed a-cyclic graph, where each edge is directed from the voxel with the higher distance transform to the one with lower distance transform. See figure 3.7.

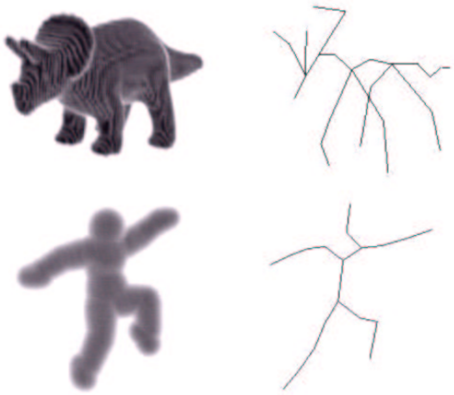
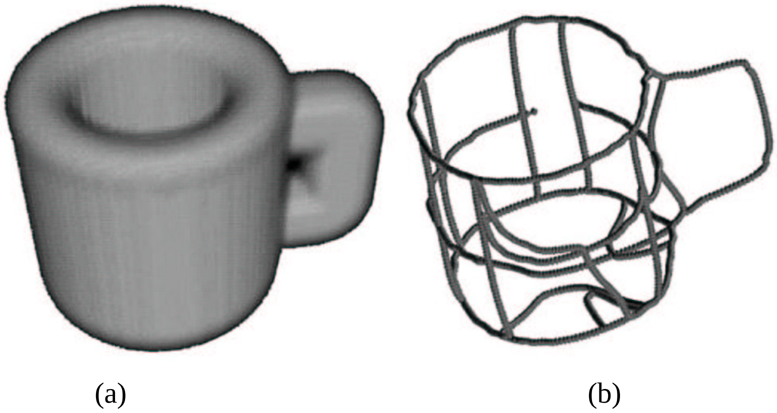


Figure 3.7: *Two models and theirs skeletons. The figure is from [SSGD03].*

The main drawback of these structures is that the resulting curve skeletons might not preserve the object topology and, even, lose the connectedness of the descriptor. Thus, the resulting skeletal graph representation could have an arbitrary number of cycles, independently of the object handles. For example, in figure 3.8[1] is shown the model of a mug (3.8(a)) and its skeleton obtained from a volumetric thinning technique based on the potential field erosion (3.8(b)), proposed in [SSC03]. In this example, many edges belonging to the body of the mug are not meaningful.



(a) (b)

Figure 3.8: *The model of a mug (a) and its curve skeleton (b).*

[1] The figure is from http://www.caip.rutgers.edu/vizlab.html

3.2.2 Scale-Space Representation

In [BSRS03, BSRS04] has been proposed a method to transform a 3D CAD model into a rooted tree encoding important information about both local features as well as global structure of the model. The basic idea of the scale-space decomposition is that a real object is made of meaningful features and the importance of such features depend on the scale of observation of the model.

The extraction of the features from the model can be formalized as in the following. Let M be a model, it can be partitioned into k features $M_1, \ldots, M_k$, such that $M_i \cap M_j = \phi$ for $1 \leq i < j \leq k$, where each feature M_i is defined as the set of surface point of the model such that maximize the function $f(M_i)$ defined on the geodesic distance among the points of M. In a finer scale, each feature M_i will be recursively decomposed into $j = 1, \ldots, k_i$ sub-features, subject to maximization of f. The depth of decomposition will be controlled depending on the quality of a feature in comparison to all its sub-features. Let A be a feature at scale i, and $A1, \ldots, Aj$, its sub-features at scale $i+1$, then the decomposition process should proceed to scale $i+1$ with respect to A if and only if $f(A) \leq f(A1) + f(A2) + \ldots + f(Aj)$. This simple criteria is motivated by the entropy of the feature A with respect to its sub-features $A1, \ldots, Aj$. Figure 3.9 is a representation of the recursive decomposition of a mechanical part into a rooted tree. The red regions of child nodes represent partitions that result in the bisection of the parent due to the use of geodesic distance definition of the function f.

The nodes set of the structural descriptor corresponds to the set of features recursively produced through the decomposition process and its edges capture the decomposition relation between a feature and all its sub-features.

3.2.3 Reeb graphs

Differential topology suggests another approach to structural shape description which mainly relates to Morse theory [GP74, Mil63]. Given an object surface S and a Morse function[2] $f : S \rightarrow \Re$, Morse theory states that the shape of the pair (S, f) is represented by the evolution of the homology groups[3] of the level sets[4]:

$$S^x = f^{-1}(x)$$

where x varies on $\Re$ and $S^x \subseteq S$ [Mas67]. Since homology groups code shape properties as the number of connected components, holes and cavities in the object, it follows that a finite collection of level sets is sufficient to fully describe the surface shape, see figure 3.10(a).

Focusing on the level set evolution, we obtain a discrete description which effectively represents the shape of S and can be encoded in a topological graph, as formalized by the following definition

[2] The function f is called a *Morse function* if all of its critical points are non-degenerate.

[3] The *homology group* is an Abelian group which counts the number of holes in a topological space. In particular, homology groups define a measure of the hole structure of a space [Mas67, Spa66, Hat01].

[4] The pre-image through f of a constant value t in the domain of f is called *level set* or, also, *contour* or *isolevel*. Level sets may be non-connected.

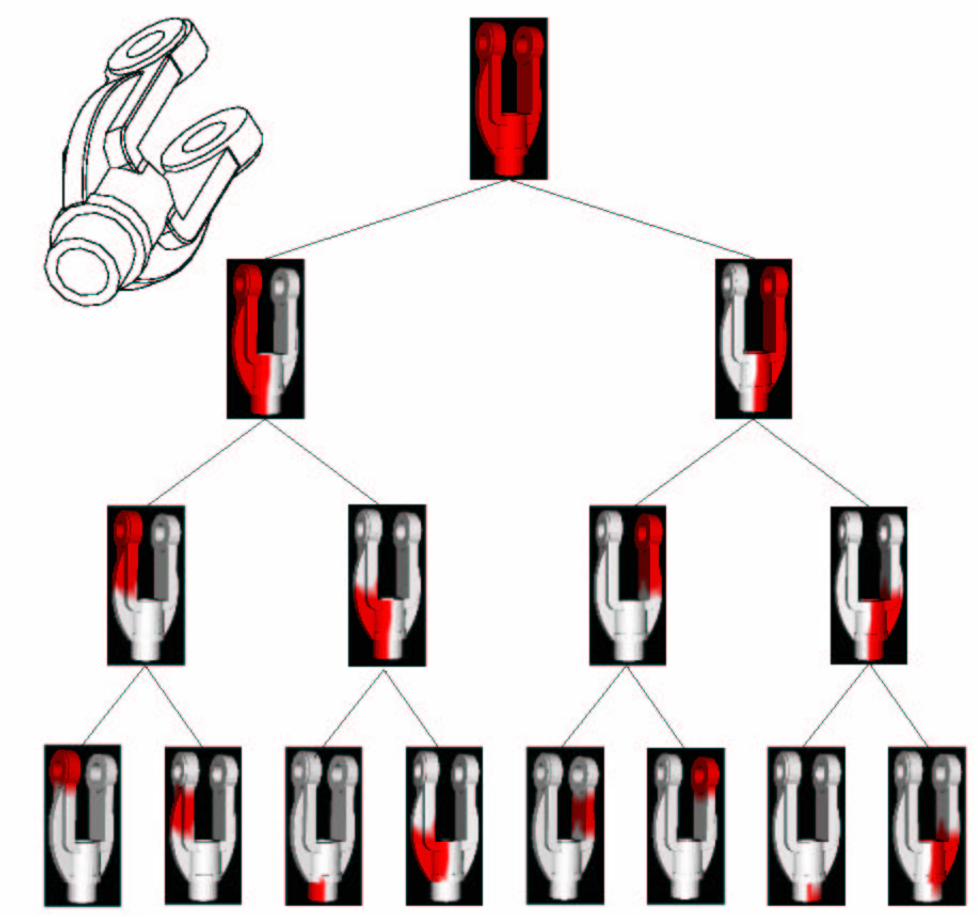


Figure 3.9: *A mechanical part and its scale-space decomposition. The figure is from [BSRS03].*

[FK97, Ree46].

Definition 3.2.1 (Reeb graph) *Let $f : S \to \Re$ be a real valued function on a compact manifold S. The* Reeb Graph *G of S with respect to f is the quotient space of $S \times \Re$ defined by the equivalence relation $\sim$, given by:*

$$(P, f(P)) \sim (Q, f(Q)) \quad \text{iff} \quad f(P) = f(Q) \quad \textit{and} \quad P, Q \in f^{-1}(f(P))$$

where $P, Q \in S$.

Mathematically, the Reeb graph is defined as the quotient space of S with respect to the value of the function f. Moreover, the critical points of f, that is the points of S where the gradient of f vanishes, correspond to topological changes of the Reeb graph of S, determining its homology groups and how cells, which correspond to critical points, glue in the graph (see figure 3.10(b)).

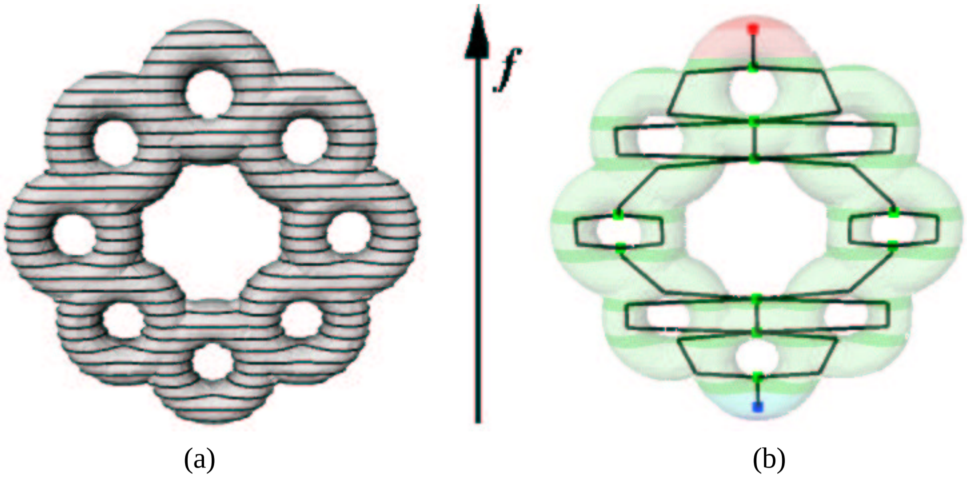


Figure 3.10: *An 8-torus model, where the level sets corresponding to a function f has been highlighted (a). The corresponding Reeb graph, where red nodes represent the maximum critical points of the function f, green nodes represent the saddle critical points, finally blue nodes represent the minima critical points. The figure is from [BMMP03].*

Therefore, the Reeb graph of S collapses into one element all points having the same value under the real function f and being in the same connected component. In practice, it contracts at a point each connected component of the level sets. Even if in definition 3.2.1 there are not hypotheses on the differentiability of the function f, requiring the function f is at least continuous is reasonable for applications and theoretical results are available. In particular, if the function f is at least C^2, Morse theory states that the topological changes of the level sets occur only in correspondence of critical points.

The Reeb quotient space can be effectively represented as a graph structure as shown in figure 3.10(b): a node is defined for each critical level of f, which corresponds to the creation, merging, split or deletion of a contour, that is, to topological changes affecting the number of connected components in the counter-image of f. Each edge joins two successive critical levels in their own component. If an edge connects two nodes, n_1 and n_2, then the topology of isolevels on S between the critical levels n_1 and n_2 does not change along the connected component of S joining the corresponding critical points. Then, the Reeb graph of a surface S codes the shape of S in terms of the critical points of f, structuring them into a topologically consistent framework.

Since the properties of S and f determine those of G, the quotient function f has to be chosen in order to extract characteristics which fully describe the object with respect to the application needs. From a topological point of view, the continuity of f ensures that two homeomorphic manifolds are mapped into homeomorphic graphs thus guaranteeing their identification. Even if previous considerations enable to fully identify topological properties of S through G, different application fields, such as matching, compression, etc., require to select, among different but equivalent representations of S, that more suitable for the algorithm which uses it as input. For instance, matching requires an input graph with a minimal number of redundant leaves in order to avoid a preprocessing for pruning.

In addition, the Reeb graph requires the storage of an amount of geometric information for reconstruction tasks, [BMS00]. Therefore, according to the application domain, the choice of the most appropriate function f varies and should enable the extraction of some geometric information on S, such as section length, area, etc.... The family of continuous or Morse functions is a natural set for identifying f, even if the choice of a candidate faces with computational constraints.

3.2.3.1 Reeb graph with respect to the height function

A standard choice of f is the height function in the three-dimensional space (see figure 3.11), which has extensively been studied in [SKK91, SK91, BFS00, BFS02, ABS03]. The equivalence classes induced by the height function correspond to the intersection of the mesh with a set of planes that are orthogonal to a given direction. In this case, the critical points correspond to peaks, pits and passes in their usual meaning; the generated graph very intuitively resembles the skeleton of the object silhouette but it depends on the chosen direction of the height function, so that different orientations may produce different results.

The use of the height function is best suited for digital terrain models, where to associate to each point of the model its elevation is a natural choice and, as previously described the introduction of a global virtual minimum provides a unique interpretation of the surface boundary. Methods for extracting the Reeb graph with respect to the height function of a closed surface have been proposed in [SKK91, ABS03]. Although these methods have different input requirements and computational costs, the properties of the extracted graph are comparable; in particular, Morse theory guarantees that, with a suitable object slicing, the number of cycles in the graph corresponds to the number of holes of S but does not answer to the problem of calculating an optimal slicing of the model. As shown in [ABS03], the choice of the slicing thickness is critical and the density of slices determines the scale of the shape features detected. An extended discussion on the best slicing approach is proposed in chapter **??**, where the complete framework for the Reeb graph extraction from triangular meshes is proposed. However, the topological correctness of the graph is guaranteed by an adaptive slicing process that refines the thickness of contour levels in correspondence of holes.

The main property of the Reeb graph calculated with respect to such a f is the independence from translations and uniform scaling. Moreover, the evaluation of the function f is immediate and the graph is a very powerful tool for understanding the shape in contexts where surfaces have a naturally privileged direction, as terrain models, [BFS02, CSA03, TIS^{+}95]. Furthermore, the dependence of the height function on rotations, makes this graph unsuitable for matching and classifying 3D shapes unless shapes are somehow coherently oriented, for example using the principal component analysis (PCA) method proposed in [VSR01, Vra04].

Finally, starting from the sections that are the boundary components of the critical regions of f and following the connectivity relationship coded in the Reeb graph, a topologically consistent framework for surface reconstruction has been proposed in [BMS00].

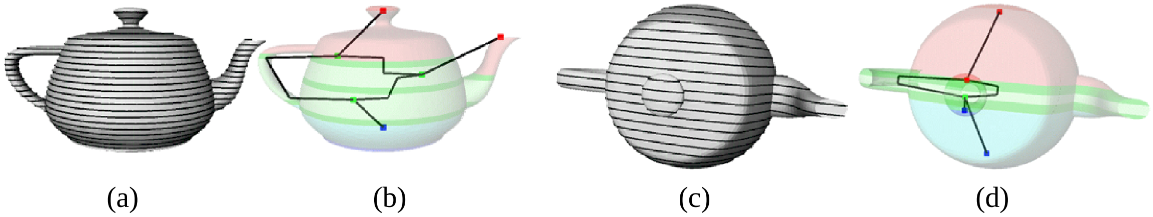


Figure 3.11: *Reeb graph representation with respect to the eight function (b) and (d) of the same object in to two different position (a) and (c).*

3.2.3.2 Reeb graph with respect to a distance function

Differential topology suggests another class of Morse maps: the distance functions of the surface points from a given point p of the Euclidean space. As demonstrated in [FK97], a distance point is Morse if and only if the point p is not a focal point. Such a point could belong to the mesh or not, even though a reasonable choice is the centre of mass or barycentre of the object which is easily calculated and, due to its linear dependence on all vertexes is stable to small perturbations. Then, the evaluation of the function on the mesh vertexes is straightforward: in fact, at each vertex v of the mesh, the Euclidean distance between v and p is associated.

According to the criteria proposed in [ABS03], the isocontours of the distance map on S can be detected by interpolating the values of f (for example, see figure 3.13). The contour levels, which are computed for the height function, induce a characterization of the regions of the mesh, which is done by comparing the value and the number of each boundary components. Depending on this choice of f, the interpretation of maximum and minimum areas differs from those obtained with the height function: in fact, the isocontours correspond to the intersection of the mesh with a collection of spheres centered in the barycentre and with different radii. In other words, it is possible to recognize a set of protrusions and hollows of the mesh with respect to the barycentre by analyzing the maxima and the minima of the function f that have only one boundary component. An example of the different behavior of minima is shown in figure 3.12(b): the minima corresponding to the palm and the back of the hand represent two concavities while minima on the ring and little finger locate two surface protrusions. Figures 3.12(a,c) represent, respectively, the contour levels of f on such a hand model and the resulting Reeb graph representation. Because the barycentre and the sphere/mesh intersection are independent on translation, rotation and uniform scaling of the object, these properties are reflected on the resulting Reeb graph. Another example of Reeb graph calculated for a tea pot model with respect to the barycentre distance in depicted in figure 3.13.

The Reeb graph representation with respect to the distance from the center of mass is suitable for those application fields where the spatial distribution of the shape is important. In particular, in [BMM$^+$03] such a choice of the function f has been considered for shape matching purposes.

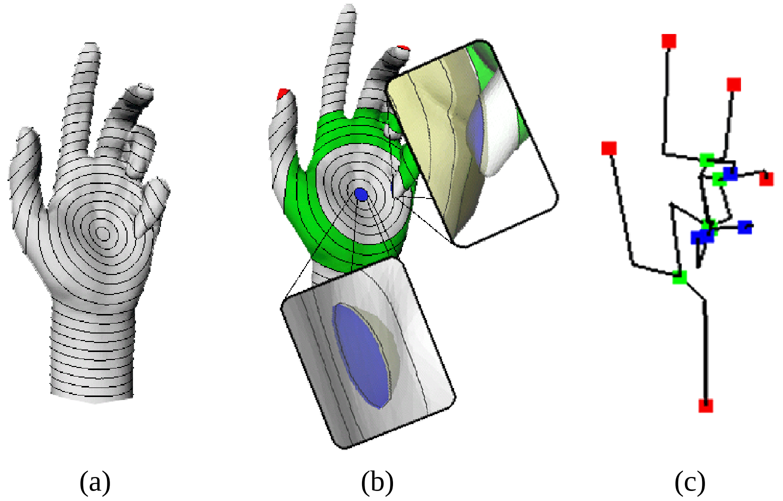


Figure 3.12: *Level sets of the distance from the barycentre of a hand model (a), the interpretation of minima as protrusions or concavities (b) and its Reeb graph representation.*

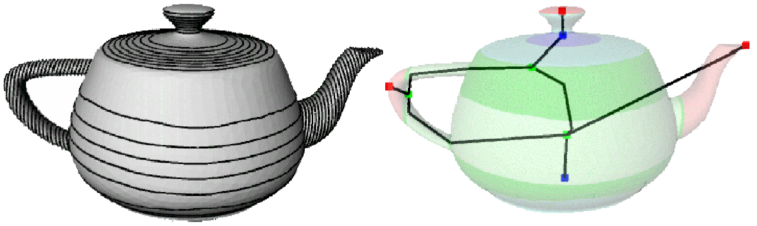


Figure 3.13: *Level sets of the distance from the barycentre of a pot and its Reeb graph representation.*

3.2.3.3 Reeb graph with respect to the integral geodesic distance

A different mapping function has been defined by Hilaga et al. [HSKK01], where the notion of an integral geodesic distance has been introduced for matching purposes. In particular, for each vertex on a triangulation S, the value of the function f is given by:

$$f(v) = \int_{p \in S} g(v,p) dS$$

where $g(v,p)$ represents the geodesic distance between v and p, when p varies on S. This function is not invariant to scaling of the object and thus it is replaced by its normal representation defined as:

$$f(v) = \sum_i g(v,b_i) \cdot area(b_i)$$

where $b_i = b_0, \ldots, b_k$ are the base vertexes for the Dijkstra's algorithm which are scattered almost equally on the surface and $area(b_i)$ is the area of the neighborhood of b_i.

The resulting Reeb graph is theoretically invariant with respect to rotation, translation and uniform scaling. Even if the surface curvature and the geodesic distance are good shape descriptors, their dependence on second order derivatives makes them numerically unstable, preventing their direct use for the graph definition. Due to the time complexity of the exact evaluation of the function f, an approximation based on the Dijkstra's algorithm has been proposed. Unfortunately, this choice does not guarantee the absolute independence of its values from the object orientation and it is computationally expensive. Examples of evaluation of the function f are depicted in figure 3.14 where the darkest regions represent the regions on the model where the topology of contour levels changes; finally, the graph structure of a pot is shown in figure 3.15.

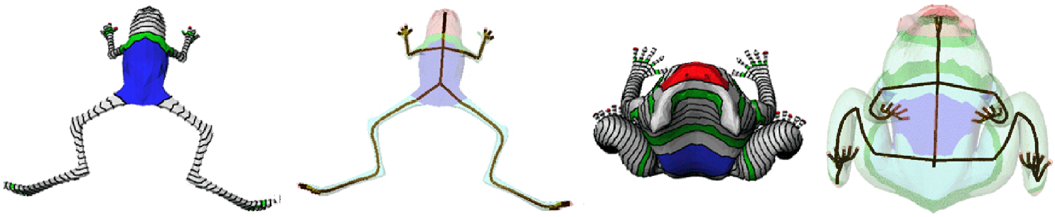


Figure 3.14: *Isocontours of the function in [HSKK01] on the model of two frog in different positions.*

3.2.3.4 Centerlines based on discrete geodesic distance from a source point

For representing three dimensional polyhedral objects a first approach which deals with the construction of centerlines from unorganized point sets has been presented in [VL00], and later developed for polyhedral objects in [LV99].

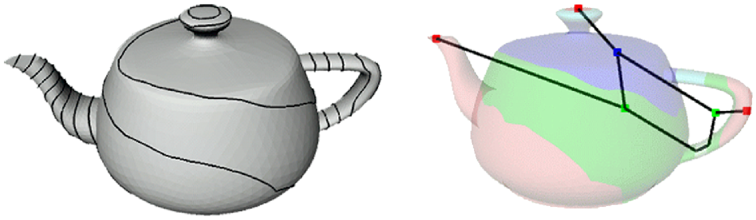


Figure 3.15: *The Reeb graph representation of a pot with respect to the function in [HSKK01].*

With reference to [VL00], a skeleton-like structure, available for triangular meshes homeomorphic to a sphere, is proposed, which is essentially a tree made of the "average point" associated to the connected components of the level sets of a given function; in particular, the geodesic distance from a source point is chosen, as shown in figure 3.16(a).

To automatically select the source point an heuristic is used, which seems to work well on elongate tubular shapes. In this case, skeletal lines obtained with different source points are very similar and the resulting skeleton is invariant under rotation, translation and scaling. Anyway, the choice of only one source point determines a privileged "slicing direction", which can lead to the loss of some features if the object is not tubular shaped (like the horse ears in figure 3.16(b)).

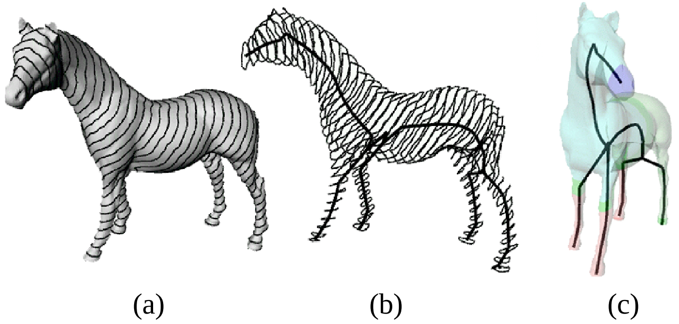


Figure 3.16: *Isolevels (a) and the centerline (b,c) of the horse as computed in [LV99].*

An extension of the previous approach to non-zero genus surfaces has been presented by Wood et al. [WDSB00]. There, the graph is implicitly stored for generating high quality semi-regular multi-resolution meshes from distance volumes. Also in this case, the object topology is achieved by considering a wavefront-like propagation from a seed point, [AE98] (see figure 3.17). The calculation of the isosurfaces is obtained by applying the Dijkstra's algorithm. This makes the approach unavailable for non-uniform scaling. Afterward, the framework deriving from this Reeb graph representation has been effectively used as starting point for the detection of topologically correct quadrangulations on the surface [HA03].

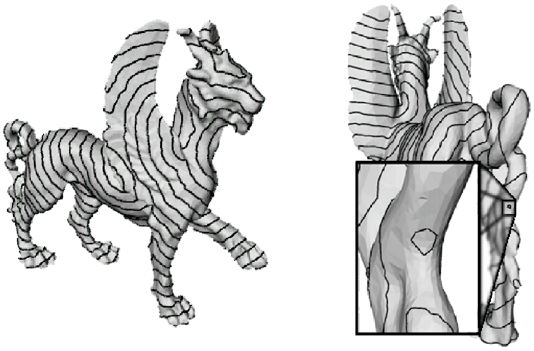


Figure 3.17: *The wave-front propagation in [WDSB00].*

3.2.3.5 Reeb graph based on topological distance from curvature extrema

The strategy proposed in [MP02] extracts the skeleton of a surface represented by a simplicial complex and combines differential and computational topology techniques.

In this context, a multi-resolution curvature evaluation, [MPS$^+$03] is introduced to locate seed points which are sequentially linked by using the natural topological distance on the simplicial complex (see figure 3.18(a,b)).

More precisely, once computed the approximated Gaussian curvature for the mesh vertexes, for each high curvature region R_i, $i = 1, \ldots, n$, a *representative vertex* p_i is selected. Starting at the same time from all representative vertexes, rings made of vertexes of increasing neighborhoods are computed in parallel until the whole surface is covered (see figure 3.18(c)), in a way similar to the wave-traversal technique [AE98]. Rings growing from different seed points will collide and join where two distinct protrusions depart, thus identifying a branching zone; self-intersecting rings can appear expanding near handles and through holes.

A skeleton is drawn according to the ring expansion: *terminal nodes* are identified by the *representative vertexes*, while union or split of topological rings give *branching nodes*. Edges are drawn joining the center of mass of all rings (see figure 3.18(d)).

The function

$$f_{p_i}(x) = \min\{k \mid x \in k - neighborhoods\}$$

defines the topological distance of x from p_i and can be extended to a finite set of vertexes $\{p_1, \ldots, p_n\}$ as

$$f(x) = \min_{k=1,\ldots,n} \{f_{p_k}(x)\}, \forall x \in M$$

that is, f assigns to x its minimal topological distance with respect to more than one vertex. Given M and f it follows that $G = M/\sim$ is a Reeb graph with respect to such a function f.

The complexity of the proposed graph, in terms of number of nodes and branches, depends on the

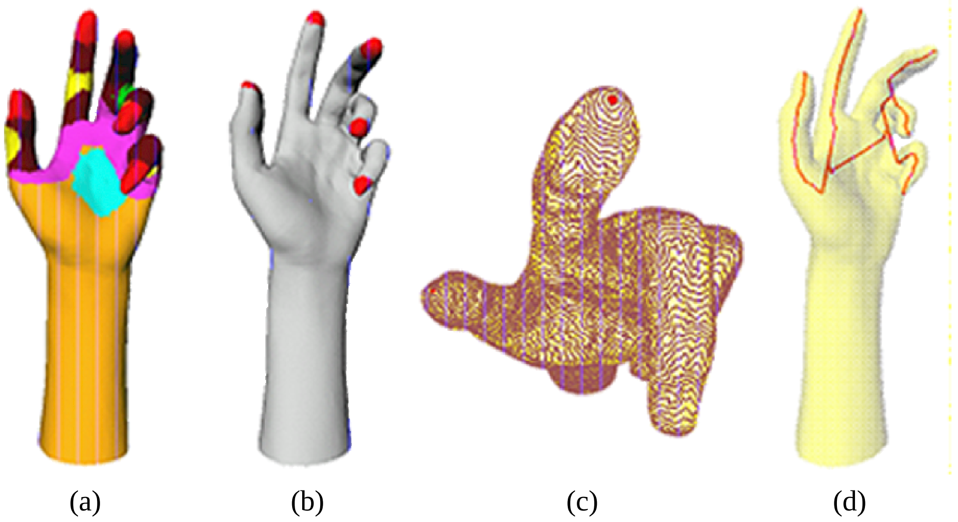


Figure 3.18: *(a) Vertex classification based on Gaussian curvature, (b) high curvature regions are depicted in red; (c) topological rings expanded from centers of high curvature regions (d) obtained graph.*

shape of the input object and on the number of seed points which have been selected using the curvature estimation criterion. This graph is rigid-invariant (translation, rotation, scaling) because the chosen function f does not rely either on a local coordinate system or on surface embeddings as it happens, for example, using the height function. On the other hand, if the curvature evaluation process does not recognize at least one feature region, e.g. surfaces with constant curvature value as spheres, this approach is not meaningful for extracting a description of the shape; on the contrary, the height function always guarantees to get a result.

Since such a function is continuous, the Reeb graph representation has at least as many cycles as the number of holes of the surface, however, some unforeseen cycles may appear in correspondence of the wavefront collisions. Nevertheless experimental results have shown that this framework works on shapes of arbitrary genus.

3.3 Survey on Methods for Structural Shape Similarity

The structural shape descriptor is usually encoded as skeleton and when it is used in application tasks dealing with shape comparison, a graph is extracted from the skeletal structure. Therefore shape evaluation of similarity among 3D objects is achieved through graph comparison techniques.

Investigation of such techniques is a field of the computer science that is widely issued from the beginning of the second half of the last century, and since then, a lot of results has been achieved. The most used techniques to compare graphs are the graph isomorphism, the analysis of the clique of the association graph and the inexact graph/subgraph isomorphism. The last technique is the best suitable for object similarity. The graph isomorphism is a too strong condition to be satisfied, since it holds when two graph are identical and thus, also the two objects described by them are identical Instead, the clique analysis of the association graph and the inexact graph/subgraph isomorphism, provide information about how much two graphs resemble themselves. For these reasons they are suitable to evaluate the similarity among objects.

Since the computational costs of these problems is intrinsically exponential (NP-complete for the decision problem and NP-hard for the optimization problem) a lot of heuristic algorithms have been developed to approximate the optimal solution, especially for the inexact graph/subgraph isomorphism problem. These heuristics deal with assumptions made on the graph used as structural shape descriptors (i.e. directed, undirected, a-cycled rooted tree, etc), on the attributes associated to nodes and edges and finally with semantic information coming from the interpretation of the structural shape descriptor.

In the following, the most common inexact subgraph isomorphism techniques for the evaluation of the shape similarity among 3D objects will be shown and discussed.

In [SSDZ98] the shock graph discussed in section 3.1.2, are used as signature of 2D objects to compare their topological structure. In order to perform the graph matching, the directed shock graph is transformed into a rooted tree and then compared using a subtree isomorphism algorithm. This comparing methodology has the drawback to discard the information relative to the graph cycles, thus relative to the handles of the object.

The basic idea to transform a directed shock graph into a rooted tree, is to break its loops in accordance with a set of rules called shock grammar and shown in [SSDZ98]. As consequence of applying such rules, the node where two directed paths forming a loop meet, is duplicated breaking the loop, moreover, the application of such grammar to the tree produce rooted tree. The similarity measure between two input rooted tree $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is formalized as an optimization problem where the optimal solution is the $\{0, 1\}$ matrix M representing the bijective mapping among the nodes of two input rooted tree:

$$\begin{array}{ll} \min & -\frac{1}{2}\sum_{u\in V_1}\sum_{v\in V_2} M(u,v)\|u.v\| \\ s.t. & \sum_{u'\in V_2} M(u,u') \leq 1, \forall u \in V_1 \\ & \sum_{v'\in V_1} M(v,v') \leq 1, \forall v \in V_2 \\ & M(x,y) \in \{0,1\}, \forall x \in V_1, \forall y \in V_2 \end{array}$$

where $\|\cdot\|$ is the similarity value between the attributes associated to the node trees.

After the graph is transformed into a tree, its topological structure is captured. Let $G = (V, E)$ be the tree with $|V| = n$ nodes and $|E| = m$ edges. The adjacency matrix associated to G is the $n \times n$ matrix where each element $a_{ij} = 1$ if node i is connected to node j through an edge of E, $a_{ij} = 0$ otherwise. Let $\delta(v)$ be the degree of the node $v \in V$, and $\delta(G)$ the maximum degree of all the nodes of G. For every node u, $\chi(u)$ is the vector in $\Re^{\delta(G)-1}$ obtained through the following procedure:

1. for any child v of $u \in V$ construct the adjacency matrix A_v of the subtree of G induced by v.

2. extract $\lambda_v = \lambda_1(A_v) + \ldots + \lambda_{\delta(v)}(A_v)$ from A_v, where $\lambda_1(A_v), \ldots, \lambda_{\delta(v)}(A_v)$ are the first $\delta(v)$ eigenvalues of A_v

3. construct the vector $\chi(v) = (\lambda_{v_1}, \ldots, \lambda_{v_{\delta(u)}})$, where $\lambda_1 \geq \ldots \geq \lambda_{\delta(v)}$

This procedure yields a vector assigned to each node of the tree. Such vector encodes the topological structure of the node subgraph, approximated through the sum of its eigenvalues. The comparison process between two shock trees is obtained recursively finding matches between nodes, starting at the root of the tree and proceeding through the subtrees in depth-first fashion. The matching between nodes incorporates both geometric attributes associated to each node and the topological information encoded by the vector χ.

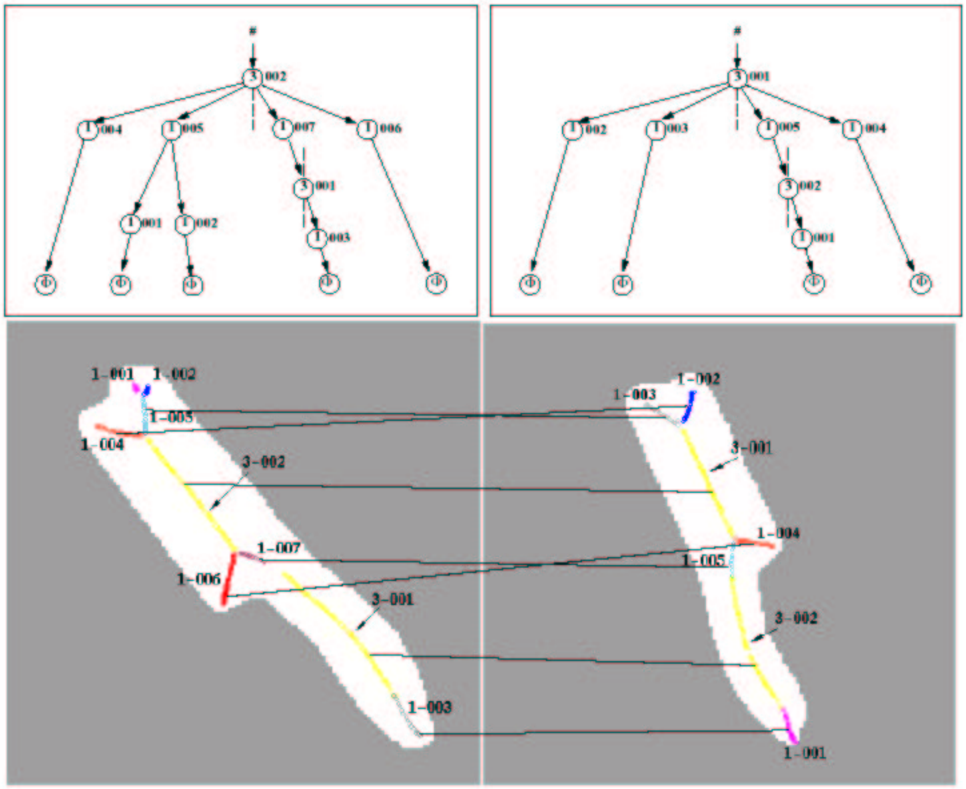
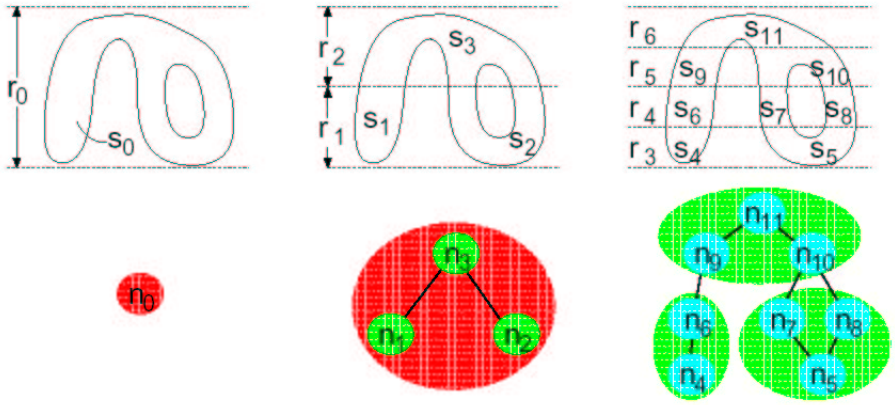


Figure 3.19: *. The figure is from [SSDZ98].*

Another approach to shock tree comparison has been proposed in [PSZ99], where subtree isomorphism is computed as maximal clique of the Tree Association Graph (TAG) associated to the two trees. The TAG is obtained through the definition of the path-string between two nodes of a tree, where a path-string $str(u, v)$ is a representation of the path between the two nodes u and v. Because of the orientation induced by the root, only two types of moves can be done: going down to one of the children (if one exists) or going up to the parent (if any). Then, the path-string of u and v is simply the string of elementary moves required to reach v, starting from u. The tree association graph (TAG) of two rooted trees $T_1 = (V_1, E_1)$ and $T_1 = (V_1, E_1)$ is the graph $G = (V, E)$ where $V = V_1 \times V_2$ and, for any two nodes $(u, w), (v, z) \in V$ such that (u, w) is connected to (v, z) through an edge of the TAG if and only if $str(u, v) = str(w, z)$. The extraction of the maximum clique of the TAG is formalized as a quadratic problem and dynamical systems developed in theoretical biology are used to solve it.

The similarity estimation between 3D objects proposed in [HSKK01] (section 3.2.3.3), uses a Multi-Resolution Graph (MRG) obtained through a coarse-to-fine representation of the Reeb graph based on the integral geodesic distance. Therefore a series of Reeb graphs at different level of detail are generated.

To build a Reeb graph of a certain level of detail, the object is partitioned into regions based on the function f. A node of the Reeb graph represents a connected component in each region, and adjacent nodes are linked by an edge if the corresponding connected components of the object contact each other. The Reeb graph for a finer level is constructed by repartitioning each region. In the method proposed in [HSKK01], the re-partitioning is done in a binary manner for simplicity. In figure 3.20(a) shows one region r_0 and one connected component s_0 and the graph is one node n_0 corresponding to s_0. In Figure 3.20(b) the region r_0 is partitioned to r_1 and r_2 and the nodes n_1, n_2 and n_3 correspond to the connected components s_1, s_2 and s_3. Finally, the finer level of detail is represented figure 3.20(c).



(a) (b) (c)

Figure 3.20: *Coarse to fine representation using the eight function. The picture is from [HSKK01].*

The comparison process between two MRG is obtained using a list of graph nodes $NLIST$ and a list of node pairs $MPAIR$, through the following steps:

1. **initialization**: Inserting the coarsest nodes of two MRGs R and S to $NLIST$.

2. **matching**: In $NLIST$, find a matching node pair ($m \in R$, $n \in S$) which preserves the topological consistency of the MRG.

3. **unpacking**: Remove m and n from $NLIST$ and insert (m, n) to $MPAIR$. Then, if not at the finest resolution, insert the child nodes of m and n to $NLIST$.

4. **loop**: If $NLIST$ is not empty, repeat step 2 and step 3. Otherwise, calculate the similarity value using $SIM(R, S)$ defined in equation (3.2).

where $SIM(R, S)$ is the similarity between two MRG and $\overline{n}$ and $\overline{m}$ are the attribute associated to the nodes of R and S. In the implementation proposed in [HSKK01], the attribute deals with geometric properties of the model.

$$SIM(R,S) = \sum_{(n,m)\in MPAIR} sim(\overline{n}, \overline{m}) \tag{3.2}$$

In figure 3.21 an example of matching between two MRG is shown. In figure 3.21(a), the two nodes m_0 and n_0 are inserted to $NLIST$. The nodes m_0 and n_0 represent the coarsest level in R and S respectively. In figure 3.21(b), (m_0, n_0) are matched and unpacked to their child nodes m_1, m_2, n_1 and n_2 which are inserted into $NLIST$. In the next iteration, (m_2, n_2) are matched and then unpacked as shown in figure 3.21(c), and then in a following iteration (m_4, n_5) are matched and unpacked as shown in figure 3.21(d). During the comparison process, the topological consistency of the object have to be maintained. For this reason two nodes can be matched only if they are in the same region (e.g. the regions r_1 and r_2 of figure 3.20(b)) and their parents are matched. Moreover matching of nodes belonging to different branches have to be avoided.

In [BRS03] the MRG has been used to demonstrate the e effectiveness of the MRG to the problem of the manufacture-model retrieval. Some experiments have been proposed to show the performance of the MRG technique on primitive CAD models, such as cubes and spheres, on more complex models, such as LEGO and mechanical parts, and finally, on complex CAD models. The results of such experiments are meaningful, nonetheless several problems arise from this technique. For example, the 3D model has to be two-manifold; furthermore, the comparison process may produce false positive results and it is more sensitive to the geometry of a model rather than to its topology. The constraint of using only two-manifold models could be relaxed, but problems dealing with the computational complexity and the shape representativeness of the graph may occur. Moreover in [TS04] the method proposed by Hilaga has been improved endowing the MRG with new geometric attributes of the object.

The scale-space decomposition described in section 3.2.2 produce a rooted tree where its vertex set corresponds to the set of features recursively obtained by decomposing the object model, and the edges capture the hierarchical relationship between features and sub-features. The problem of comparing two model is moved to the comparison of the two correspondent trees.

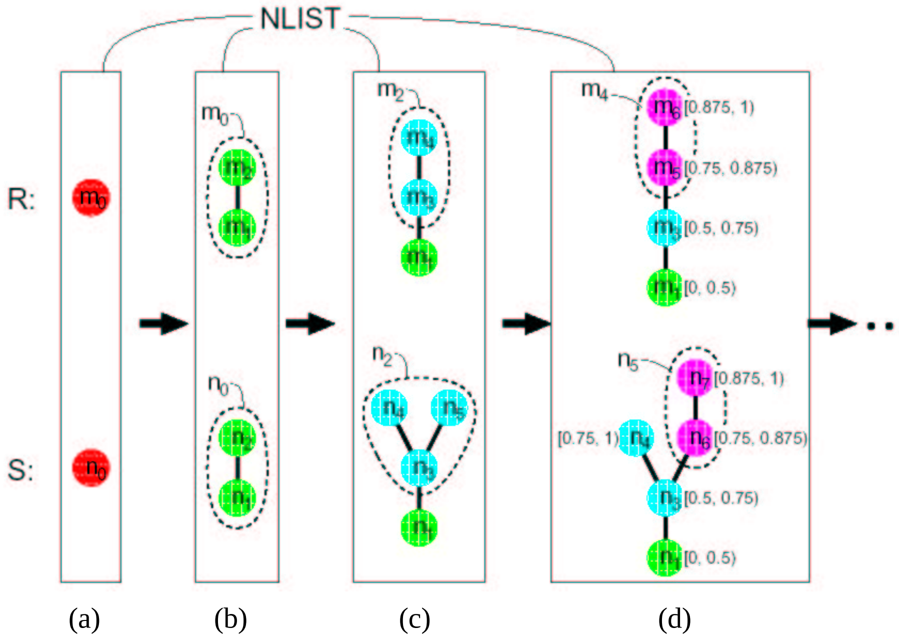


Figure 3.21: *An example of matching between two MRGs. The picture is from [HSKK01].*

The matching algorithm proposed in [BSRS03] is based on the dynamic programming framework defined in [WSS$^+$98], where the similarity between the two nodes u and v is measured in terms of the similarity between the subtrees $T_1(u)$ and $T_2(v)$ rooted at u and v respectively. The algorithm for the similarity estimation can be summarized with the following equations:

$$C(T_1(u), T_2(v)) = C(F_1(u), F_2(v)) + \gamma(u, v) \tag{3.3}$$

$$C(T_1(u), \phi) = \gamma(u, \phi) \tag{3.4}$$

$$C(F_1(u), \phi) = \sum_y \gamma(y, \phi) \tag{3.5}$$

Equation (3.3) formalize the cost matching between the two trees $T_1(u)$ and $T_2(v)$ as the cost matching between the two forests $F_1(u)$ and $F_2(v)$ obtained from the $T_1(u)$ and $T_2(v)$ after removing u and v respectively, while the difference between the two node u and v is represented by $\gamma(u, v)$. Degenerate cases are managed by equation (3.4) and (3.5), where the sum in the latter is over the roots of all trees in $F_1(u)$. The cost matching between two forests is:

$$C(F_1(u), F_2(v)) = \frac{1}{2}\left(\sum_x C(x, \phi) + \sum_y C(\phi, y)\right) + \sum_{(x,y)\in\mathcal{B}(u,v)} w(x, y)$$

where $\mathcal{B}(u, v)$ represents all the node pairs among the root nodes of the forests $F_1(u)$ and $F_2(v)$ and the function $w(x, y)$ represents the cost between such node pairs. Such cost is defined as:

$$w(x, y) = C(x, \phi) + C(\phi, y) + C(T_1(x), T_2(y))$$

Finally the function $\gamma(u, v)$ represents the cost between the geometric attributes describing the (sub)features associated to the nodes u and v.

An example of 3D object matching is shown in figure 3.22.

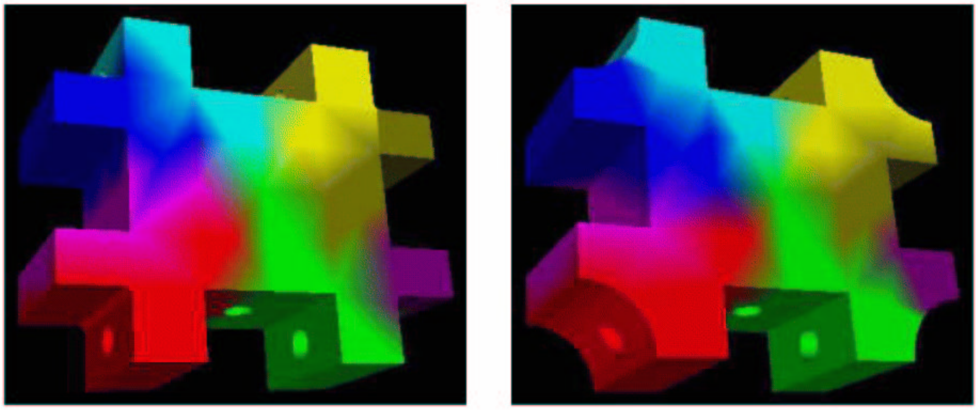


Figure 3.22: *The matching between two slightly different mechanical parts, where matched regions have the same colors. The picture is from [BSRS03].*

3.4 Discussion

Structural shape descriptors, as discussed in section 3, describe the shape of an object in terms of its meaningful parts, and adjacency relationship among them. Different shape descriptors capture different aspects of the shape, thus also the meaning of the described parts is different. For example, the Reeb graph with respect to a distance function (section 3.2.3.2) describe the shape object as a spatial distribution with respect to a given point of the space, while the shape aspects described by Reeb graph with respect to the integral geodesic distance (section 3.2.3.3) do not depend by any spatial reference, but only by the relative importance of its features with respect to the overall object. Moreover the Reeb graph based on topological distance from curvature extrema (section 3.2.3.5), captures the shape starting from a set of seed points. The different behavior of this structural descriptors is explained in figure 3.23, where a teapot model is shown together with its modified version obtained braking the teapot handle (figure 3.23(a)). The graph shown in figure 3.23(b) is a representation of the spatial distribution of the object with respect to the barycentre: even if a part of the handle has been removed the remaining part folds on itself, generating a critical points in the Reeb function. The graph based on the integral geodesic distance (figure 3.23(c)) does not take into account the spatial embedding, thus the broken handle of the teapot results in a maximum critical point with respect to the geodesic distance, neglecting the shape of the handle itself. Concerning the distance from the curvature extrema (figure 3.23(d)) the modification of the teapot handle results in a new curvature

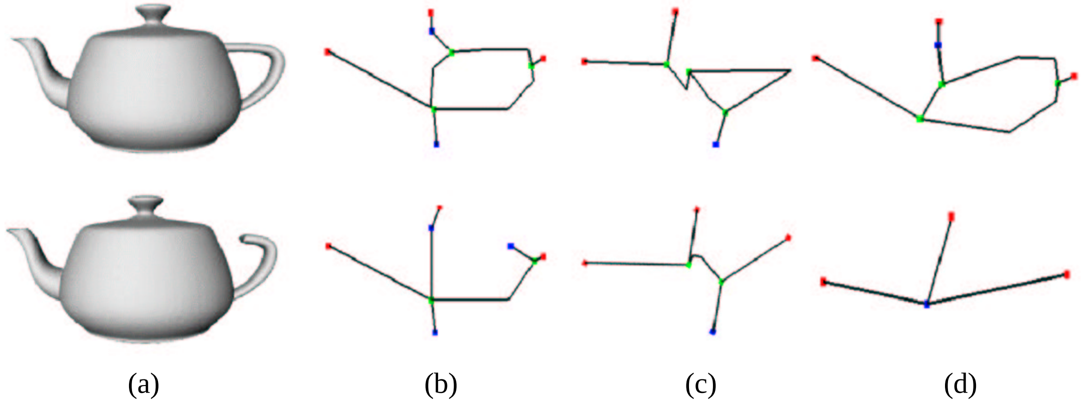


Figure 3.23: *The model of a teapot and the same model where the handle has been broken (a). The two models described by the Reeb graph with respect to distance from the barycentre (b), the Reeb graph with respect to distance from the curvature extrema (c), and Reeb graph with respect to the integral geodesic distance (d).*

extreme generating a new maximum critical point.

As in the case of the geometric shape descriptors, the effectiveness of structural descriptors depends on the application task for which they are used for. Focusing on shape similarity tasks, the comparison among 3D objects implies the capability, of structural descriptors, to capture the salient and relevant features and to describe each of them through a suitable set of attributes. Moreover the methodologies used to compare structural descriptors, have to consider both global aspects of the shape (as it happens comparing two geometric descriptors, see chapter 2) and also local aspects dealing with single features. This is hard to achieve: algorithms perform an exhaustive comparison among the features encoded into the shape descriptor, but their performance are not good when compared with the human intuitive ability to reasoning on shape objects. An impressive example on the lack of the ability to reason, at the same time, on local and global characteristics of an object, is shown in [Sac86]. Where a neurologist, asked a patient named Dr P. and affected by a mental disease, to recognize a common object:

"'What is this?' I asked, holding up a glove. 'May I examine it?' he asked, and, taking it from me, he proceeded to examine it as he had examined the geometrical shape. 'A continuous surfaces,' he announced at last, 'in-folded on itself. It appears to have' - he hesitated - 'five outpouchings, if this is the word.' 'A container of some sort?' 'Yes,' I said, 'and what would it contain?' 'It would contain its contents!' said Dr P., with a laugh. 'There are many possibilities. It could be a change-purse, for example, for coins of five sized. It could . . . ' I interrupted the barmy flow. 'Does is not look familiar? Do you think it might contain, might fit, a part of your body?' No light of recognition dawned on his face. No child would have the power to see and speak of 'a continuous surface . . . in-folded on itself', but any child, any infant, would immediately know a glove as a glove, see it as familiar, as going with a hand. Dr P. didn't. He saw nothing as familiar. Visually, he was lost in a world of lifeless

abstraction. Indeed he did not have real visual world, as he did not have a real visual self."

This is because Dr P., due to his mental disease, is able to reason on the meaningful subparts of the object (the outpouchings of the glove), but he can not see the whole object as a whole. He can reason on a structured set of features but he can not reason on the overall shape, the same is for graph-based comparisons algorithms. On the other hand geometric descriptors allow the reasoning on the overall shape, but they neglect the structural nature of the object[5]. Structural descriptors explicitly encode in a single framework all the necessary information (local and global), thus the success of the reasoning process (e.g. comparison) depends mainly on the comparison algorithm used. In spite of the difficulties to develop algorithms able to take in account both global and local aspects of shapes, the most evident advantage using structural shape descriptors is the capability to produce not only a real number representing how much two object resemble themselves (as it happens for geometrical descriptors and theirs comparison methodologies, e.g. [Vel01, OFCD01, OFCD02, NK01, CRC$^+$02, VT03, KFR03]), but also the information about sub-parts correspondence between descriptors and thus sub-parts correspondence between objects. A powerful method to obtain this information is the inexact subgraph isomorphism, where graph node and edge mappings give information on the sub-parts resemblance.

In order to investigate the expressive power of the structural descriptors, it is interesting to discuss them with respect to the properties shown in section 1.4. The ability to capture the **salient** features depends on the methodology used to extract the descriptor. Though the descriptors shown in this chapter describe/represent shape objects capturing structural information, they can also explicitly represent geometrical information encoding it as attributes associated to the skeleton or graph subparts. This is useful when, for example roundness and/or sharpness of an object subpart has to be took into account for application tasks. The medial axis descriptor encode the information needed to approximatively reconstruct the object (shape representation). This is because it explicitly encode, as skeletal subparts, all the features of the object as shown in figure 3.2(b). Same considerations hold for the shock graph descriptor (section 3.1.2), since it differ form the medial axis only by the edge orientation of the skeleton. Also methodologies based on volumetric thinning (section 3.2.1) produce skeletal descriptors that explicitly encode salient features of the object. Moreover, the method proposed in [GS99, SSGD03] use the distance transform to approximate the shape object. Nevertheless this kind of descriptors may produce skeletal extra-edges not representing meaningful subparts: for example most of the edges belonging to the mug body shown in figure 3.8 are not meaningful. Even if the scale space decomposition discussed in section 3.2.2 do not generate a skeletal structure, in the sense of medial axis, shock graphs and volumetric thinning, it produce a hierarchical binary tree structure that describes the object, where each salient feature is captured and stored as node of the tree. In this case, differently from the other descriptors, the genus of the object is not explicitly represented, even if it influences the overall descriptor as in the case of geometrical descriptors. Moreover the granularity of the analysis process, can be controlled in order to capture the meaningful object features. Reeb graphs (section 3.2.3) have differential topology, as theoretical background, that guarantee to capture

[5] The geometrical shape descriptors are influenced by the structure of the object, but, as discussed in section 2.4, they do not explicitly encode this information

the holes structure of the object. As shown in figure 3.23, the choice of different Morse functions generate different graphs describing the same object: all the graphs of the upper row describe the handle, the spout, the tip and the body of the teapot, where the different node configuration describe different aspects of the shape. Differently, the Reeb graph based on the height function, may loose some salient feature, depending on the position of the analyzed object. For example, the graph reprsented in figure 3.11(d) do not capture the spout of the teapot. Even if structural features are captured into the Reeb graph, geometrical aspects of the objects, as for example the roundness of the teapot body, are not captured, but they can be explicitly stored as node or edge attributes.

The **relevant** features of an object are explicitly stored by all the structural descriptors discussed in this chapter, apart the scale space decomposition descriptor of section 3.2.2, where holes and protrusions are not encoded explicitly. For example, it is not possible to find out the handle of the teapot, or to understand the protrusion of an object that fit a specific hole of another object. Also for skeletons extracted through volumetric thinning (section 3.2.1) is not easy reasoning about relevant features: as shown in figure 3.8(b) is not possible to discriminate among the skeleton cycles representing the handle of the mug and the skeleton extra-edges, thus the difficulties are to manage holes as relevant features. Reeb graphs, medial axis and shock graphs (sections 3.2.3, 3.1.1 and 3.1.2) allow the identification of salient features as relevant depending on the application context.

The number of branches of the skeletons, and also nodes and edges of graphs, depend on the structural complexity of the described object: features and sub-features of the object are encoded by skeletons or graphs parts. This make the structural descriptors **concise** with a small number of branches, nodes and edges. Even if structural descriptors do not capture geometric aspects of the object, they can be endowed with concise geometric attributes (e.g. node and edge attributes describing the geometry of the related feature). For medial axis, shock graphs and volumetric thinning (sections 3.1.1, 3.1.2 and 3.2.1) the geometric attributes correspond to the radii of the spheres used to define the points of the skeleton. Instead the Reeb graphs and scale space descriptors (sections 3.2.3 and 3.2.2) construction process do not need shape attributes, nevertheless attributes can be easily defined.

The structural descriptors mentioned in this chapter, are extracted from shape object through deterministic algorithms (no statistical methods has been used). This implies that given a methodology, an extraction algorithm and tuned the parameters necessary to run the algorithm on the input model, all the descriptors are **unique**. Medial axis, and thus also shock graphs (sections 3.1.1 and 3.1.2) has been defined as the set of points that is the locus of centres of all maximal discs contained in the 2D object. By this definition the medial axis is univocaly determined, but several algorithms (e.g. based on the distance transform) may produce different approximations of the medial axis skeleton. Also the unicity of the skeletons obtained through volumetric thinning (section 3.2.1) depend on the parameter that control the extraction algorithm, for example the thinness parameter of equation (3.1). The unicity of the scale space decomposition (section 3.2.2) depends on the function f defined on a subset of the surface points. The definition of Reeb graph (section 3.2.3) make it unique by the theoretical view point. Nevertheless, by the operative point of view, the algorithms for the graph extraction need to sample the Morse function on which it is defined, and the sample resolution influences the nodes and edges configuration of the graph.

All the descriptors discussed in this chapter are **invariant to rigid transformations** with the exception of the Reeb graph described in section 3.2.3.1, where its skeleton depends on the position of the input model. This Reeb graph is not invariant to rotations along axes not parallel to the height direction. For example in figures 3.11 two different Reeb graphs has been obtained from the same teapot model in two different positions.

By the point of view of **invariance to the model representation**, medial axis and shock graphs descriptors (sections 3.1.1, 3.1.2) can be extracted both from polygonal and algebraic two-dimensional closed curves. Volumetric thinning descriptors (section 3.2.1) need a voxel representation of the model. In this case, the best suitable model representation is a closed two-manifold polygonal mesh without boundaries (as used in [SSGD03]), where the voxelization really approximate the object surface and it is easy to be computed. The scale-space decomposition descriptor (section 3.2.2), computes a recursive decomposition of the object into features and sub-features, where such decomposition is based on a geodesic distance function among surface points. Also in this case the best suitable model representation for the geodesic distance computation is a closed two-manifold polygonal mesh without boundaries as noticed in [BRS03]. Similar considerations holds also for Reeb graphs (sections 3.2.3). In this case a two-manifold polygonal mesh is necessary to evaluate the Morse function.

Multi-scale representation/description of objects is one of the most interesting **scalability** properties of the structural descriptors. It is the capability to reduct the size of the descriptor selecting relevant features and discarding the non relevant one. For this reason it is tightly correlated to the capability to represent/describe salient and relevant features. For example in a context where it is important understand if a model can be grasped or not, a structural shape descriptor can describe/represent features dealing with handles and discard non-relevant features. Medial axis, shock graphs and Reeb graphs (sections 3.1.1, 3.1.2 and 3.2.3) allow a powerful multi-scale representation/description of the object because they exactly encode holes and protrusions of the object, while volumetric thinning and scale space decomposition (sections 3.2.1 and 3.2.2) do not. Also multi-resolution description/representation can be obtained for structural descriptors. A multi-resolution framework for Reeb graphs has been proposed in [HSKK01] and described in section 3.3. While scale space decomposition descriptor is itself a multi-resolution framework of the shape object.

Due to the **meaning of the information stored** structural descriptors can be used for defining an approximation of the input surface geometry by using basic primitives, i.e. generalized cylinders, each one related to building elements of the skeleton, edges or vertexes with the local information stored during the graph construction [AC02]. In implicit modeling, skeletons, considered as a collection of elements with associated implicit primitives, provide a compact representation that is useful in defining both motion and deformation. Skeleton-like structures are also essential for implicit model animation, in fact during animation, its attributes may change, varying for instance, radius, blending and other surface details. Moreover, depending on the kind of the manipulation task (animation, metamorphosis, growth, etc.), skeletal elements may rotate, stretch, appear or disappear. However, the skeletal elements of the intermediate shapes obtained during the animation evolution remain simply to define, articulate and display and the skeletal hierarchy (that is the internal relationships between arcs and nodes) generally does not change [BBB^{+}97]. Finally as described in section 3.3, structural descriptors are suitable for object comparison, with the merit of provide information about structural sub-part correspondence.

Related publications

S. Biasotti and S. Marini. 3D object comparison based on shape descriptors. *International Journal of Computer Applications in Technology*, 23(2/3/4):57–69, 2005.

S. Biasotti, S. Marini, M. Mortara, and G. Patané. An overview on properties and efficacy of topological skeletons in shape modeling. In *Proceedings of Shape Modelling and Applications*, pages 245–254, Seoul, South Korea, June 2003. IEEE Press.

Chapter 4

Comparison Between Structural Shape Descriptors

As shown in chapter 3, skeletons and Reeb graphs provide an efficient encoding of the structure of the object, where this encoding can be easily represented as a graph (see definition A.0.1). This allow the use of graph matching methodologies in order to compare structural descriptors.

To be used for shape matching purposes, a structural descriptor should be independent of object position, rotation and scaling. Usually skeletons satisfy these requirements, while for the Reeb graph the choice depends on the mapping function as discussed in chapter 3. As proposed in [SSGD03] (discussed in section 3.2.1) and [BMM$^+$03], both skeletons and a Reeb graphs may be represented as a-cyclic, directed graphs. However, the skeletal structure requires a number of simplification steps and artifacts, which might alter the topology of the signature, while for the Reeb graph there is a strict relationship between the object topology and the graph structure. Therefore, in the rest of the thesis, the Reeb graph structural shape descriptor will be used for the experiments.

By the assumption that the Reeb graph representation is a directed a-cyclic graph that encodes the salient shape features and the most significant spatial relations between them, the approach proposed in this chapter is to find a mapping function between the structural parts of two Reeb graphs. This is achieved through the construction of a common subgraph A.0.6 between two input graphs. This common subgraph should highlight how much the two shapes overlap.

4.1 Problem Statement

Beside the topological information stored in the graph structure, also attributes associated to edges and nodes, and the information deduced by the directions of the edges concurs to the construction of the mapping function between the structural parts of the two Reeb graph.

The following theorem assures that the Reeb graph is directed and a-cyclic.

Theorem 4.1.1 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mu^{\mathcal{V}}, \mu^{\mathcal{E}})$ be a Reeb graph, it is directed, a-cyclic and attributed graph.*

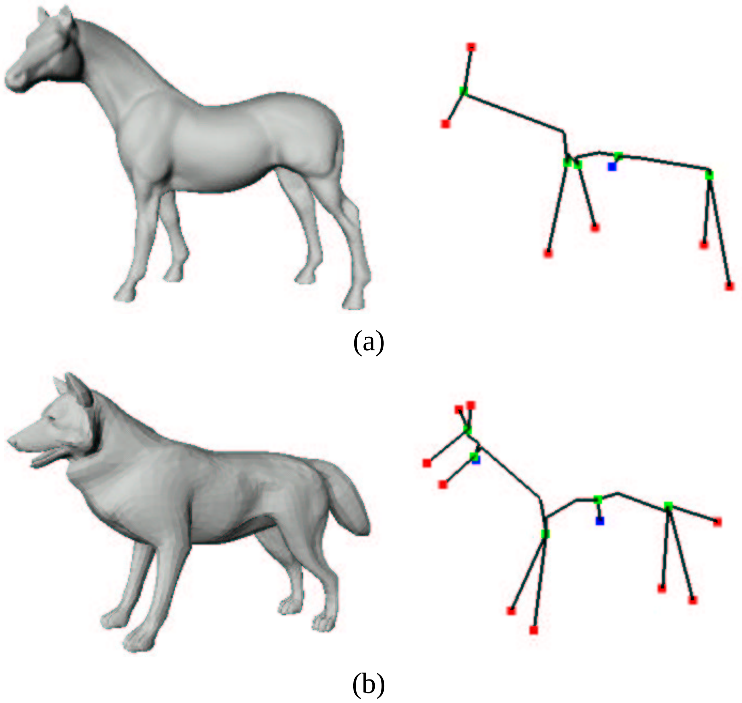
Proof.

- *Directed:* Let e be an edge of $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mu^{\mathcal{V}}, \mu^{\mathcal{E}})$, that is $e \in \mathcal{E}$. From the definition 3.2.1 it follows that the function f cannot be constant along any edge e; therefore, denoting v_1 and v_2 the nodes of e, the relation $f(v_1) \neq f(v_2)$ holds. Then, the edge e may be oriented according to the monotonicity of f along that: the edge e $e = (v_1, v_2)$ (resp. $e = (v_2, v_1)$) is directed from v_1 to v_2 if $f(v_1) < f(v_2)$ (resp. $f(v_2) < f(v_1)$).
- *Acyclic:* The monotonicity of f along each edge $e \in \mathcal{E}$ implies that $\mathcal{G} i$s acyclic. In fact, let P be a path (see definition A.0.3) with starting and ending point v_1. Therefore, there would exist a sequence of nodes $v_1, v_2, \dots, v_n, v_1$ and edges $e_1, e_2, \dots, e_n$ such that $e_1 = (v_1, v_2), \dots, e_n = (v_n, v_1)$. This would imply the following sequence of inequalities: $f(v_1) < f(v_2) < \dots < f(v_1)$, which is clearly impossible from the definition 3.2.1.
- *Attributed:* Attributes may be easily associated to $\mathcal{G} b$y considering either the geometric properties of the part of the object described by nodes and edges, or shape attributes such as texture, color, etc. While their choice may be questionable, the existence of such attributes, both for nodes and edges, is always possible.

■

In particular have to be observed that, since the Reeb graph is directed, each node identifies a subgraph S, where V_S contains the node v itself and all nodes for which the node is an ancestor.

The mapping function between the structural parts of two Reeb graph can be obtained by computing a bijective function between nodes and edges of the two graphs. As shown in appendix A there are different way to build such function: for example the graph isomorphism, subgraph isomorphism, common subgraph and error tolerant graph isomorphism. The existence of a graph isomorphism (see definition A.0.4) implies that the graphs must be equivalent. This is a too strong condition that can not be satisfied for the evaluation of shape similarity, in fact, due to the capability of a descriptor to capture the salient features of an object, two similar object (similar and not identical) are described/represented with two similar and not identical shape descriptors. In figure 4.1 the models of a wolf and a horse are shown together with their Reeb graphs. The two models are similar but due to small morphological differences the two structural descriptors are slightly different: the two subgraphs related to the heads are different because the mouth of the wolf is open. The two front legs of the horse are connected to two different branching nodes, while the two front legs of the wolf are joined to the same branching node. The strong requirements of the graph isomorphism can be relaxed using a weaker similarity measure based on the subgraph isomorphism (definition A.0.5) and the notion of common



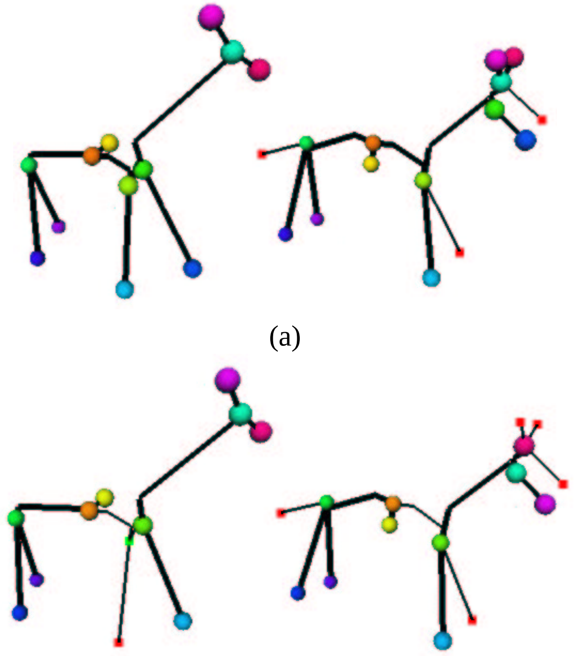
(a)

(b)

Figure 4.1: *Horse (a) and wolf (b) models together with their Reeb graphs based on the distance from the barycentre function.*

subgraph (CS) (A.0.6). From an intuitive point of view, comparing two structural descriptors means constructing the most suitable common sub-graph: the wider the common subgraph is, the more the two structural descriptors are similar. The word ”suitable” has been intentionally used because more than one common subgraph can be defined. For example in figure 4.2 two common subgraphs are shown for the graphs representing the horse and the wolf of figure 4.1 respectively. In figure 4.2(a) is represented the maximum common subgraph (MCS) as defined in A.0.7, while in figure 4.2(b) the common subgraph represented is smaller than the MCS but more meaningful, because semantically equivalent sub-parts of the object are correctly recognized and mapped together.

Unfortunately, the construction of the MCS is a well known NP-complete problem, thus, its exact computation is time consuming, when the shape descriptors are composed by a large numbers of nodes and edges. Therefore, several strategies and heuristic assumptions have been adopted to simplify this problem [BMM$^+$03, BM05, MSF05]. In the following sections of this chapter will be shown an algorithm for the computation of the maximum common subgraph (MCS) between two directed and a-cyclic graphs and several heuristic techniques Will be discussed.



(a)

(b)

Figure 4.2: *Two common subgraphs between the horse and wolf graphs. Nodes having the same colors are mapped together. Thick edges represent edge mapping*

4.2 Algorithm Description

A naive algorithm for the computation of the $\mathcal{M}CS_{\mathcal{G}_1,\mathcal{G}_2}$ between the two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ is shown in figure 4.3. This algorithm is optimal in the sense that it returns the correct result, but it has an exponential computational complexity. For this reason, heuristic techniques have to be considered in many applications in order to approximate the MCSs of two input graphs.

1. enumerate all the possible mappings m among $\mathcal{V}_1$ and $\mathcal{V}_2$;
2. search those mappings m that satisfy the definition of maximum common subgraph. Of course, m is not necessarily unique.

Figure 4.3: *The naive algorithm*

Even if the described algorithm is very simple, it is not easy to define heuristic techniques based on the attributes of edges and nodes, or on reasoning about the graph structure. Also, it is not easy to devise an approximation which makes the structural shape matching robust to structural noise in the graphs. With the aim to introduce such techniques the point 2 of the algorithm can be modified as in the figure 4.4, [MSF05].

1. enumerate all the possible mappings m among $\mathcal{V}_1$ and $\mathcal{V}_2$;
2. for each listed mapping m, compute the common subgraphs of $\mathcal{G}_1$ and $\mathcal{G}_2$ obtainable by expanding m.

Figure 4.4: *The new algorithm obtained modifying the naive algorithm shown in figure 4.3*

The point 2 of the new algorithm expands the input mapping m as much as possible while respecting the definition of common subgraph.

Since the first point of both algorithms enumerates all the possible mappings, proving that the second formulation is correct reduces to proving that the expansion always produces a correct common subgraph and that it does not alter the structure of a maximum common subgraph, should this be given as input to the expansion process.

4.2.1 Pseudo code

In this section, the alternative algorithm is described by explaining the steps of its pseudo code.

The data structures involved in the algorithm are:

- `G_1` = $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mu^{\mathcal{V}_1}, \mu^{\mathcal{E}_1})$ and `G_2` = $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mu^{\mathcal{V}_2}, \mu^{\mathcal{E}_2})$, are the two input graphs of the algorithm;
- `M`: is a set of node pairs (v_1, v_2), where $v_1 \in \mathcal{V}_1$ and $v_2 \in \mathcal{V}_2$;
- `CS`: is a common subgraph of `G_1` and `G_2`. Each element of `CS` is a four-tuple (v_1, v_2, e_1, e_2), where $v_1 \in \mathcal{V}_1, v_2 \in \mathcal{V}_2$ and $e_1 \in \mathcal{E}_1, e_2 \in \mathcal{E}_2$. The node pair (v_1, v_2) and the edge pair (e_1, e_2) of each four-tuple, are the node-mapping and the edge mapping of the common subgraph.
- `MCS`: is the set of all the common subgraphs computed by the algorithm;
- `CANDIDATES`: is the set of four-tuples (v_1, v_2, e_1, e_2)candidate to became an element of `CS`.
- `CS_SET`: is a set of pairs (`CS`, `CANDIDATES`), where each pair represents a common subgraph `CS` and the set `CANDIDATES` is the set of possible node and edge mappings candidate to expand `CS`.

Listing 4.1: *The main procedure*

```
MCS(G_1, G_2)
{
    M   = empty_set
    CS  = empty_set
    MCS = empty_set

    M = Mappings_Set(G_1, G_2)

    for each m in M{
        CS = CS_from_Mapping(m)
        Add(CS, MCS)
    }
    return Max(MCS)
}
```

The main procedure of the algorithm is `MCS()`, shown in the listing 4.1. It enumerates the set of initial mappings $\mathcal{M}_{\mathcal{G}_1,\mathcal{G}_2}$ among the nodes of $\mathcal{G}_1$ and $\mathcal{G}_2$. For each $m \in \mathcal{M}_{\mathcal{G}_1,\mathcal{G}_2}$ it generates the set `CS` of the common subgraphs obtained expanding m. `CS` is computed by the function `CS_from_Mapping(`m`)` applied on the set of initial mappings m. Obviously, each m may generate more than one common subgraph and the expansion procedure `CS_from_Mapping(`m`)` produces all of them. Finally, the MCSs are obtained selecting the common subgraphs with the largest number of nodes.

The procedure `CS_from_Mapping()` (listing 4.2) expands the set of node pairs in `m` in order to generate a set of common subgraphs of the two input graphs. First `Init_Candidates()` (listing 4.3) transforms the set of node pairs in m in a set of candidates, where each pair become a candidate for growing the common subgraph. At the line 8 of the procedure `Init_Candidates()`, the two edges e_1 and e_2 are set to `NULL`, because it make no sense to account for the edge mapping for the initial candidates generated from `m`. After the initialization of `CANDIDATES`, the set `CS_SET` (line 10 of the listing 4.2) contains only one pair (`CS`, `CANDIDATES`) with `CS` $= \phi$, since no node and edge mappings has been yet produced.

The main loop of the procedure `CS_from_Mapping()` (starting at line 12 of the listing 4.2) aims at adding new elements to `CS` by checking if the nodes of the candidate element can be mapped correctly. This loop iterates through all the elements (`CS`, `CANDIDATES`) of `CS_SET`.

Given the pair (`CS`, `CANDIDATES`), the secondary loop starting at line 13, iterates through the candidate elements `(v_1, v_2`, $\star$, $\star$`)` until `CANDIDATES`become empty. The symbols $\star$ replace the edges of the candidate element because it is not necessary at this moment. The candidate element is extracted through `Pop()`, that remove it from `CANDIDATES`. For each node pair `(v_1, v_2)` line

Listing 4.2: *Expansion of the initial mapping*

```
CS_from_Mapping(m)
{
  CS         = empty_set
  CANDIDATES = empty_set
  CS_SET     = empty_set
  v_1        = empty_node
  v_2        = empty_node

  Init_Candidates(CANDIDATES, m)
  Add((CS, CANDIDATES), CS_SET)

  for each (CS, CANDIDATES) in CS_SET{
    while not Empty(CANDIDATES){
      (v_1,v_2,*,*) = Pop(CANDIDATES)
      if(Mapped(v_1) or Mapped(v_2))
        Resolv_Conflict((v_1,v_2,*,*), (CS,CANDIDATES), CS_SET)
      else{
        Add((v_1,v_2,*,*), CS)
        Update((v_1,v_2,*,*), CANDIDATES)
      }
    }
  }
  return Max(CS_SET)
}
```

Listing 4.3: *Transforms the in ital set of node mappings into the initial candidates*

```
Init_Candidates(CANDIDATES, m)
 {
   CANDIDATES = empty_set
   v_1        = empty_node
   v_2        = empty_node

   for each (v_1, v_2) in m
      Add((v_1, v_2, NULL, NULL), CANDIDATES)
 }
```

15 checks if the nodes of the candidate extracted are already mapped or not: if not, that is if they do not belong to any four-tuple in `CS`, they are added to `CS`, and new candidates are generated by `Update()` (listing 4.4); otherwise, the `Resolve_Conflict()` (listing 4.5) is called to handle the situation.

Listing 4.4: *`Update` adds new elements to `CANDIDATES`by processing the four-tuple `p`.*

```
Update((v_1, v_2, *, *), CANDIDATES)
{
  u   = empty_node
  v   = empty_node
  e_1 = empty_edge
  e_2 = empty_edge

  for each edge e_1 out coming from v_1 {
     for each edge e_2 out coming from v_2 {
        u = Opposite(v_1, e_1)
        v = Opposite(v_2, e_2)
        Add((u,v, e_1, e_2), CANDIDATES)
     }
  }
}
```

The procedure `Update()` generates new candidate elements from the four-tuple `(v_1, v_2, *, *)`. A new four-tuple is obtained for each out coming edge `e_1` from `v_1` and `e_2` from `v_2`, where the new candidate element `(u, v, e_1, e_2)` is generated exploiting the direction of the edge `e_1` = `(v_1, u)` and `e_2` = `(v_2, v)`.

The conflicts are solved by the procedure `Resolve_Conflict()` (listing 4.5) by forking the expansion of the current `CS` into two common subgraphs where the new one is obtained by eliminating the pairs of nodes responsible of the conflict and the subsequent part of the common subgraph through the procedure `Reset()` (listing 4.6). The lines 10, 12 and 14 controls which nodes raises the conflict, and as consequence `Reset()` is called. After `Reset()` finished, the two sets `NEW_CS` and `NEW_CANDIDATES` are produced, each one representing a new common subgraph and a new set of candidates, respectively. Therefore the four-tuple `(v_1,v_2, *, *)` is added to `NEW_CS`, the new set of candidates `NEW_CANDIDATES` is updated and finally the new pair `(NEW_CS, NEW_CANDIDATES)` is added to `CS_SET`.

The procedure `Reset()` (listing 4.6) updates the sets `CS` and `CANDIDATES` by deleting all the nodes involved in the conflict. The statement `Delete(v_1,CS)` eliminates all the four-tuples containing the node `v_1` from the common subgraph `CS`, and the loop at line 7 iterates among all the nodes `v` of `CS`, connected through a directed path in `CS`to `v_1`. All the four-tuples of `CS` containing such `v` are

Listing 4.5: *The procedure* `Resolve_Conflict()` *is called to solve a conflict*

```
Resolv_Conflict((v_1,v_2, *, *), (CS, CANDIDATES), CS_SET)
{
 NEW_CS            = empty_set
 NEW_CANDIDATES = empty_set
 CS_SET            = empty_set

 NEW_CS            = CS
 NEW_CANDIDATES = CANDIDATES

  if(Mapped(v_1) and Mapped(v_2))
     Reset(v_1, v_2, NEW_CS, NEW_CANDIDATES)
  else if(Mapped(v_1) and not Mapped(v_2))
          Reset(v_1, NULL, NEW_CS, NEW_CANDIDATES)
        else if(not Mapped(v_1) and Mapped(v_2))
                Reset(NULL, v_2, NEW_CS, NEW_CANDIDATES)

 Add((v_1,v_2, *, *), NEW_CS)
 Update((v_1,v_2, *, *), NEW_CANDIDATES)
 Add((NEW_CS, NEW_CANDIDATES), CS_SET)
 }
}
```

deleted at line 8. The same considerations hold for the loops starting at the lines 9, 14 and 16.

To better understand how the `Resolve_Conflict()` works, the example of figure 4.5 has been provided, where only bold edges are the mapped ones. The node and edge mapping of the common subgraph shown in figure 4.5(a) is represented by $\texttt{CS\_SET} = \{(\texttt{CS}, \texttt{CANDIDATES})\}$ where:

$$\texttt{CS} = \{(\texttt{a,a1}, \texttt{NULL}, \texttt{NULL}), (\texttt{c,f1}, \beta, \beta 1), (\texttt{d,c1}, \chi, \chi 1), (\texttt{b,b1}, \alpha, \alpha 1)\}$$

and

$$\texttt{CANDIDATES} = \{(\texttt{c,c1}, \delta, \delta 1), (\texttt{e,c1}, \varepsilon, \chi 1)\}.$$

If `Pop()` selects the candidate ($\texttt{c,c1}$,δ, $\delta 1$), then both nodes `c` and `c1` are already mapped and the statement `Resolve_Conflict()` has to be executed. The new pair generated by `Reset()` (listing 4.6) is ($\texttt{CS}'$, $\texttt{CANDIDATES}'$), represented in figure 4.5(b), where:

$$\texttt{CS}' = \{(\texttt{a,a1}, \texttt{NULL}, \texttt{NULL}), (\texttt{b,b1}, \alpha, \alpha 1), (\texttt{c,c1}, \delta, \delta 1)\}$$

and

$$\texttt{CANDIDATES}' = \{(\texttt{d,d1}, \chi, \phi 1), (\texttt{d,e1}, \chi, \varepsilon 1), (\texttt{e,d1}, \varepsilon, \phi 1), (\texttt{e,e1}, \varepsilon, \varepsilon 1)\}$$

Listing 4.6: *The procedure* `Reset()` *starts a new common subgraph as consequence of the conflict.*

```
Reset(v_1, v_2, CS, CANDIDATES)
{
  v = empty_node

  if(v_1 not NULL){
    Delete(v_1,CS)
    for each v such that Path(v,v_1,CS)
      Delete(v, CS)
    for each v such that Path(v,v_1,CANDIDATES)
      Delete(v,CANDIDATES)
  }
  if(v_2 not NULL){
    Delete(v_2,CS)
    for each v such that Path(v,v_2,CS)
       Delete(v, CS)
    for each v such that Path(v,v_2,CANDIDATES)
       Delete(v,CANDIDATES)
  }
}
```

`CS`′ has been obtained from `CS` deleting the two elements involving `c` and `c1`, and the elements whose nodes are connected to `c` or `c1` with a path contained in `CS`. Analogous considerations are used to obtain `CANDIDATES`′ from `CANDIDATES`.

4.2.2 Correctness

The aim of this section is to provide an outline of the proof that the algorithm of figure 4.4 produces the MCSs of two input graphs. The first step of the naive algorithm enumerates all mappings among the nodes of the two graphs. If a node mapping m produced by the step 1 of the algorithm shown in figure 4.4 represents the MCS, the proposed algorithm is correct if the statement `CS_from_Mapping(`m`)` (listing 4.2) produces as output a common subgraph $\mathcal{CS}_m$ where the corresponding node mapping is identical to m.

In order to prove the previous assertions the following results have to be shown:

- the nodes involved in $\mathcal{CS}_m$ are an injective function among the nodes of the two graphs;
- $\mathcal{CS}_m$ is a common subgraph of the two input graphs, for each $m \in \mathcal{M}_{\mathcal{G}_1,\mathcal{G}_2}$;
- if m is a MCS, than the node mapping related to $\mathcal{CS}_m$ corresponds to the same MCS.

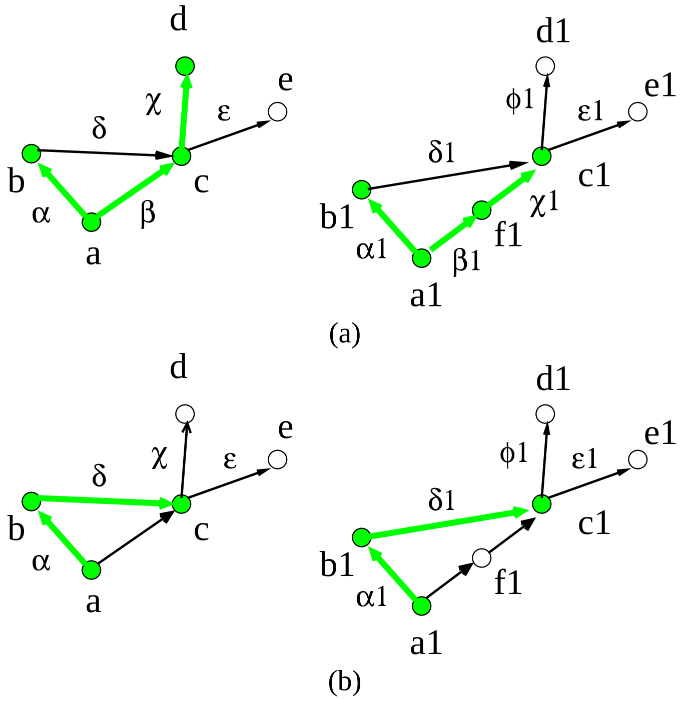


Figure 4.5: *The conflict between the candidates (*`c,c1`*,δ, $\delta 1$) of the two graphs shown in a) is solved and a new common subgraph is produced b).*

In the following, these notations are used: let $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mu^{\mathcal{V}_1}, \mu^{\mathcal{E}_1})$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mu^{\mathcal{V}_2}, \mu^{\mathcal{E}_2})$ be two graphs,

$$\mathcal{CS}_m \subseteq (\mathcal{V}_1 \times \mathcal{V}_2 \times \mathcal{E}_1 \times \mathcal{E}_2)$$

is the set of four-tuples built from the input mapping $m \in \mathcal{M}_{\mathcal{G}_1,\mathcal{G}_2}$ by `CS_from_Mapping(`m`)`, and

$$K = \{(v_1, v_2) \mid (v_1, v_2, \cdot, \cdot) \in \mathcal{CS}_m\}$$

is the set of node mappings of $\mathcal{CS}_m$.

Lemma 4.2.1 *The set of node pairs K is a injective function between $\mathcal{V}_1$ and $\mathcal{V}_2$.*

Proof. When the execution of `CS_from_Mapping(`m`)` ends, `CS_SET` is a set of pair (CS_i, C$ANDIDATES_i$) where C$ANDIDATES_i$ = ϕ for each i, and the sets K_i are injective functions between nodes. This assertion can be proved by induction on $|K_i|$:

base: initially `CS_SET` $= \{(\phi,$ C$ANDIDATES_0)\}$, thus $K_0 = \phi$ and $|K_0| = 0$. In this case $K_0 = \phi$ satisfies the definition of injective function.

induction step: K_0 is a injective function where $|K_0| = n$. Let $(v_1, v_2, \cdot, \cdot)$ be the candidate extracted by `Pop()` from `CANDIDATES`. If the control at line 15 of `CS_from_Mapping()` (listing 4.2) is not satisfied, then there is no exists a pair $(v_1', v_2') \in K_0$ such that $v_1 = v_1'$ and $v_2 = v_2'$. Thus the four-tuple $(v_1, v_2, \cdot, \cdot)$ is added to CS_0 and $K_0 = K_0 \cup \{(v_1, v_2)\}$ is an injective function where $|K_0| = n+1$. If the `if` condition is satisfied, `Resolve_Conflict()` has to be executed and a new pair (CS_1,C$ANDIDATES_1$) is added to `CS_SET`. CS_1 is obtained from CS_0 deleting the four-tuple $(v_1', v_2', \cdot, \cdot)$ that makes the `if` condition false and deleting also the four-tuples whose nodes are linked to v_1 and v_2 through a path in CS_0, then $(v_1, v_2, \cdot, \cdot)$ is added to CS_1. Thus K_1 is again an injective function. ■

This result asserts that, during the construction of the common subgraph between $\mathcal{G}_1$ and $\mathcal{G}_2$, each node belonging to $\mathcal{G}_1$, is associated to one and only one node of $\mathcal{G}_2$. The following theorem uses this lemma to prove that each element (CS, C$ANDIDATES$) $\in$ `CS_SET` is a common subgraph of $\mathcal{G}_1$ and $\mathcal{G}_2$.

Theorem 4.2.2 *$\mathcal{C}S_m$ represents a common subgraph of the two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$.*

Proof. The results to prove are:

1. the two graphs $\mathcal{G}_1' = (\mathcal{V}_1', \mathcal{E}_1', \mu^{\mathcal{V}_1'}, \mu^{\mathcal{E}_1'})$ and $\mathcal{G}_2' = (\mathcal{V}_2', \mathcal{E}_2', \mu^{\mathcal{V}_2'}, \mu^{\mathcal{E}_2'})$ are subgraphs of $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively, where
$$\begin{aligned}\mathcal{V}_1' &= \{v \mid (v, \cdot, \cdot, \cdot) \in \mathcal{C}S_m\}\\ \mathcal{V}_2' &= \{v \mid (\cdot, v, \cdot, \cdot) \in \mathcal{C}S_m\}\\ \mathcal{E}_1' &= \{e \mid (\cdot, \cdot, e, \cdot) \in \mathcal{C}S_m\}\\ \mathcal{E}_2' &= \{e \mid (\cdot, \cdot, \cdot, e) \in \mathcal{C}S_m\}\end{aligned}$$

2. $\mathcal{G}_1'$ and $\mathcal{G}_2'$ are isomorphic.

The proof of the previous two points are given below:

1. $\mathcal{G}_1'$ is a subgraph of $\mathcal{G}_1$ because $\mathcal{V}_1' \subseteq \mathcal{V}_1$ and $\mathcal{E}_1' \subseteq \mathcal{E}_1$ where each edge e $\in \mathcal{E}_1'$ has extreme nodes belonging to $\mathcal{V}_1'$. If an edge $e = (v_1, v_1') \in \mathcal{E}_1'$ exists such that $v_1 \notin \mathcal{V}_1'$ or $v_1' \notin \mathcal{V}_1'$, the statement `Update(`$(v_1, v_2)$`, CANDIDATES)` would find an edge not having v_1 as source node as out coming edge from v_1, or an edge having as opposite node of v_1 a node v_1' not adjacent to itself. Analogous considerations hold for $\mathcal{G}_2'$.

2. K corresponds to a bijective function $f : \mathcal{V}_1' \to \mathcal{V}_2'$ ($\mathcal{V}_1' \subseteq \mathcal{V}_1$ and $\mathcal{V}_2' \subseteq \mathcal{V}_2$) because, as shown by lemma 4.2.1, it is injective and $|\mathcal{V}_1'| = |\mathcal{V}_2'|$. The result to prove is that for each edge $e_1 = (v_1, v_1') \in \mathcal{E}_1'$ there exists an edge $e_2 = (f(v_1), f(v_1')) \in \mathcal{E}_2'$ and vice versa. If an edge $e = (v_1, v_1') \in \mathcal{E}_1'$ exists such that $f(v_1)$ and/or $f(v_1')$ are not extreme nodes of an edge belonging to $\mathcal{E}_2'$, then `Update(`$(v_1, f(v_1))$`, CANDIDATES)` would generates a candidate $(v_1', f(v_1'), e_1, e_2)$ where v_1' would be adjacent to v_1 and $f(v_1')$ would not be adjacent to $f(v_1)$, which is impossible. Analogously for the vice versa.

■

The next theorem assures that the algorithm produce the MCS of $\mathcal{G}_1$ and $\mathcal{G}_2$.

Theorem 4.2.3 *Let m be a node mapping representing an MCS between two input graphs, than the node mapping K related to $\mathcal{C}S_m$ corresponds to the same maximum common subgraph as m.*

Proof. Let $\mathcal{C}S_m$ a common subgraph not representing the maximum common subgraph m, then the following cases have to be discussed:

1. $K \neq m$, $|K| < |m|$: The output of `CS_from_Mapping(`m`)` is $\mathcal{C}S_m$ = $\max\,\{\mathrm{C}S_i \mid (\mathrm{C}S_i, \mathrm{C}ANDIDATES_i) \in$ `CS_SET`$\}$, let ($\mathrm{C}S$,$\mathrm{C}ANDIDATES$) be such output. Let us suppose, without lack of generality, that the difference between K and m is the node pair (v_1, v_2). During the Execution of `CS_from_Mapping()` the four-tuple $(v_1, v_2, NULL, NULL)$ belongs to the set $\mathrm{C}ANDIDATES$ but it is never added to $\mathrm{C}S$. This means that it is never selected, but this is impossible because when `CS_from_Mapping()` ends all the $\mathrm{C}ANDIDATES_i$ are empty. Otherwise, if the four-tuple were selected, it would cause the execution of `Resolve_Conflict()`. Also this case is absurd because neither v_1 nor v_2 can be already mapped because from lemma 4.2.1 K is a bijective mapping.

2. $K \neq m$ and $|K| > |m|$: from theorem 4.2.2, $\mathcal{C}S_m$ corresponds to a common subgraph between the two input graphs, and since m corresponds to the MCS of th same two graphs, $|K| > |m|$ is absurd.

3. $K \neq m$ and $|K| = |m|$: the proof of this case is analogous to the proof of the point 1.

■

4.3 Heuristics

The computation of the maximum common subgraph of two graphs is a common approach for comparing graphs, but its computational costs make the problem not tractable in many application domains. Most importantly, it is often necessary to insert heuristics in the matching process to be able to adapt the process to the characteristics of the shapes under examination. The algorithm described in section 4.2 is structured in a way that heuristic techniques can be easily plugged in it. Also, the expansion mechanism allows to gain in efficiency and speed as it will be discussed in this section.

The maximum common subgraph is obtained by providing as input to the `CS_from_Mapping()` all the mappings among the nodes of the two input graphs and by selecting the common subgraph with the largest number of nodes. A sensible improvement of the matching process can achieved by relaxing the problem setting and allowing a common subgraph to be accepted also an approximated solution.

4.3.1 Node Relevance and Initial Mapping

with respect to the example in Figure 4.6, the optimal solution, that is the MCS, is obtained running the expansion process simply on the pair $m = \{(\texttt{a},\texttt{a1})\}$. In this case the common subgraph obtained as output from `CS_from_Mapping(`m`)` corresponds to the MCS of the two graphs, and the process is run on a highly reduced input set of mapping. In general, running the algorithm on a subset of the initial mapping yields to approximations of the maximum common subgraph, and heuristics or semantic knowledge can be used to select the best candidate initial mappings. It is clear, indeed, that some nodes are more relevant than others, depending on the attributes and on the topology of the graph.

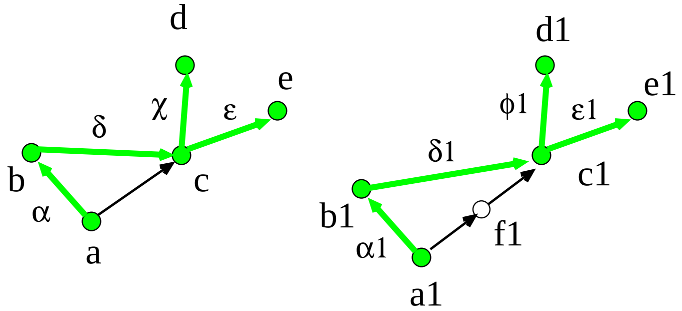


Figure 4.6: *Nodes and edges belonging to the maximum common subgraph are:* $\{$*(*`a`*,* `a1`*, NULL, NULL), (*`b`*,* `b1`*,* $\alpha, \alpha1$*), (*`c`*,* `c1`*,* $\delta, \delta1$*), (*`d`*,* `d1`*,* $\chi, \phi1$*), (*`e`*,* `e1`*,* $\varepsilon, \varepsilon1$*)*$\}$

A sensible improvement of the computational cost of the process can be obtained by reducing the number of input mappings, at the cost of accepting solutions that are not optimal, that is common subgraphs which might be not maximum. This task can be achieved taking into account the direction of the edges. Since the considered input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mu^{\mathcal{V}}, \mu^{\mathcal{E}})$ is directed, each node $v \in \mathcal{V}$ identifies a subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}', \mu^{\mathcal{V}'}, \mu^{\mathcal{E}'})$ induced by $\mathcal{V}'$, where $\mathcal{V}'$ is the set of nodes with v as ancestor included v itself. For example, in figure 4.6, the nodes of the subgraph associated to the node **c** are: **d**, **e** and **c** itself.

The notion of node relevance can be captured by the subgraph associated to the node: for example, the larger the subgraph associated to the node is, the more the node is relevant. With reference to figure 4.6, the node **c** is more relevant than **d** and the node **a** is more relevant than **c**. This concept of relevance can be used to drive the selection of the best initial candidates for the expansion process. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mu^{\mathcal{V}}, \mu^{\mathcal{E}})$, for each node $v \in \mathcal{V}$, has been computed the related subgraph $\mathcal{G}_v = (\mathcal{V}_v, \mathcal{E}_v, \mu^{\mathcal{V}_v}, \mu^{\mathcal{E}_v})$, than the average node relevance $anr_{\mathcal{G}}$ has been computed as:

$$anr_{\mathcal{G}} = \frac{\sum_{v \in \mathcal{V}} |\mathcal{V}_v|}{|\mathcal{V}|} \tag{4.1}$$

The nodes that are relevant has been computed selecting all the nodes whose subgraph has a size bigger than the average node relevance defined in equation 4.1:

$$R_{\mathcal{G}} = \{v \in \mathcal{V} \mid |\mathcal{V}_v| \geq anr_{\mathcal{G}}\} \tag{4.2}$$

Figure 4.8(a) shows the two graphs of the horse and the wolf of figure 4.2 where the nodes are endowed with labels. In this example, the relevant node of the graph representing the horse and computed through the equation 4.2 are: $anr_{Horse} = \{\mathrm{n}, \mathrm{m}, \mathrm{l}, \mathrm{i}\}$; while for the wolf are: $anr_{Wolf} = \{\mathrm{r1}, \mathrm{q1}, \mathrm{p1}, \mathrm{n1}, \mathrm{i1}\}$.

A more general definition of the equations 4.1 and 4.2 takes in account also attributes of nodes and edges of the graph. The relevance $Rel_{\mathcal{G}}(v)$ of the node v can be defined as:

$$Rel_{\mathcal{G}}(v) = |\mathcal{V}_v| \, w_{\mathcal{G}}(v) \tag{4.3}$$

where $w_{\mathcal{G}}(v)$ is a weight varying on the $[0, 1]$ range and depending on the attributes of nodes and edges of the subgraph $\mathcal{G}_v$ associated to the node v. The weight captures the meaning of the object sub-parts represented as nodes and edges of the graph. For example a complex subgraph, with many nodes and edges but representing small object sub-parts, may be less relevant of a simple subgraph, with few nodes and edges but representing a big object sub-part. Starting from the equation 4.3, the equation 4.1 can be rewritten as:

$$\overline{anr_{\mathcal{G}}} = \frac{\sum_{v \in \mathcal{V}} Rel_{\mathcal{G}}(v)}{|\mathcal{V}|} \tag{4.4}$$

and thus also the set of relevant nodes can be rewritten as:

$$\overline{R_{\mathcal{G}}} = \{v \in \mathcal{V} \mid Rel_{\mathcal{G}}(v) \geq \overline{anr_{\mathcal{G}}}\} \tag{4.5}$$

For example, assuming that the attribute of the edge represents the volume of the object sub-part described by the edge itself, the weight determining the relevance of a node v, can be defined as:

$$w_{\mathcal{G}}(v) = \frac{\sum_{e \in \mathcal{E}_v} \mu^{\mathcal{E}_v}(e)}{\sum_{e \in \mathcal{E}} \mu^{\mathcal{E}}(e)} \tag{4.6}$$

In this case nodes whose subgraphs describe sub-parts with a small volume (with respect to the volume of the whole object) are less relevant than nodes describing sub-parts with a large volume. In figure 4.7(a) the model of a human body is shown together with its structural descriptor, figure 4.7(b). The nodes belonging to the subgraph that correspond to the left hand, figure 4.7(c), are considered relevant with respect to the equations 4.1 and 4.2. On the contrary, due to the small volume of the fingers and the palm of the hand, the nodes of the Left hand are not relevant with respect to the equations 4.3, 4.6 and 4.5.

After the relevant nodes has been computed, they can be used as input for the expansion process defined by `CS_from_Mapping(`m`)`. The relevant nodes can be combined to produce the initial node mapping m. The simplest way to compute the initial mapping between two input graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, is to define it as Cartesian product between the set of relevant nodes:

$$m = R_{\mathcal{G}_1} \times R_{\mathcal{G}_2} \tag{4.7}$$

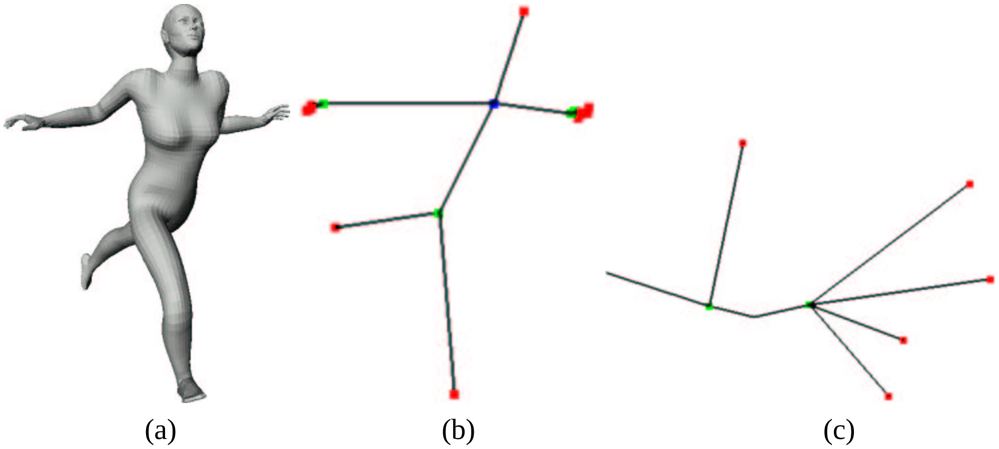


Figure 4.7: *The model of a human body (a), its Reeb graph with respect to the integral geodesic distance (b) and the subgraph representing the left hand of the model (c).*

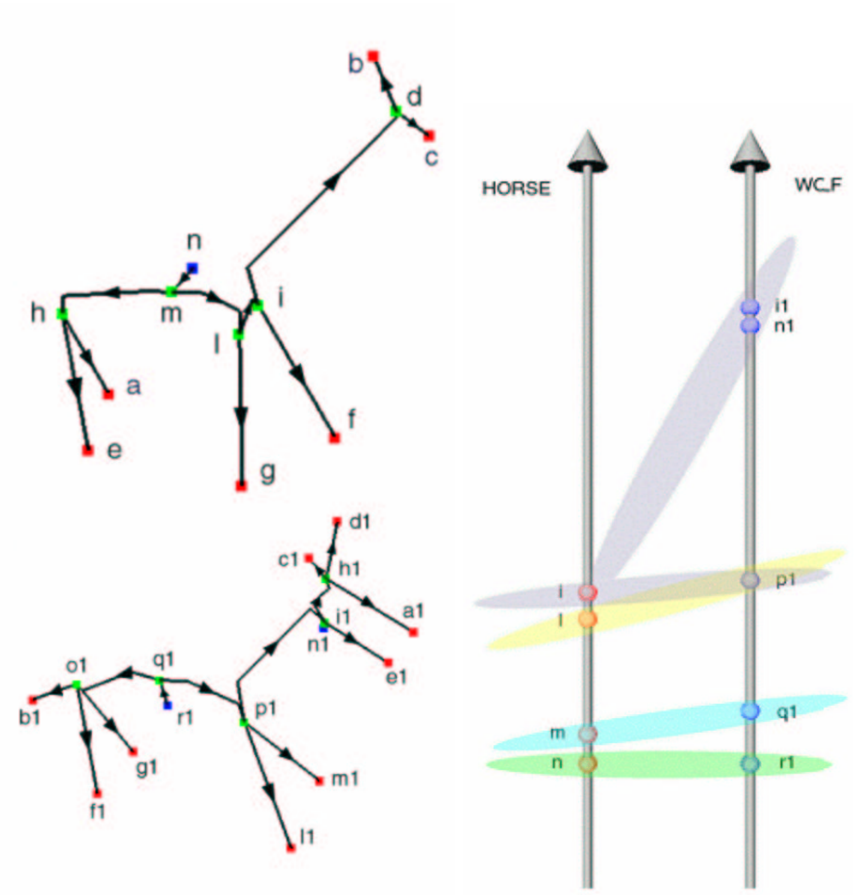
An example of common subgraph obtained considering the set of relevant nodes defined by the equation 4.2 and the initial mapping defined by the equation 4.7 is shown in figure 4.2(a). In this case, although heuristic technique has been used, the comparison algorithm produces as output the maximum common subgraph.

Another method to combine the relevant node is shown in figure 4.8(b), where they are combined with respect to their attributes, in this case, the distance value from the barycentre of the object. Relevant nodes has been grouped in order to combine nodes close to the barycentre with node of the other graph close to the barycentre, and node far from the barycentre with nodes of the other graph far from the barycentre. Nodes close to the barycentre are associated to large subgraphs, thus, are associated to internal large sub-parts of the object, while nodes that move away from the barycentre are associated to progressively small subgraphs (possibly empty subgraphs), thus, are associated to small external sub-parts of the object. The node relevance detection together with the relevant node grouping, allow a coarse to fine matching process that first produces a correspondence between sub-parts of the two object and then it refines the coarse correspondences producing finer sub-part correspondence.S

4.3.2 Distance Function Between Nodes

Another useful heuristic technique can be constructed by combining the notion of subgraph relevance to the idea of expansion process, in particular associating to the pair of nodes (v_1, v_2) the information about how much the common subgraph would expand with the addition of that pair to the subgraph itself. A distance function d between two nodes v_1 and v_2 could be defined in order to capture this information.

The distance $d(v_1, v_2)$ is defined involving node and edge attributes and an approximation of the



(a) (b)

Figure 4.8: *The graphs of a horse and a wolf (a), and an example of node clustering among relevant nodes (b).*

structure of the subgraph related to v_1 and v_2. Two examples of distances that can be plugged in the algorithm come from [SD01] and [HSKK01] both discussed in section 3.3. In [SD01] a topological signature vector $\chi(v)$ describing the structure of the subgraph related to the node v is defined for each node of the graph. The distance $d(v_1, v_2)$ corresponds to the euclidean distance between $\chi(v_1)$ and $\chi(v_2)$. In [HSKK01] the distance value depends mainly by the node attributes.

The distance $d(v_1, v_2)$, proposed in this thesis, involves node and edge attributes and the approximation of the structure of the subgraphs related to v_1 and v_2. It is defined as:

$$d(v_1, v_2) = \frac{w_1 G\_S + w_2 St\_S + w_3 Sz\_S}{w_1 + w_2 + w_3} \tag{4.8}$$

Where G_S, St_S and Sz_S are real numbers belonging to the range $[0, 1]$. They represent geometrical similarity between the node attributes, structural similarity between the node subgraphs and Sz_S

evaluate the similarity between the size of the sub-parts associated to nodes, finally, the three weight w_1, w_2 and w_3 belong to the range $[0, 1]$ and combine the three components of d.

G_S compares the geometric shape descriptors associated to v_1 and v_2. The geometric descriptors capture the parts of the object corresponding to the subgraph associated both to v_1 and v_2. In figure 5.7 is shown the graph of a calf model and the sub-parts related to the nodes of the graph. St_S compares the structure between the subgraphs associated to the nodes v_1 and v_2. As discussed in section 3.3, the structures of two subgraphs can be compared by analyzing the spectrum of the graph, while another coarse but efficient technique can be defined as follow:

$$St\_S = \frac{\overline{in} + \overline{out} + \overline{sub\_n} + \overline{sub\_in} + \overline{sub\_out}}{5} \tag{4.9}$$

where

$$\overline{X} = \frac{|X(v_1) - X(v_2)|}{\max(X(v_1), X(v_2))}$$

and where in and out represent the in degree and the out degree of the two nodes, sub_n the number of the subgraph nodes, sub_in and sub_out the in degree and out degree sum of the subgraph nodes. Finally, two nodes may have sub-parts similar both in structure and in geometry but not in size, in this case they have to result dissimilar. This kind of dissimilarity is captured by Sz_S defined as:

$$\begin{array}{rcl} Sz\_S & = & \overline{sub\_s} \\ sub\_s & = & w_{\mathcal{G}} \end{array} \tag{4.10}$$

where $w_{\mathcal{G}}$ is the sum of the edge attributes of the subgraph and can be defined as in the equation 4.6.

The distance d can be used to reduce the number of elements of `CS_SET`. It acts on the selection of the $\mathrm{C}ANDIDATES$ elements and the `Resolve_Conflict()` statement. The simplest way to use d is to extract the best element $(v_1, v_2, \cdot, \cdot)$ (minimum distance between v_1 and v_2) from $\mathrm{C}ANDIDATES$ and add it to $\mathrm{C}S$ if and only if neither v_1 nor v_2 are already mapped. If v_1 or v_2 are mapped, the candidate is discarded and a new one is extracted until $\mathrm{C}ANDIDATES$ becomes empty. In this case the `Resolve_Conflict()` statement is never recalled. Another example of use of d is to add the best candidate (v_1, v_2) to $\mathrm{C}S$ even if v_1 and/or v_2 are already mapped if and only if the new mapping has a minor distance than the previous ones. In both the previous cases the `CS_SET` set corresponds to a single pair $(\mathrm{C}S, \mathrm{C}ANDIDATES)$.

4.4 Computational complexity

The computational cost of the algorithm described in figure 4.4 and detailed in section 4.2.1 is exponential, because it depends on the first step of the algorithm, that enumerates all the mappings among the nodes of the two input graphs. Actually, given two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mu^{\mathcal{V}_1}, \mu^{\mathcal{E}_1})$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mu^{\mathcal{V}_2}, \mu^{\mathcal{E}_2})$ where $\|\mathcal{V}_1\| = n$ and $\|\mathcal{V}_2\| = m$, the set of node mappings from $\mathcal{G}_1$ to $\mathcal{G}_2$ has m^n elements.

The heuristic techniques discussed in section 4.3 reduce drastically the computational costs of the matching algorithm. The processes that extract the information describing the structure of the subgraph related to a node, and the computation of the relevant nodes of the graphs can be considered as pre-processing steps that are off-line with respect to the graph matching algorithm. However the information involved in the equations 4.3 and 4.9 correspond to the number of nodes belonging to the subgraph associated to a specific node, the sum of the in degree and out degree of the subgraph nodes and the information encoded into the nodes and edges attributes describing the size of the sub-part of the object related to the subgraph. The extraction of all these information have $O(n^2)$ as computational cost, where n is the number of nodes of the graph. This quadratic cost is obtained by analyzing the descendant nodes of each node of the graph. The choice of the set of relevant nodes, defined in the equation 4.5 has a linear computational cost, and the generation of the initial mapping is quadratic with respect to the sum of the relevant nodes of both the graphs, while the evaluation of the similarity between two nodes defined in the equation 4.8 depend only by the computational cost of the comparison process used for the geometric attributes, because all the involved information are computed in the pre-processing step.

Given the initial node mapping and the heuristics defined in section 4.3.2 also the expansion process has quadratic cost with respect to the maximum number of nodes of both the two graphs. When a pair of nodes is selected from the set of candidates (`Pop()` procedure mentioned in the listing 4.2) is added to the common subgraph or it is discarded because, at least, one of the two nodes is already mapped (heuristic rule discussed in section 4.3.2 that influences the `Resolve_Conflict()` procedure shown in the listings 4.2 and 4.5). For each pair of nodes added to the common subgraph, at most, $O(n^2)$ new candidates are added to $CANDIDATES$, and such pair is added to the common subgraph exactly once. The extraction of the best candidate from $CANDIDATES$ is constant if the candidates are arranged with respect to the distance function between nodes (equation 4.8), while the insertion of a candidate into the candidates set is $\log m$, where m is the number of candidates.

Finally the algorithm described in section 4.2 endowed with the heuristic techniques discussed in section 4.3, approximates the maximum common subgraph with a computational cost of $O(n^3)$, where n is the maximum between the number of nodes of the two input graphs.

4.5 Reeb Graph Simplification

Due to the capability of the structural shape descriptors to capture salient structural and topological features of the object (sections 1.4 and 3.4), the object may be represented by a graph with a some redundant nodes and edges, as shown in figure 4.9(a). In this case, such redundant information can be considered as structural noise of the shape descriptor, and it can be removed from the graph through a simplification algorithm able to highlight the significative shape of the object, maintaining the topological information captured by the descriptor, as shown in figure 4.9(b).

The simplification algorithm proposed in this section is not a simple pruning but a more complex

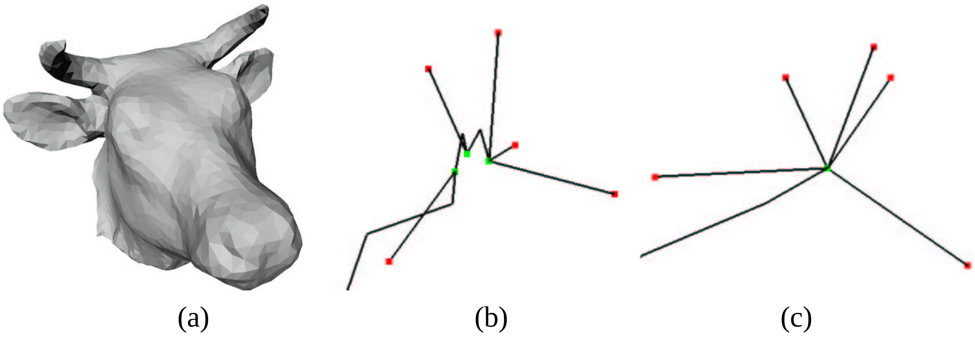


Figure 4.9: *A 3D model (a), its structural shape descriptor (b) and a simplified version of the descriptor (c). In (c) the redundant nodes (green nodes) of the graph has been detected and collapsed into only one node.*

process able to transform the input graph into a new one, which describes the global structure of the object and discards the graph elements responsible for the noise.

Aim of the simplification algorithm is to eliminate the non-relevant object features represented in the graph by guaranteeing the topological consistency of the underlying model. In order to conserve the capability of the Reeb graph to represent the topology of the object, when an edge is removed from the graph also a node has to be deleted. Moreover, in order to preserve the topology and the acyclicity of the graph, a node n and an edge $e = (v, n)$ are deleted from $\mathcal{G}$ only if there are not other edges e_1 from the same node v to n.

Minima and maxima are regarded as feature nodes, while saddle ones describe how the features and the ”body” parts of the object are connected. Three threshold values, Th_m, Th_M and Th_s, are set to check the relevance of both minima, maxima and saddles nodes respectively. Such thresholds, which vary from the smallest to the biggest edge attribute, may be either automatically computed, for instance as the average of the edge attributes, or user defined. The main idea is to simplify nodes that have relevance less or equal to the given thresholds. The simplification is the more incisive, the higher threshold is.

Simple minimum and maximum simplification: In the Reeb graph, simple minima and maxima are represented by nodes, which may be linked to both saddle and maximum (minimum) nodes. However, when a simple maximum M is connected to a simple minimum m, the graph is composed only by the two nodes m and M and the edge (m, M) because a surface S having only two critical points is homeomorphic to a sphere [Mil63].

Only minimum and maximum nodes that are adjacent to saddle nodes and less relevant than Th_m and Th_M are simplified. When all minima and maxima adjacent to a saddle node are removed, the saddle might be transformed into a minimum or a maximum node (eventually complex), according to the number and the direction of the edges incident into it, see figure 4.10.

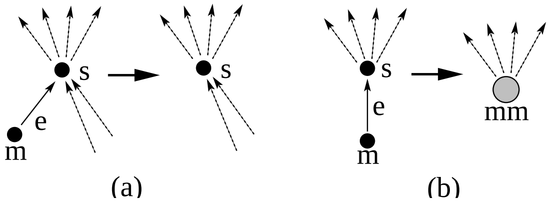


Figure 4.10: *The simplification of a minimum node. The labels of the node indicate its type.*

Complex node simplification: Complex maxima and minima can be adjacent both to nodes and macro-nodes, has to be recalled that, as defined in section 3.2.3 macro-nodes correspond to complex maxima and minima. A complex minimum mm is not relevant if there exists an edge $e = (mm, s)$ where s is a saddle node and the attribute of e is smaller than Th_m. If there exists more than one edge with such characteristics, the simplification algorithm chooses the one with the smallest relevance. The node mm is simplified removing the edge e and connecting all other outgoing nodes to the saddle node s. The node mm is also removed and the saddle s re-classified according to its behavior in the new situation (see figure 4.11). Complex maxima are handled in a symmetric manner.

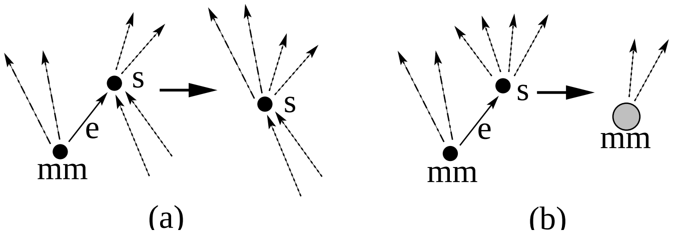


Figure 4.11: *The simplification of a minimum macro-node. The labels of the node indicate its type.*

Saddle simplification: Saddles are graph nodes with both ingoing and outgoing edges. A saddle node v_1 can be simplified if it is connected to another saddle v_2 and the edge $e = (v_1, v_2)$ connecting the two saddles has attribute smaller than the threshold Th_s. In this case all nodes adjacent to v_1 are connected to v_2 and both the edge e and the node v_1 are removed, see figure 4.12.

The simplification process may be arbitrarily repeated until a very simple structure is reached and no more nodes can be simplified according to the proposed criteria.

In particular, has to be highlighted that the constraints inserted in the simplification process of macro-nodes produce a final graph representation which is topologically equivalent to the original one but may not be minimal, in the number of nodes and edges. In addition the order in which the simplification operations are performed may influence the final results.

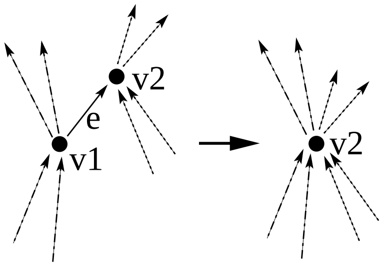


Figure 4.12: *The simplification of a saddle v_1.*

4.6 Similarity Measure

This section address the problem of finding a meaningful distance between two Reeb graphs, considering the more general notion of distance measure among attributed graphs.

As shown in definition A.0.7, the maximum common subgraph of two graphs maximizes the number of nodes and edges taking into account nodes and edges attribute. In particular, has to be observed that the choice of the representative of the maximum common subgraph is unique, up to graph isomorphisms that take into account only the graph topology and discard the nodes and edges attributes. The mappings proposed in figure 4.13(a,b) individuate two possible maximal common subgraphs and show how the attributes of the subgraph are not uniquely determined. To better preserve the relation between the structural descriptors and the shape of the object, the distance measure should take into account both the structure and the nodes and edges attributes of the two graphs. In particular, the bigger the common subgraph of $\mathcal{G}_1$ and $\mathcal{G}_2$ is, the bigger the dissimilarity measure $d(\mathcal{G}_1, \mathcal{G}_2)$ should be.

In applications that are not only related to the topology of the graph, attributes play a fundamental role for the construction of the best suitable common subgraph for a given application context. Therefore it is interesting to introduce the notion of *arc-maximizing common subgraph*, that is the maximal common subgraph that minimizes the sum of the differences of attributes of the edge pairs. More formally:

Definition 4.6.1 (Arc-maximizing subgraph) *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two attributed graphs and M_{Set} the set of their maximum common subgraphs, Using the same notation in definition A.0.8, for each common subgraph $S \in M_{Set}$, let*

$$Diff_S(G_1, G_2) = \sum_{e \in S} |\mu_E(\psi_1^{-1}(e)) - \mu_E(\psi_2^{-1}(e))|$$

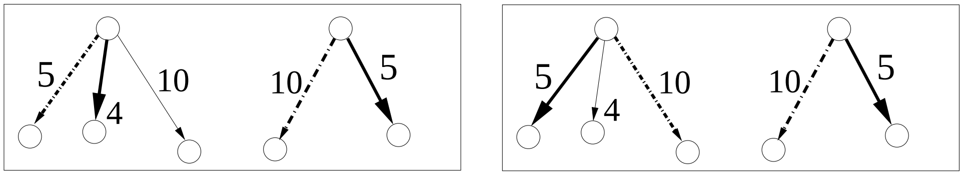
be the sum of the differences of the edge attributes of $\mathcal{G}_1$ and $\mathcal{G}_2$, which are mapped by the graph isomorphism $\psi_1 : \mathcal{G}_1 \to S$ and $\psi_2 : \mathcal{G}_2 \to S$.

Then, a subgraph $S_M \in M_{Set}$ is called arc-maximizing *if*

$$Diff_{S_M}(\mathcal{G}_1, \mathcal{G}_2) \leq Diff_S(\mathcal{G}_1, \mathcal{G}_2), \forall S \in M_{Set}$$

and let $Diff(\mathcal{G}_1, \mathcal{G}_2)$ be the value $Diff_{S_M}(\mathcal{G}_1, \mathcal{G}_2)$.

The notion of arc-maximizing common subgraph refines the concept of maximal common subgraph requiring a kind of optimization on the edge attributes. If the graph G_1 admits more than one maximal common subgraph with respect to G_2, the arc-maximizing one is that for which the edges of the two graphs “better” overlap. Obviously also such a subgraph is unique up to graph isomorphism. In particular, the algorithm proposed in section 4.2 for graph matching, endowed with the heuristics proposed in section 4.3, determines an approximation of the maximal common subgraph which is also arc-maximizing. In fact, additional constraints on the edge attributes (see the equation 4.8) are introduced during the construction of the subgraph isomorphism to minimize $Diff(G_1, G_2)$. Even if the subgraph configurations proposed in figure 4.13 are topologically equivalent, the subgraph mapping proposed in 4.13(b) is also arc-maximizing.



(a) (b)

Figure 4.13: *Two possible maximal mappings between two simple trees. Numbers represent the value of the edges attributes while the line style shows the edge mapping.*

Other approaches evaluate the “optimality” of the graph matching on the transformation of a input graph into the other one instead of the topological maximality of the common subgraph. The method proposed in [Mes97] (see definition A.0.8) and further adopted in [ZTS02] suggests to consider as optimal the graph isomorphism that minimizes the cost of the sequence of editing operations. In that case, possible operations are deletion, insertion and substitution of nodes and edges, that is the change of their attributes. Denoting S a node mapping among the nodes of $\mathcal{G}_1$ and $\mathcal{G}_2$ that satisfies the definition A.0.6, the optimal graph matching with respect to the definition A.0.8 should minimize the distance measure:

$$O_S(G_1, G_2) = \sum_{e \in (G_1 \diagdown \Delta(G_1))} (|\mu_E(e)|) + \sum_{e \in (G_2 \diagdown \Delta(G_2))} (|\mu_E(e)|) + Diff_S(G_1, G_2), \tag{4.11}$$

where $(G_1 \diagdown \Delta(G_1))$ and $(G_2 \diagdown \Delta(G_2))$ respectively indicates the subgraphs of G_1 and G_2, whose elements are not mapped by the isomorphism S, $\psi_1 : \Delta(G_1) \rightarrow S$, $\psi_2 : \Delta(G_2) \rightarrow S$ are the subgraph isomorphisms and $Diff_S(G_1, G_2)$ has the same meaning like in definition 4.6.1.

In figure 4.14 the best mapping between two graphs is shown both with respect to the notion of arc-maximizing common subgraph, see figure 4.14(a), and that with respect to the sum of the edges attributes 4.14(b). In particular has to be observed that every maximum common subgraph is made of two edges, while the optimal mapping with respect to the edge attributes, to be topologically consistent, cannot map more than one edge.

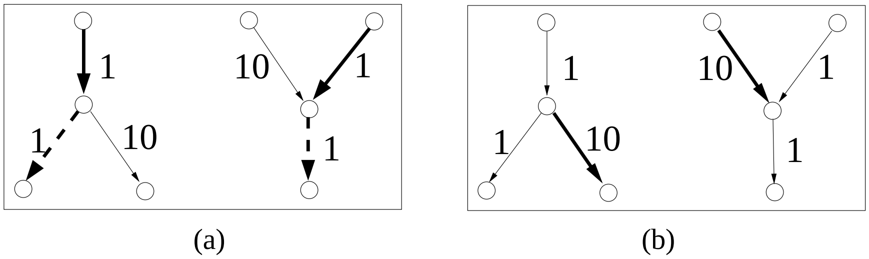


Figure 4.14: *The arc-maximizing common subgraph (a) and the optimal mapping with respect to the differences of the edges attributes (b). Numbers represent the value of the edges attributes while the line style shows the edge mapping.*

Finally, has to be introduced the notion of *arc-overlapping* common subgraph:

Definition 4.6.2 (overlapping candidate) *Let $Subgrf(G_1, G_2)$ be the set of common subgraphs of the attributes graphs G_1 and G_2. Then, $S \in Subgrf(G_1, G_2)$ is called a* best mapping *common subgraph of G_1 and G_2 if*

$$Diff_S(G_1, G_2) \le Diff_W(G_1, G_2), \forall\, W \in Subgrf(G_1, G_2)$$

such that there exists a graph isomorphism $\varphi : S \to W$ between S and W.

$S \in Subgrf(G_1, G_2)$ is an overlapping candidate *if for each S_s subgraph of S there exists another subgraph S'_s of S and a graph isomorphism $\varphi_s : S_s \to S'_s$ such that S'_s is a best mapping common subgraph.*

$S \in Subgrf(G_1, G_2)$ and overlapping candidate is called arc-overlapping *if there exists no other overlapping candidate common subgraph of G_1 and G_2 that has more nodes that S.*

By definition 4.6.2, can be observed that if a maximal common subgraph is overlapping candidate, it is also arc-overlapping, while the vice-versa it is generally false. In fact, every arc-overlapping common subgraph is topologically maximal in the set of possible overlapping candidates but it may not be a maximum common subgraph as defined in A.0.7. Analogously to the definition of optimal common subgraph in [Mes97] also the notion of arc-overlapping common subgraph emphasizes the graph attributes instead of the graph topology. However some drawbacks are limited by introducing the constraint that the overlapping candidate common subgraph has to be topologically maximal. Analogously the maximum common subgraph, also the arc-overlapping common subgraph is never empty and, indeed, it contains at least the best edge pair.

Once introduced the notions of optimal graph matchings, in the reminder of this section it will be discussed the existence and the behavior of possible distances between attributed graphs, focusing on their mathematical properties as shown in section 1.3. In fact, according to the mathematical statements, it would be desirable that a distance measure is a metric.

Edit distance measures, such that mentioned in the definition 4.11, minimize the cost of the editing operations but they are metric only if the cost of the underlying edit operations satisfy certain conditions. Therefore, they are largely used as measures [Mes97, MB98, ZTS02] but usually do not verify the triangular inequality property. In addition distance measures that are defined for tree-like structures are not longer valid for graphs. Therefore, in literature there it is very difficult to find a definition of distance between attributed graphs, which is also a metric and, generally, measures between graphs that satisfy the triangular inequality do not consider either edge or node attributes. An example of the distance measure which is a metric has been proposed in [BS98]:

$$d_B(G_1, G_2) = 1 - \frac{|maxcs(G_1, G_2)|}{max(|G_2|, |G_2|)}, \tag{4.12}$$

where $maxcs(G_1, G_2)$ is the maximal common subgraph of G_1 and G_2 and $|\ldots|$ represents the number of nodes of a graph.

Such a distance depends on the size of the subgraph. However, it does not take into account the attribute values of edges and nodes. This implies that each node of the subgraph has the same weight, despite its relevance in the graph. Moreover, such a measure does not verify the uniqueness property: in fact the distance between two graphs which are topologically isomorphic is always zero, even if they differ from some edge attributes. For instance, the distance d_B does not distinguish the two graphs in figure 4.15. In addition, has to be observed that such a distance does not measure how the sub-parts that correspond through the graph isomorphism effectively overlap; therefore, it is not the best choice in application tasks that are related to retrieval of partial common sub-parts.

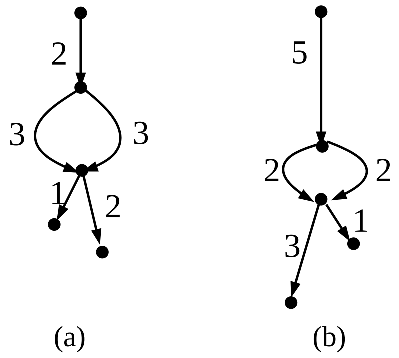


Figure 4.15: *The two graphs are isomorphic but not identical. The numbers indicate the value of the edge attributes.*

An adjustment of the distance is necessary to reflect also the differences of the graph attributes. In order to simplify the discussion on the new similarity measure, only edge attributes are considered. To adapt d_B to this purpose, the idea is to compute the distance on the edges of the graph, correcting the contribution of each edge with the "normalized" difference between the attributes of the corresponding edges of G_1 and G_2. To guarantee the contribution of each edge is a real value in the interval $[0,1]$, a real coefficient α has been introduced in the definition of the distance measure in the following formula 4.14, which is independent on the choice of the graph pairs. In addition, such a coefficient should verify the conditions: $\alpha \geq 1$ and $\alpha \geq max_{e \in G}(\mu_E(e))$, $\forall G \in M_{Gset}$. Clearly the hypothesis that α should be bigger than the maximal edge attribute of all attributed graphs implies that α cannot be evaluated a priori, thus is not computable and has to be relaxed. Therefore, the value of α currently chosen corresponds to the maximum attribute value on the edges of all graphs in the database, that is

$$\alpha = max(max_{G_i \in D_{Gset}}(max_{e \in G_i}(\mu_E(e))), 1), \tag{4.13}$$

where $D_{Gset} \subset M_{Gset}$ represents a set of graphs and E_i is the set of edges of G_i.

Lemma 4.6.3 *Let G_1 and G_2 be two attributed graphs of a data set $D_{Gset} \in M_{Gset}$ and S_M their arc-maximizing common subgraph. According to the definition A.0.6, let $\psi_i : G_i \rightarrow S_M, i = 1, 2$ be the graph isomorphisms between the subgraphs of G_i and S_M.*

Then, the function $d : D_{Gset} \times D_{Gset} \rightarrow [0,1]$ defined as follows:

$$d(G_1, G_2) = 1 - \frac{\sum_{e \in S_M}(1 - \frac{|\mu_E(\psi_1^{-1}(e)) - \mu_E(\psi_2^{-1}(e))|}{\alpha})}{max(|G_1|, |G_2|)}, \tag{4.14}$$

where α is the normalizing coefficient in the database defined in 4.13, is a semi-metric, with respect to the definition 1.3.1 discussed in section 1.3.

Proof. Non-negativity, identify and symmetry properties follow from the definition of d.

1. **Non-negativity**: $d(G_1, G_2) \geq 0$; in fact, by definition of μ_E we have:

$$0 \leq \mu_E(\psi_1^{-1}(e)),\ \mu_E(\psi_2^{-1}(e)) \leq \alpha \Rightarrow 0 \leq |\mu_E(\psi_1^{-1}(e)) - \mu_E(\psi_2^{-1}(e))| \leq \alpha,$$

therefore

$$\sum_{e \in S_M}(1 - \frac{|\mu_E(\psi_1^{-1}(e)) - \mu_E(\psi_2^{-1}(e))|}{\alpha}) \leq \sum_{e \in S_M}(1) = |S_M| \leq max(|G_1|, |G_2|).$$

2. **Identity**: we have to evaluate $d(G_1, G_2)$ when $G_1 \cong G_2$. Since S_M is the arc-maximizing common subgraph it follows that $S_M \cong G_1$, $S_M \cong G_2$ and the isomorphisms ψ_1 and ψ_2 correspond to the identity.

Therefore,

$$|\mu_E(\psi_1^{-1}(e)) - \mu_E(\psi_1^{-1}(e))| = 0, \forall e \in G_1$$

and

$$\sum_{e\ inS_M}(1) = |S_M| = |G_1| \Rightarrow d(G_1, G_1) = 1 - \frac{|G_1|}{max(|G_1|, |G_1|)} = 0$$

3. **Symmetry**: $d(G_1, G_2) = d(G_2, G_1)$ by relation 4.14 and graph isomorphism properties.

∎

Through analogous considerations the following lemma can be demonstrated:

Lemma 4.6.4 *Let G_1 and G_2 be two attributed graphs of a dataset $D_{Gset} \in M_{Gset}$ and S_M their arc-overlapping common subgraph. Then, the distance 4.14 defined in lemma 4.6.3 is a pseudo-metric.*

Both statements 4.6.3 and 4.6.4 claim that the function 4.14 do not satisfy the triangle inequality, thus they do not solve the problem of finding a metric between attributed graphs. In particular, has been observed that the notions of arc-overlapping and arc-maximizing correspond to two different ideas of optimality: that with respect to the attributes or that with respect to the topology of the graphs, respectively. In addition, given two attributed graphs G_1 and G_2, the set of their maximal common subgraphs that are also overlapping candidate may be empty. However, if the distance d 4.14 is defined on attributes graphs for which such a set exists, it is a metric. The following statement provides a condition, which is sufficient to guarantee that d verifies the triangular inequality.

Lemma 4.6.5 *Let G_1, G_2 and $G_3 \in D_{Gset}$ be three non-empty attributed graphs of a dataset $D_{Gset} \in M_{Gset}$ and $S_{1,2}$ be a maximal and overlapping candidate common subgraph of G_1 and G_2. Let $S_{1,3}$ and $S_{2,3}$ be defined analogously to $S_{1,2}$.*

Then, the distance 4.14 defined in lemma 4.6.3 verifies the triangle inequality: $d(G_1, G_2) + d(G_2, G_3) \geq d(G_1, G_3)$

Proof. The proof of the triangle inequality extends the one proposed in [BS98] and requires that the common subgraphs $S_{i,j}$, $1 \leq i < j \leq 3$ are both maximal and overlapping candidates (and therefore also arc-overlapping as previously described).

In practice, it has to verify the following relation:

$$1 - \frac{\sum_{e \in S_{1,2}} (1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha})}{max(|G_1|, |G_2|)} + 1 - \frac{\sum_{e \in S_{2,3}} (1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha})}{max(|G_2|, |G_3|)} \geq$$

$$1 - \frac{\sum_{e \in S_{1,3}} (1 - \frac{|\mu_E(\psi_{13}^{-1}(e)) - \mu_E(\psi_{33}^{-1}(e))|}{\alpha})}{max(|G_1|, |G_3|)} \tag{4.15}$$

where $S_{i,j}$, $1 \leq i < j \leq 3$ are defined like in the hypotheses and $\psi_{11} : \Delta(G_1) \rightarrow S_{1,2}$, $\psi_{21} : \Delta(G_2) \rightarrow S_{1,2}$, $\psi_{22} : \Delta(G_2) \rightarrow S_{2,3}$, $\psi_{32} : \Delta(G_3) \rightarrow S_{2,3}$, $\psi_{13} : \Delta(G_1) \rightarrow S_{1,3}$, $\psi_{33} : \Delta(G_3) \rightarrow S_{1,3}$ are the corresponding subgraph isomorphisms.

Two cases are distinguished:

(A) The common subgraphs $S_{1,2}$ and $S_{2,3}$ are disjoint on G_2. That is there are no arcs of G_2 that are in the domain of ψ_{21} and ψ_{23} at the same time. In this case we will demonstrate that a stronger relation holds:

$$1 - \frac{\sum_{e \in S_{1,2}}(1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha})}{max(|G_1|, |G_2|)} + 1 - \frac{\sum_{e \in S_{2,3}}(1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha})}{max(|G_2|, |G_3|)} \geq 1.$$

That is:

$$1 - \frac{\sum_{e \in S_{1,2}}(1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha})}{max(|G_1|, |G_2|)} - \frac{\sum_{e \in S_{2,3}}(1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha})}{max(|G_2|, |G_3|)} \geq 0.$$

Equivalently:

$$\begin{aligned} & max(|G_1|, |G_2|)\; max(|G_2|, |G_3|) \\ & -max(|G_2|, |G_3|) \sum_{e \in S_{1,2}} (1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha}) \\ & -max(|G_1|, |G_2|) \sum_{e \in S_{2,3}} (1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha}) \geq 0. \end{aligned} \tag{4.16}$$

More explicitly form, six cases have to be analyzed.

1. Case: $|G_1| \geq |G_2| \geq |G_3|$.

   The left-hand side of inequality 4.16 becomes:

$$\begin{aligned} & |G_1|\; |G_2| - |G_2| \sum_{e \in S_{1,2}} (1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha}) \\ & -|G_1| \sum_{e \in S_{2,3}} (1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha}) \geq \end{aligned}$$

   and, since $|G_1| \geq |G_2|$,

$$\begin{aligned} & |G_1| \Bigg(|G_2| - \sum_{e \in S_{1,2}} (1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha}) - \\ & \sum_{e \in S_{2,3}} (1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha}) \Bigg) \geq \\ & |G_1|(|G_2| - |S_{1,2}| - |S_{2,3}|) \end{aligned} \tag{4.17}$$

   The latter inequality holds because, by the definition of α, we have

$$(1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha}) \leq 1$$

and, therefore,

$$\sum_{e \in S_{1,2}} (1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha}) \leq |S_{1,2}|$$

and, analogously,

$$\sum_{e \in S_{2,3}} (1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha}) \leq |S_{2,3}|.$$

Finally, by the hypothesis that $S_{1,2} \cap S_{2,3} = \emptyset$, we can conclude that $|S_{1,2}| + |S_{2,3}| \leq |G_2|$, that is $|G_2| - |S_{1,2}| - |S_{2,3}| \geq 0$. $\Rightarrow Q.E.D.$

2. Case: $|G_1| \geq |G_3| \geq |G_2|$

   With a similar reasoning as before we obtain the relation:

$$|G_1| \left(|G_3| - \sum_{e \in S_{1,2}} (1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha}) - \sum_{e \in S_{2,3}} (1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha}) \right) \geq |G_1|(|G_3| - |S_{1,2}| - |S_{2,3}|) \tag{4.18}$$

   Therefore, since $|G_3| \geq |G_2| \geq |S_{1,2}| + |S_{2,3}|$ we have $|G_1|(|G_3| - |S_{1,2}| - |S_{2,3}|) \geq 0$.

3. Analogously we can demonstrate the remaining four cases: $|G_2| \geq |G_1| \geq |G_3|$, $|G_2| \geq |G_3| \geq |G_1|$, $|G_3| \geq |G_2| \geq |G_1|$ and $|G_3| \geq |G_1| \geq |G_2|$.

(B) The intersection S_M on G_2 of the subgraphs $S_{1,2}$ and $S_{2,3}$ is not empty.

Since $S_{1,2}$, $S_{2,3}$ and $S_{1,3}$ are all maximal common subgraphs, it follows that there exists at least a subgraph of $S_{1,3}$, which is isomorphic to S_M; we denote $Sub(S_{1,3}, S_M)$ the set of subgraphs of $S_{1,3}$ that are isomorphic to S_M. In addition, the hypothesis that $S_{1,3}$ is an overlapping candidate implies that there is a subgraph $S_{1,3}^M$ of $S_{1,3}$, $S_{1,3}^M \in Sub(S_{1,3}, S_M)$ such that the following relations are verified:

$$Diff_{S_{1,3}^M}(G_1, G_3) \leq \sum_{e \in S_M} |\mu_E(\psi_{13}^{-1}(e)) - \mu_E(\psi_{33}^{-1}(e))| \tag{4.19}$$

$$Diff_{S_{1,3}^M}(G_1, G_3) \leq \sum_{e \in S_M} |\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{23}^{-1}(e))|. \tag{4.20}$$

In particular, $|S_M| = |S_{1,3}^M|$.

Since $S_M = (S_{1,2} \cap S_{2,3})$ and $|\dots|$ is a metric in $\mathbb{R}$, for every edge $e \in S_M$ we have the relation:

$$|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{23}^{-1}(e))| \le |\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))| + |\mu_E(\psi_{23}^{-1}(e)) - \mu_E(\psi_{23}^{-1}(e))|.$$

By summing the terms of the previous inequality and using the inequality 4.20 we obtain:

$$Diff_{S_{1,3}^M}(G_1, G_3) \le Diff_{S_M}(G_1, G_2) + Diff_{S_M}(G_2, G_3). \tag{4.21}$$

Relation 4.21 will be useful in the following, to deduce the triangular inequality.

Then, we observe that the following inequality holds:

$$1 - \frac{(|S_M| - \frac{Diff_{S_{1,3}^M}(G_1,G_3)}{\alpha})}{max(|G_1|, |G_3|)} \ge 1 - \frac{\sum_{e \in S_{1,3}}(1 - \frac{|\mu_E(\psi_{13}^{-1}(e)) - \mu_E(\psi_{33}^{-1}(e))|}{\alpha})}{max(|G_1|, |G_3|)}$$

because $|S_{1,3}^M| \le |S_{1,3}|$ and

$$\sum_{e \in S_{1,3}} (1 - \frac{|\mu_E(\psi_{13}^{-1}(e)) - \mu_E(\psi_{33}^{-1}(e))|}{\alpha}) \ge \sum_{e \in S_{1,3}^M} (1 - \frac{|\mu_E(\psi_{13}^{-1}(e)) - \mu_E(\psi_{33}^{-1}(e))|}{\alpha}) \ge$$

(by the relation 4.19)

$$|S_M| - \frac{Diff_{S_{1,3}^M}(G_1, G_3)}{\alpha}.$$

Moreover, analogously to the case the sub-graphs $S_{1,2}$ and $S_{2,3}$ are disjoint, we will show that the following relation, which is stronger than 4.15, is verified:

$$1 - \frac{\sum_{e \in S_{1,2}}(1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha})}{max(|G_1|, |G_2|)} + 1 - \frac{\sum_{e \in S_{2,3}}(1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha})}{max(|G_2|, |G_3|)} \ge$$

$$1 - \frac{(|S_M| - \frac{Diff_{S_{1,3}^M}(G_1,G_3)}{\alpha})}{max(|G_1|, |G_3|)}$$

which implies that the triangular inequality is verified.

Equivalently, we will consider the following inequality:

$$\begin{aligned}
&max(|G_1|, |G_2|)max(|G_2|, |G_3|)max(|G_1|, |G_3|) - \\
&max(|G_2|, |G_3|)max(|G_1|, |G_3|) \sum_{e \in S_{1,2}} (1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha}) - \\
&max(|G_1|, |G_2|)max(|G_1|, |G_3|) \sum_{e \in S_{2,3}} (1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha}) + \\
&max(|G_1|, |G_2|)max(|G_2|, |G_3|) \left(|S_M| - \frac{Diff_{S_{1,3}^M}(G_1, G_3)}{\alpha} \right) \ge 0
\end{aligned} \tag{4.22}$$

Also in this case, we will distinguish six cases, according to the number vertexes of each graph:

1. Case: $|G_1| \geq |G_2| \geq |G_3|$

   In this case the expression 4.22 becomes:

$$|G_1||G_1||G_2| - |G_1||G_2| \sum_{e \in S_{1,2}} (1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha}) -$$

$$|G_1||G_1| \sum_{e \in S_{2,3}} (1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha}) +$$

$$|G_1||G_2| \left(|S_M| - \frac{Diff_{S^M_{1,3}}(G_1, G_3)}{\alpha} \right) \geq$$

   (by simplifying $|G_1|$ and considering that $|G_1| \geq |G_2|$)

$$|G_1||G_2| - |G_1| \sum_{e \in S_{1,2}} (1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha}) -$$

$$|G_1| \sum_{e \in S_{2,3}} (1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha}) + |G_1| \left(|S_M| - \frac{Diff_{S^M_{1,3}}(G_1, G_3)}{\alpha} \right) =$$

$$|G_1||G_2| - |G_1|(|S_{1,2}| + |S_{2,3}| - |S_M|) +$$

$$|G_1| \left(\sum_{e \in S_{1,2}} \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha} + \right.$$

$$\left. \sum_{e \in S_{2,3}} \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha} - \frac{Diff_{S^M_{1,3}}(G_1, G_3)}{\alpha} \right) \geq$$

   (being $|S_{1,2}| + |S_{2,3}| - |S_M| \leq |G_2|$)

$$|G_1| \left(\frac{Diff_{S_{1,2}}(G_1, G_2)}{\alpha} + \frac{Diff_{S_{2,3}}(G_2, G_3)}{\alpha} - \frac{Diff_{S^M_{1,3}}(G_1, G_3)}{\alpha} \right) \geq 0. \tag{4.23}$$

   Finally, by the relation 4.21, the inequality 4.23 is satisfied.

2. Case: $|G_1| \geq |G_3| \geq |G_2|$

   In this case the inequality 4.22 becomes:

$$|G_1||G_3| - |G_3| \sum_{e \in S_{1,2}} (1 - \frac{|\mu_E(\psi_{11}^{-1}(e)) - \mu_E(\psi_{21}^{-1}(e))|}{\alpha}) -$$

$$|G_1| \sum_{e \in S_{2,3}} (1 - \frac{|\mu_E(\psi_{22}^{-1}(e)) - \mu_E(\psi_{32}^{-1}(e))|}{\alpha}) +$$

$$|G_3|\left(|S_M| - \frac{Diff_{S^M_{1,3}}(G_1,G_3)}{\alpha}\right) \geq$$

with analogous reasoning as before we obtain:

$$\left(\frac{Diff_{S_{1,2}}(G_1,G_2)}{\alpha} + \frac{Diff_{S_{2,3}}(G_2,G_3)}{\alpha} - \frac{Diff_{S^M_{1,3}}(G_1,G_3)}{\alpha}\right) \geq 0$$

and

$$|G_1||G_3| - |G_1|(|S_{1,2}| + |S_{2,3}| - |S_M|) + |G_1| \geq 0$$

because $|G_1||G_3| \geq |G_1||G_2|$ and $|G_2| \geq (|S_{1,2}| + |S_{2,3}| - |S_M|)$

3. The remaining four cases are verified in analogous way.

■

Finally, has to be observed that the measure is an extension of the metric 4.12 proposed in [BS98], in fact when the edges of G_1 mapped in S_M have the same attribute values of those ones in G_2, it follows that

$$|\mu_E(\psi_1^{-1}(e)) - \mu_E(\psi_2^{-1}(e))| = 0, \forall\, e \in S_M$$

and

$$d(G_1,G_2) = 1 - \frac{\sum_{e\in S_M}(1)}{max(|G_1|,|G_2|)} = 1 - \frac{|S_M|}{max(|G_1|,|G_2|)} = d_B(G_1,G_2).$$

When the hypotheses of lemma 4.6.5 are satisfied, the distance 4.14 is a metric.

4.7 Discussion

The framework for graph matching proposed in this thesis is valid not only for Reeb graphs but also for each graph-like representation directed and a-cyclic. In particular, in the digital context, the shock graphs described in section 3.1.2, the graph extracted from the volumetric thinning described in section 3.2.1, moreover the component tree [Jon99], the max-tree [SOG98] and the topological graph proposed in [SL01], seem to be natural candidates.

The algorithm proposed in figure 4.4 compute the maximum common subgraph between two structural descriptors. Its correctness has been shown in section 4.2.2 and it assures that the node mapping produced during the expansion process of the algorithm always produces a common subgraph with respect to the definition A.0.6 (theorem 4.2.2), moreover the maximum common subgraph is always computed (theorem 4.2.3).

The heuristic techniques described in section 4.3 approximate the maximum common subgraph by constructing a not necessarily connected common sub-graph, which is able to detect and map together similar parts of the model object (partial matching), and makes the algorithm robust with respect to slight structural and topological deformation. Therefore, the proposed approach should not be considered just a coarse filter for shape comparison, rather a finer shape analysis tool where structure and topology are taken into account. Moreover, even if the adopted matching approach is mainly based on the structural information stored in the graph, node and edges can be endowed with attributes aimed at describing the geometry of the sub-parts associated to the subgraphs of the descriptor, increasing the discriminant power of the matching algorithm.

A new similarity measure (equation 4.14) between attributed graphs has been defined in order to enrich the similarity measure of the equation 4.12 with an evaluation of the attributes associated to the edges of the two graphs, this measure is a pseudo-metric with respect to the definition 1.3.1 discussed in section 1.3. Currently the challenge is to find conditions that guarantee the measure to be a metric, since the hypotheses of lemma 4.6.5 are too strong for a real application in the database retrieval context and they have been used only marginally in the proof of lemma 4.6.5.

As shown in the definition A.0.6, the common subgraph obtained as output of the matching algorithm corresponds to a subgraph isomorphism of the two input graphs. From such isomorphism it can be computed the set of the graph editing operations able to transform one of the two input graph into the other input graph. The edit operations involved in the transformation process are the addition and the deletion of nodes and edges, as well as, the transformation of the attributes associated to nodes and edges belonging to the common subgraph. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two graphs, and let f be the subgraph isomorphism between $\mathcal{G}_1{}'$ and $\mathcal{G}_2{}'$, both subgraphs of $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively. Transforming $\mathcal{G}_1$ into $\mathcal{G}_2$ means:

- to compute the subgraph isomorphism f and the two common subgraph identified by $\mathcal{G}_1{}'$ and $\mathcal{G}_2{}'$;

- to remove (delete edit operation) the nodes and the edges of $\mathcal{G}_1$ belonging to the graph $\mathcal{G}_1$-$\mathcal{G}_1{}'$, representing the nodes and edges of $\mathcal{G}_1$ not involved in f;

- to add (addition edit operation) to $\mathcal{G}_1$ the nodes and edges of $\mathcal{G}_2$-$\mathcal{G}_2{}'$, representing the nodes and edges of $\mathcal{G}_2$ not involved in f;

- the attributes of the nodes and edges of $\mathcal{G}_1{}'$ have to be transformed into the attributes of nodes and edges of $\mathcal{G}_2{}'$.

This kind of transformation process between graphs is inspired by the error tolerant graph isomorphism (see definition A.0.8) and coupled with an algorithm for the reconstruction of objects from the structural descriptor [BMS00], it can be used for metamorphosis purposes between three dimensional objects.

Related publications

S. Marini, M. Spagnuolo, and B. Falcidieno. From exact to approximate maximum common subgraph. In *Proceedings of the 5rd IAPR -TC-15 Workshop on Graph-based Representations in Pattern Recognition*, Lecture Notes in Computer Science, Poitiers (France), April 11-13 2005.

S. Biasotti and S. Marini. 3D object comparison based on shape descriptors. *International Journal of Computer Applications in Technology*, 23(2/3/4):57–69, 2005.

Marini, S.: Methods for common subgraph approximation. Technical Report 23/2004, Istituto di Matematica Applicata e Tecnologie Informatiche, Consiglio Nazionale delle Ricerche, Genova (2004)

Biasotti, S., Marini, S.: 3D object comparison based on shape descriptors. Technical Report 17/04, Istituto di Matematica Applicata e Tecnologie Informatiche, Consiglio Nazionale delle Ricerche, Genova (2004)

S. Biasotti, S. Marini, M. Mortara, G. Patané, M. Spagnuolo, and B. Falcidieno. 3D shape matching through topological structures. In I. Nyströn, G. Sanniti di Baja, and S. Svennson, editors, *Proceedings of the* 11th *Discrete Geometry for Computer Imagery Conference*, volume 2886 of *Lecture Notes in Computer Science*, pages 194–203, Naples, 2003. Springer Verlag.

S. Biasotti, S. Marini, M. Mortara, and G. Patané. An overview on properties and efficacy of topological skeletons in shape modeling. In *Proceedings of Shape Modelling and Applications*, pages 245–254, Seoul, South Korea, June 2003. IEEE Press.

S. Marini. Topology, structure and form for understanding shape similarity. In *Proceedings of Eurographics Italian Chapter*, Università degli Studi di Milano, July 11–12 2002.

Chapter 5

Experiments and Results

Due to the recent improvements of 3D object acquisition, visualization and modeling technologies, the number of 3D models available on the web is more and more growing, therefore, there is an increasing demand for tools supporting the automatic search for 3D objects and their sub-parts in digital archives. While there are already existing techniques for rapidly extracting knowledge from massive volumes of texts (like Google [htt]) it is harder to structure filter, organize, retrieve and maintain archives of digital shapes like images, 3D objects, 3D animations and virtual or augmented reality. These considerations suggest that in the future a primary challenge in computer graphics will be how to find models having similar global and/or local appearance.

Concerning 3D shapes, there is a great number of techniques for shape matching. The methods so far developed, span from coarse filters suited to browse very large 3D repositories on the web, to domain-specific approaches for assessing similarity of part models containing semantic, as well as structural information. Although their effectiveness and efficiency strongly depend on the input representations and on the chosen target application, all of them can be fit a multi-step approach which considers a series of filters that progressively refine the set of geometrically similar candidates of a multi-modal query mechanism aimed at providing a combination of various shape descriptors and measures for the evaluation of similarities suitable for reasoning and analysis of 3D shapes.

5.1 Sub-Part Shape Correspondence

Aim of this section is to demonstrate the effectiveness of the structural shape descriptors discussed in chapter 3 and the comparison framework described in chapter 4, when applied to the problem of recognizing the sub-parts correspondence between 3D objects. Such correspondence, is achieved when structural and geometric similarity is locally maximized for sub-parts of two input objects. In figure 5.1 is shown an example of sub-part correspondence. Colors in figures 5.1.c) and 5.1.d) denote sub-parts classified with respect to the number of boundary components: red/one, orange/two and yellow more than two boundaries. In figures 5.1.e) and 5.1.f) the same colors between the models represent the sub-part correspondence.

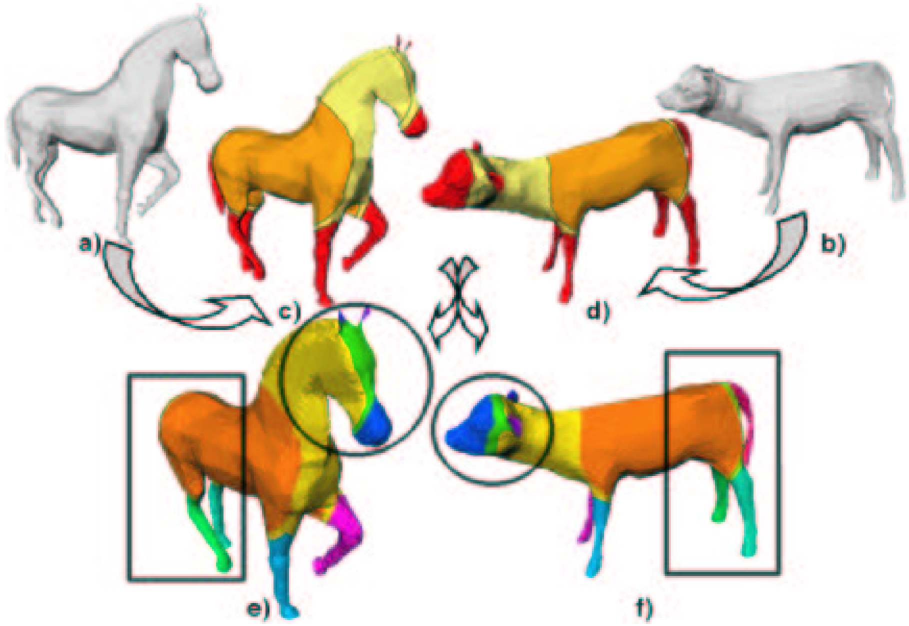


Figure 5.1: *Two models a) and b) are segmented c) and d) and then matched, through the approach proposed in chapter 4. The similar sub-parts correspondence is highlighted e) and f): for example the front and the rear parts.*

Since the correct correspondence depends on the semantic interpretation of the involved sub-parts, the problem may have no exact formulation[1]. Nevertheless it is not difficult to provide an intuitive validation of the results, especially when the two input objects are somehow classified similar[2].

One important aspect of the approach proposed in this chapter is the coupling between geometric and structural descriptors for identifying the maximal sub-parts having similar structure and geometry.

Traditional, methods proposed in the literature for shape retrieval [VSR01, NK01, OFCD01, KFR03] mainly consider geometric descriptors encoding the shape distribution of the object in the 3D space. These methods do not require that the models are connected and/or invariant to topological changes of the shape. Nevertheless, the shape structure is completely forgotten and the similarity distance between two objects depends only by their spatial embedding. Therefore, as remarked in chapter 3, advanced queries to a shape repository taking into account the object structure are not possible: for

[1] Although structural shape decomposition is the bridge between the geometrical and semantical level ([IST]), the models used for these experiments are not semantically annotated. This means that, even if the object is correctly decomposed (e.g. head, front legs, rear legs, tail and body), the semantic of each sub-part is not explicitly annotated, thus it can not be used in the comparison process. For this reason is not possible to provide a formal validation the correspondence.

[2] At this stage it is not important which method for the evaluation of similarity has been used to judge the two models similar: for example can be used both methods based on geometrical and structural shape descriptors.

example search for a model having at least one handle in order to be grasped by a human hand.

The work described in this section, is a method for deducing structural correspondences and similar sub-parts for 2-manifold 3D polygonal meshes described by the Reeb graph discussed in section 3.2.3.3, where its effectiveness in the shape matching context has been shown in [HSKK01, BRS03, BMM$^+$03]. To guarantee that the matching approach takes into account not only the object topology, attributes describing the geometry of the sub-parts have been associated to the node and edge of the Reeb graph. To accomplish this task the geometric descriptor proposed by Kazdan et al. [KFR03] and discussed in section 2.2.2, has been chosen as geometric descriptor.

Reasoning on shape structure allows the deduction of the most similar sub-parts, directly from the Reeb graph, by computing an approximation of the maximal common sub-graphs between the two attributed structural descriptors.

5.1.1 Previous work

Many methods for 3D object comparison return as output a positive real number which measures how much an object resembles to another one [Vel01, OFCD01, NK01, CRC$^+$02, VT03, KFR03]. Since no information on the sub-part correspondence of the compared objects is stored, these approaches are not available for partial matching among objects. On the contrary, such information is needed in applications like object modeling, registration and recognition [FKS$^+$04, RCSM03, HKDH04]. Mapping methods between sub-parts 3D object have been mainly used in application contexts dealing with object recognition. Several approaches recognize the class to which an input object belongs to, comparing it to models already grouped in that object-classes, other methods try to obtain sub-part mapping among slightly dissimilar objects.

The method proposed in [JH97] and discussed in section 2.1.2 does not require any feature extraction or segmentation; it samples the object surface into a set of oriented points (3D points with surface normals) and associates to each sampled point a description of the surface around it. This description corresponds to an image (spin-image) obtained from a local basis generated by an oriented point. The positions of other points on the surface with respect to this basis can be described by two parameters. By coding these parameters in a 2-D array, a descriptive image associated with the point is created (see figure 2.6 of section 2.1.2). During the recognition process, images from points on the model are compared (2.3, in particular the equation 2.9) with images from points in the scene; when two images are similar enough, a point correspondence between the model and the scene is established as shown in figure 5.2.

Other approaches like those described in [FGN89, RCSM03, HKDH04] represent 3D objects as a set of parts, which are compared to obtain an object sub-part mapping. In [FGN89] the surface of an object is described by segmenting it into patches; the complete surface description separately represents each patch and their interrelationships. Complex surfaces are segmented into simpler meaningful components (the so-called patches) through shape discontinuities, such as jump boundaries, limbs and creases. Therefore, such a description can be viewed as an attributed graph whose nodes correspond

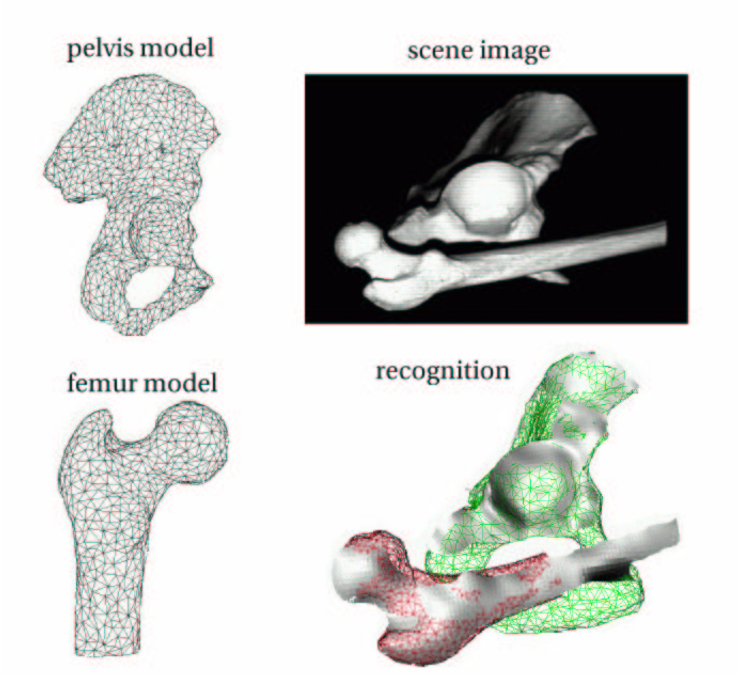


Figure 5.2: *Recognition between the models of a femur and a pelvis. The red and green points represent the sub-part correspondence between the two models and the scene. The figure is from [JH97].*

to the surface patches and the edges codify the relations between them, see figure 5.3. Each patch is discretized into a set of geometric descriptors such as the surface area, the average orientation, the

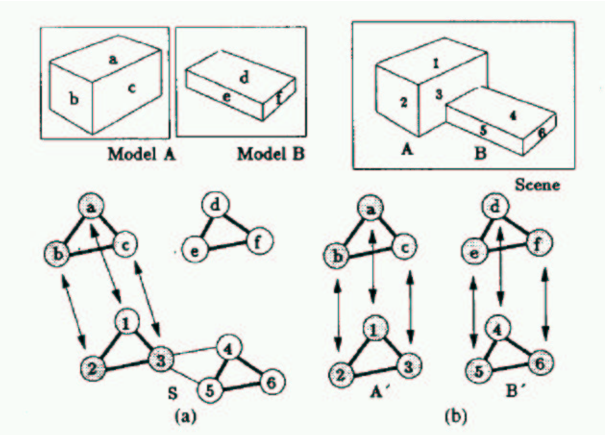


Figure 5.3: *Three simple objects and their sub-part correspondence. The figure is from [FGN89].*

average of principal curvatures and the centroid. Two nodes are connected by an edge if there exists at least one common boundary component between the two patches. Attributes associated to each edge are a label describing the type of the adjacency (convex or concave crease, jump or limb) and a real number R, $R \in [0, 1]$, representing the relevance of the connection. The comparison of two objects is accomplished through a graph matching process that starts enumerating all possible node pairs among the nodes of the two graphs. To choose the best pair, with respect to a set of constraints and a goodness measure, the process try to incrementally expand the mapping. The expansion ends when the set of node pairs is good enough with respect to the goodness measure and/or no further expansion is possible and/or if the number of node pairs exceed a given threshold.

The method proposed in [HKDH04] classifies scene sub-parts of 3D objects into a set of pre-determined object classes. The method consists of a learning phase and an execution phase. During the learning phase a set of 3D objects is used for training the system; this phase produces both a set of part classes and a relation between the sub-part classes and an object sub-parts. Each object is arbitraitly split into sub-parts, each one represented by a collection of semi-local signatures that capture the regional object shape at locations distributed over the part surface. In particular, each object is split exactly into three parts: front, middle and rear part. The proposed signature is a spin-image, as described in [JH97]. Parts with similar signatures are expected to have similar overall shape. All object sub-parts are automatically grouped into classes where a class is represented by the union of the signatures of the parts within that class, see figure 5.4.a). Finally, during the execution phase, novel parts may be mapped into objects belonging to the database using the same method adopted in the learning phase, see figure 5.4.b).

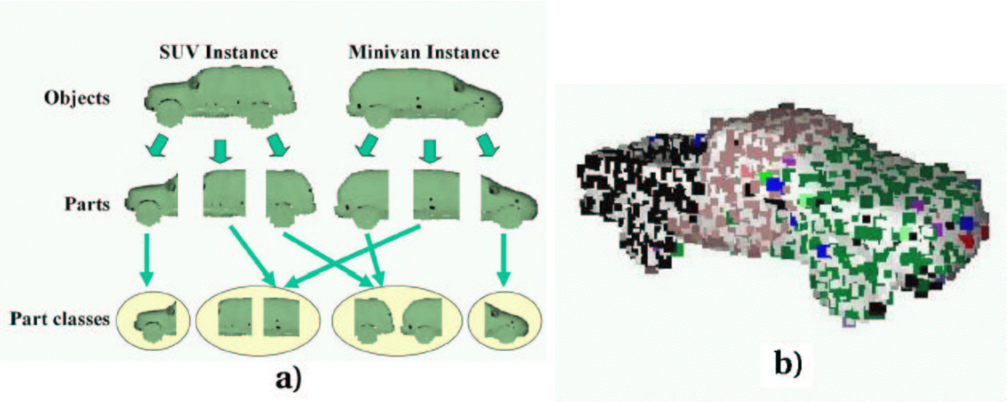


Figure 5.4: *a) Two objects divided into parts and grouped into part classes. b) An example query object with the sample points classified according to their most likely part class. The color coding shows that the part classes are consistently recognized. The figures are from [HKDH04].*

In [RCSM03] the use of the component detectors is trained to identify regions having similar shape across all instances of a class. Salient points are manually selected. These detectors provide a segmentation of a surface into components that can be regarded as parts. Then, symbolic surface detectors are trained to recognize the spatial relationship between labeled components. At run-time, the component and the symbolic surface detectors are used to find points on query scenes that are similar in shape to the salient points of the training objects.

Finally, the scheme in [FKS$^+$04] describes a methodology to interactively build new objects composing sub-parts of existing objects already stored in a database. The steps the user has to perform to build a new model involve several shape modeling skills: choosing a model from the database (that is a whole-object matching), selecting a part of the model to edit (i.e. interactive segmentation of 3D surfaces), executing a search of the database for similar parts (i.e. partial object matching), selecting one of the models returned by the search and performing editing operations in which parts are cut out from the retrieved model (i.e. interactive segmentation of 3D surfaces) and, finally, composing the current model (i.e. composition of parts to form new models). The comparison process is the most interesting for our purposes, because it matches global shapes with a special emphasis on a selected part. There, the notion of shape similarity is defined for models aligned in the same coordinate system. The similarity distance between two objects is the integral over an object surface of the squared Euclidean distances of the other surface. Then, the feature based matching is obtained associating a weight to each point and then accordingly scaling the contribution of each summand. In addition, the user may select parts (points with higher weight) that contribute to the measure of shape similarity more than others. In figure 5.5 the model of a statue has been used as query in a database. The blue and yellow boxes (sub-parts with higher weight) have been interactively put by an user to retrieve similar models having at least a left arm. Such a matching process is able to retrieve only objects well aligned in the same coordinate system of the statue, while other statues or human bodies that are not aligned with that will be discarded even if they have similar structure and, in particular, they have the left arm.

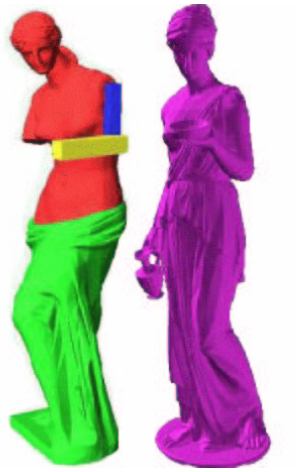


Figure 5.5: *The query model (left statue) and the retrieved result (right statue). The figure is from [FKS$^+$04].*

5.1.2 Problem Statement

The partial matching problem can be intuitively split in the following three sub-problems:

1. recognizing similar sub-parts in objects that are both structurally and geometrically similar (that is, having similar overall shape);
2. recognizing similar sub-parts in objects having different overall shape;
3. deducing if an object shape is itself a sub-part of another.

In particular, has to be observed that the sub-problem 3 is a special case of the sub-problem 2, while in the sub-problem 1, similar sub-parts of the two objects should be automatically recognized and mapped. The method proposed in this section is able to solve all the three sub-problems with major emphasis on the first one. For example, in figure 5.6.a) a mug and a teacup, which belong to the same class of cups with an handle, have been compared. These models have similar appearance and structure. Then, our structure-driven matching correctly recognizes and maps handles, internal and external parts of the two cups (regions in the two models that correspond are depicted with the same color).

The second kind of correspondence deals with objects having different overall shape but similar sub-parts. The expected partial correspondence should recognize similar sub-parts of the two objects and produce the correspondent mapping. In figure 5.6.b) the body and the handle of the pot have been recognized with respect to the whole object while the pot spout is not mapped.

The implementation of the matching process involves both structure and geometry of the two objects. The structural information of the object is captured by the Reeb graph computed with respect to the position invariant descriptor based on the integral geodesic distance proposed in [HSKK01] and discussed in section 3.2.3.3. Figure 5.7(a) shows an example of a 3D model described by such Reeb graph. As shown in [Bia04], the shape characterization and the Reeb graph construction proposed in [ABS03] naturally induce a decomposition of the shape in topologically significant regions, as shown in 5.7(b). The region decomposition obtained from that contouring approach does not admit slices with internal holes and each border component of the surface patches is shared by only two distinct patches. Since each border component is completely shared by two patches, cutting and pasting operations along such a contour may be performed independently from other contours. Moreover, such a segmentation produces a directed graph as discussed in section 4.1, in which each node corresponds to an object patch and each edge connects two paths. Edges are oriented according to the increasing direction of the mapping function that induces the surface characterization, see figure 5.7(c). Since such a graph is directed, each node identifies a subgraph and the geometric attribute associated to the node is obtained from the surface related to its subgraph, see figure 5.7.d).In addition, to each node of the graph is associated a geometric descriptor, which is obtained from the spherical harmonic analysis of the surface related to the subgraph. Therefore, main innovation of our approach with respect to the methods discussed into the section 5.1.1 is to combine a geometric and a structural descriptor of shape and to use a structural matching algorithm which is robust to noise.

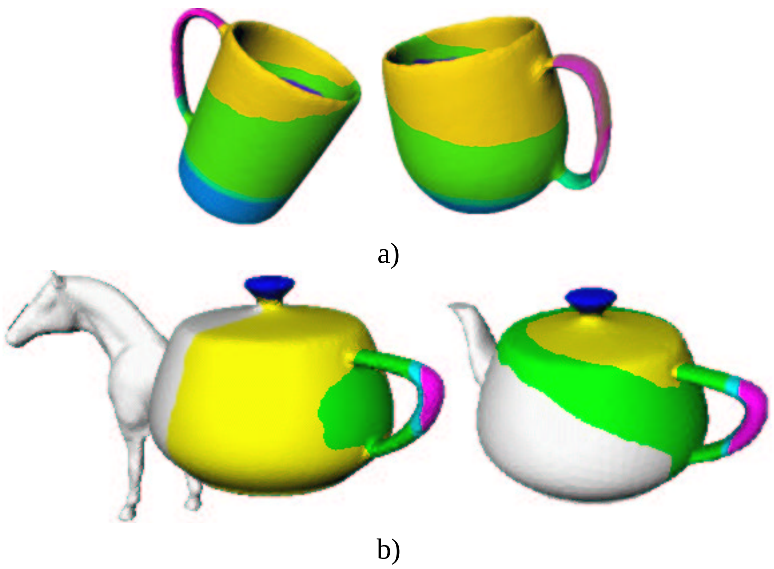


Figure 5.6: *The sub-part correspondence between a mug and a teacup a) and between a pot and an hybrid model b).*

5.2 Discussion

As discussed in section 5.1.2 the approach addresses the problem of finding the correspondence among model sub-parts of objects having or not the same overall shape. In this section some experimental results are provided and discussed.

In figure 5.8 the partial correspondence obtained comparing similar models, in our case three horses, is shown. Although the overall shape of the animals is the same, the models differ over some details: for example, the structure of the head, the tail and the posture. Our partial matching correctly recognizes the correspondence among the bodies and the front/rear sub-parts of the models. If some shape features have no correspondence in the other models (like the tail) they are not mapped at all. Nevertheless it could happen that features like the legs may be switched. This is caused by a lack of structural information into the leg description: each leg is represented by two nodes connected by an edge. In these cases the sub-parts mapping is completely demanded to the geometric descriptor that produces this output.

Other experimental results of the application of the matching method to models having same overall shape but different spatial embedding are shown in figure 5.9. The geodesic distance distribution on a human model does not change if the legs and the arms are stretched rather than curled up. Therefore, since the structural descriptor is independent of different poses of the same object, the algorithm (discussed in the chapter 4) can recognize human features like hands, head, legs and body in arbitrary positions, see figure 5.9. Results in figure 5.9 demonstrate that providing a significant structural decomposition is fundamental. Unfortunately, the geodesic distance does not solve every matching

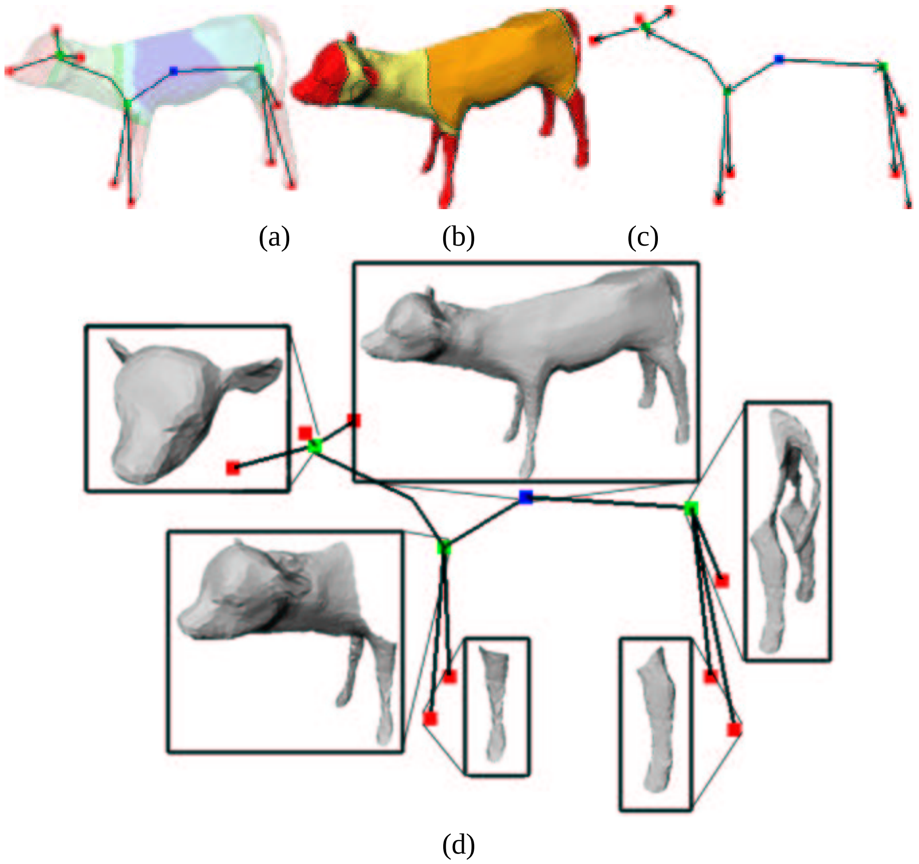


Figure 5.7: *A Reeb graph of a calf a), the surface segmentation associated b) and the oriented graph c). In d) the surface portions associated to some nodes are highlighted; these regions contain all patches associated to the subgraph nodes.*

problem; its sensitiveness in capturing salient features (discussed in section 1.5) may produce graph element that behave as noise in the comparison process. For example the model of a man with a big stomach deforms a lot when he is sitting or running. Such a local deformation may produce changes into object segmentation and thus into the structural representation of the body. This phenomenon may be partially solved increasing the robustness performance to structural noise of the matching algorithm.

An example of partial correspondence among models having different overall appearance is proposed in figure 5.10. In this case has been considered two models (a horse and a pot) belonging to two different classes and has been composed a new model replacing the spout of the pot with the front part of the horse. Then, the algorithm for graph matching has been used for analyzing similar sub-parts. As shown in figure 5.10 the correspondences are promising. Results are grouped in order to highlight the obtained mappings.

In particular, has to be observed that the front part of the hybrid model is correctly mapped with the

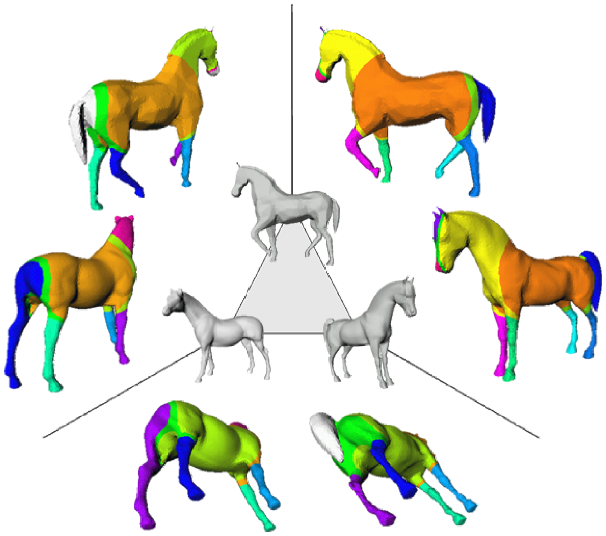


Figure 5.8: *Sub-part correspondence among three animals: these models have similar structure and geometry*

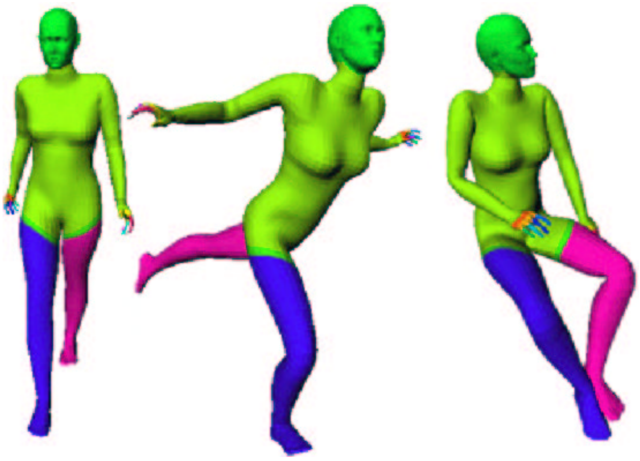


Figure 5.9: *Correspondence of shape features on a human model in different poses.*

original model and vice-versa. Analogously, the relevant features of the pot are correctly recognized in both the models. The colors show that the handle, the tip and the body of the pot and the mixed model are correctly mapped even if the model segmentation changes, as shown in figure 5.10. The edge/node correspondence between the pot and the mixed model is depicted in figure 5.11. In figure 5.12 an example where a whole model is a sub-part of another one is shown: the graph of the cow head is a subgraph of the cow graph. The matching algorithm computes the common subgraph reasoning on the graph structure and on the geometric attributes, resulting in a correct sub-parts correspondence between the two mouths, the ears and the horns. As in the case of the horse model compared to

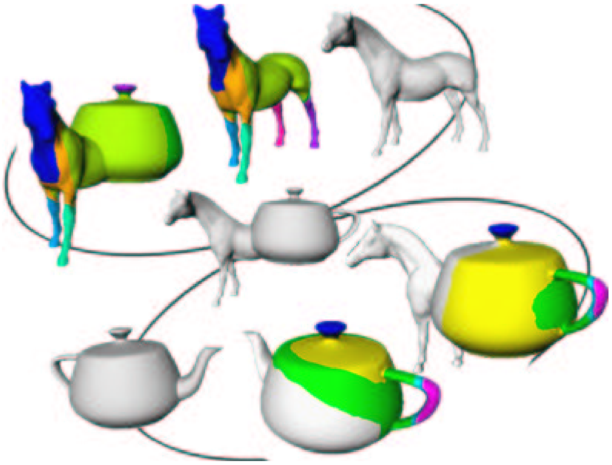


Figure 5.10: *Sub-part correspondences among an horse, a pot and the mixed model.*

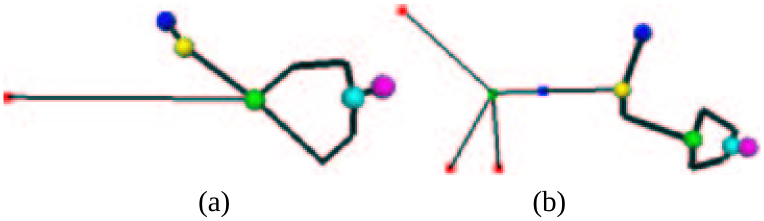


Figure 5.11: *Graph matching between the pot (a) and the mixed model (b)*

the mixed one, also the cow head compared to the whole cow produces some unexpected mappings. There, the matching algorithm generates a correspondence between the back part of the horse and the body of the mixed model (figure 5.10); this is caused by the behavior of the algorithm that looks for a maximal common subgraph. In this case, the algorithm expands as much as possible the initial mapping among the relevant nodes, including also dissimilar sub-parts into the common subgraph. Analogously, the front part of the cow head is correctly mapped, but its neck is mapped with a part of the body. This is intrinsic to the sub-parts correspondence problem. The cow head is described as a global object and the Reeb graph has a different edges orientation with respect of the head of the whole model. This fact may produce unexpected node mappings where the structural descriptor carries a small amount of information, that is the neck of the cow head.

5.3 Shape Retrieval

As for the experiments on the sub-part correspondence of the section 5.1, it is interesting to investigate the Reeb graphs structural descriptors discussed in the sections 3.2.3 and 3.4 applied to the shape retrieval application contexts. This is because the different behavior of the mapping function for the

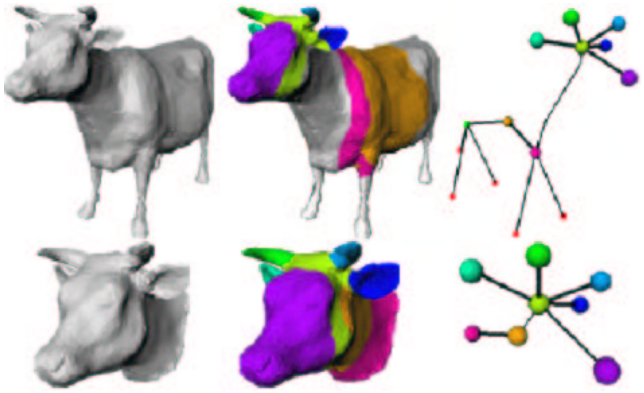


Figure 5.12: *Recognition of the head of a cow with respect to the whole animal model.*

Reeb graph representation shown in figure 3.23 on page 85, emphasizes different aspects of the object shape, and influence the results of the matching algorithm discussed in chapter 4, as shown in figure 5.13.

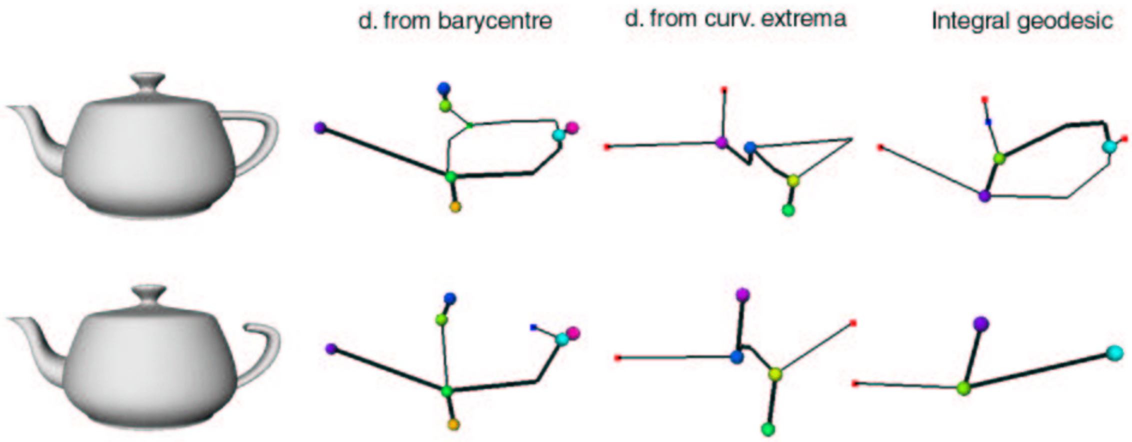


Figure 5.13: *Matching between the three Reeb representations of the teapot and of its modified version (see figure 3.23 on page 85) and their similarity evaluation. Thick arcs and nodes with same color and label represent the graph mapping.*

5.3.1 Problem Statement

At the conceptual level, a typical 3D shape retrieval framework consists of a repository of models with an index structure, created offline, and an on-line query engine. Each 3D model has to be identified with a shape descriptor that provides all the necessary information useful to characterize the salient and relevant aspects of the shape, in a concise and eventually scalable encoding of the shape, as discussed in section 1.4.

To efficiently search a large collection of models, an indexing data structure and a searching algorithm should be available. Their interaction is explained in the following:

- the on-line query engine computes the descriptor of the query model;
- models belonging to the repository and similar to the query model are retrieved by matching their shape descriptors to the query descriptor;
- The index structure is used to sped-up the search among the models of the repository. An example of indexing structure, based on the triangle inequality property of the dissimilarity measure, has been provided in section 1.3.

Usually three approaches can be distinguished to provide a query object:

- browsing the shape repository in order to select a query object from the obtained results;
- a direct query by providing a query descriptor;
- query by example by providing an existing 3D model or by creating a 3D shape query from scratch using a 3D tool or sketching 2D projections of the 3D model.

Finally, the retrieved models can be visualized in order to be submitted to the judgment of the user.

5.4 Discussion

The experiments have been done using the CAD models in the two shape repositories proposed in [BRS03, BSRS03][3]. Moreover at least one hundred models has been picked-up from the Internet and classified into a set of families dealing with human bodies, four-legs animals, furnitures, fishes, etc. Finally the algorithm is able to perform more than 10.000 graph comparisons in less than 10 seconds on a AMD Athlon 1GHz with 512Mb of RAM.

Since for manufacturing models, high curvature points may be not isolated and individuate sharp features, the function based on the distance from the curvature extrema does not seem to be a good choice as structural shape descriptors for shape retrieval in the CAD context. Two repositories of shape descriptors have been created: the first for Reeb graph with respect to the distance from the barycentre (section 3.2.3.2), and the second for the Reeb graph with respect to the integral geodesic distance (section 3.2.3.3).

Two experimental results obtained by applying the method for comparing graphs, described in chapter 4, to the two repositories of structural descriptors, are shown in figures 5.14 and 5.15. The results

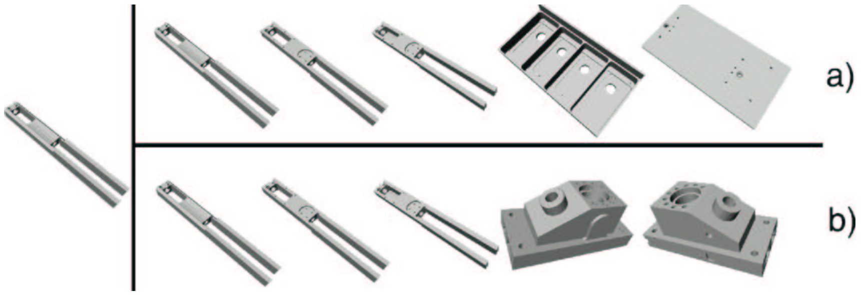


Figure 5.14: *Matching results for the linkage model.*

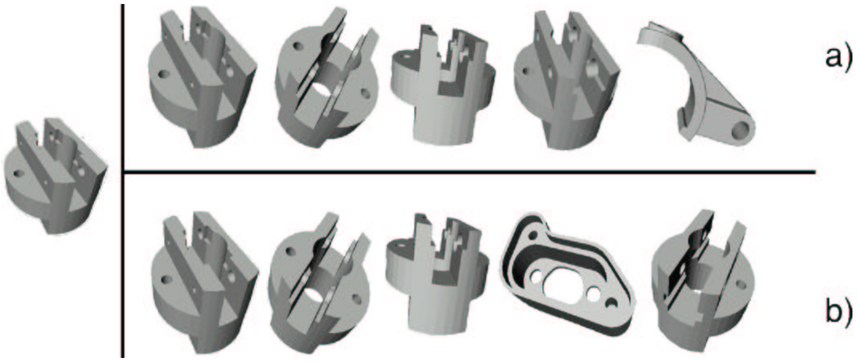
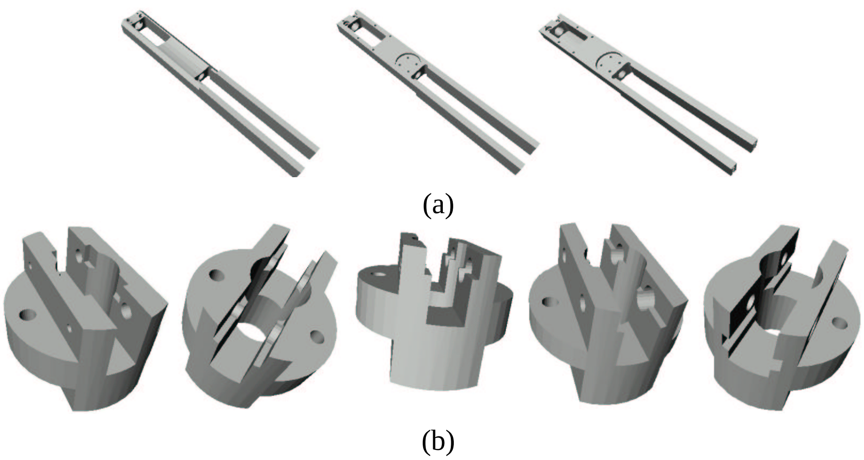


Figure 5.15: *Matching results for the socket model.*

show the first five best matching arranged according to their similarity values, with respect to the query models (the linkage and socket mechanical parts), in decreasing order from left to right. In both the figures the line a) corresponds to the query performed on the repository of descriptors based on the distance from the barycentre, while the line b) corresponds to query performed on the repository of descriptors based on integral geodesic distance.

Experimenting over the two repositories, has been noticed that each model has been correctly recognized as member of the family of objects it belongs to, even if some false positives may occur. In figure 5.16 are shown the families of the linkage (a) and the sockets (b) mechanical parts. As can be noticed, the results obtained by querying the repository for the linkage model (figure 5.14), correctly recognize all the models belonging to its family, in the first three top positions, while the choice of the other two models depends on the function. In particular, the distance from the barycentre (figure 5.14(a)) favors the choice of models whose shape is lengthened as the shape of the linkage, while the integral geodesic distance (figure 5.14(b)) selects objects having similar features, even if spatially distributed in a different manner. This fact is further emphasized for the socket model (figure 5.15), where the fourth object retrieved, which has the same number of holes and the same smoothed appearance of the query model, is preferred to a socket with a different number of holes.

[3] http://www.designrepository.org/SM03 and http://www.designrepository.org/DECT03.



(a)

(b)

Figure 5.16: *The families of the linkage models (a) and of the sockets models (b).*

In Figure 5.17, are shown the Reeb graphs of two objects (pictures (a) and (d)) with respect to the distance from the barycentre (pictures (b) and (e)) and with respect to the integral geodesic distance (pictures (c) and (f)). The two mechanical parts are almost identical, apart the sub-parts highlighted in the circles: smoothed corner for the model depicted in (a) and sharp corner for the model depicted in (d). As for the handles of the teapots shown in figure 3.23 on page 85, the Reeb graph based on the integral geodesic function captures, as a protrusion, the sub-parts highlighted in the circles of the models (a) and (d), but they are not able to describe its smoothness or sharpness (see graphs (c) and (f)). On the other hand this difference is captured by the Reeb graph based on the distance from the barycentre, as shown by the graphs in (b) and (e): the graph represented in (e) has four edges, adjacent to the nodes **g** and **i** that do not occur in the graph (b).

In figure 5.18, the best matching objects retrieved by the two query models (a child and a dog) are shown. Results are arranged according their similarity value with respect to the query models, in decreasing order from left to right. Differently from the CAD models, three Reeb graphs representations has been compared: line (a) corresponds to the distance from the barycentre, line (b) to the integral geodesic distance and (c) to the geodesic distance from curvature extrema. For each function the best match was the model itself and was not depicted. Also in this case the results reflects the intuitive notion of similarity and groups the objects in a set of families (for instance quadrupeds, humans, pots, hands, etc.) even if some false positive results are obtained. This phenomenon rises more frequently when the graph representation of the query model is very simple, both in terms of number of nodes and edges and of configuration: the query graph itself is easily contained into other graph representations of the models in the repository.

The use of structural descriptors to describe/represent model features allows a good representation both of topology and structural aspects of the shape. The ability of taking into account both topological, structural and geometrical aspects of the model shape strongly depends on the comparison process adopted: as shown in figure 5.17 a correct sub-part correspondence, as discussed in section 5.1, gives

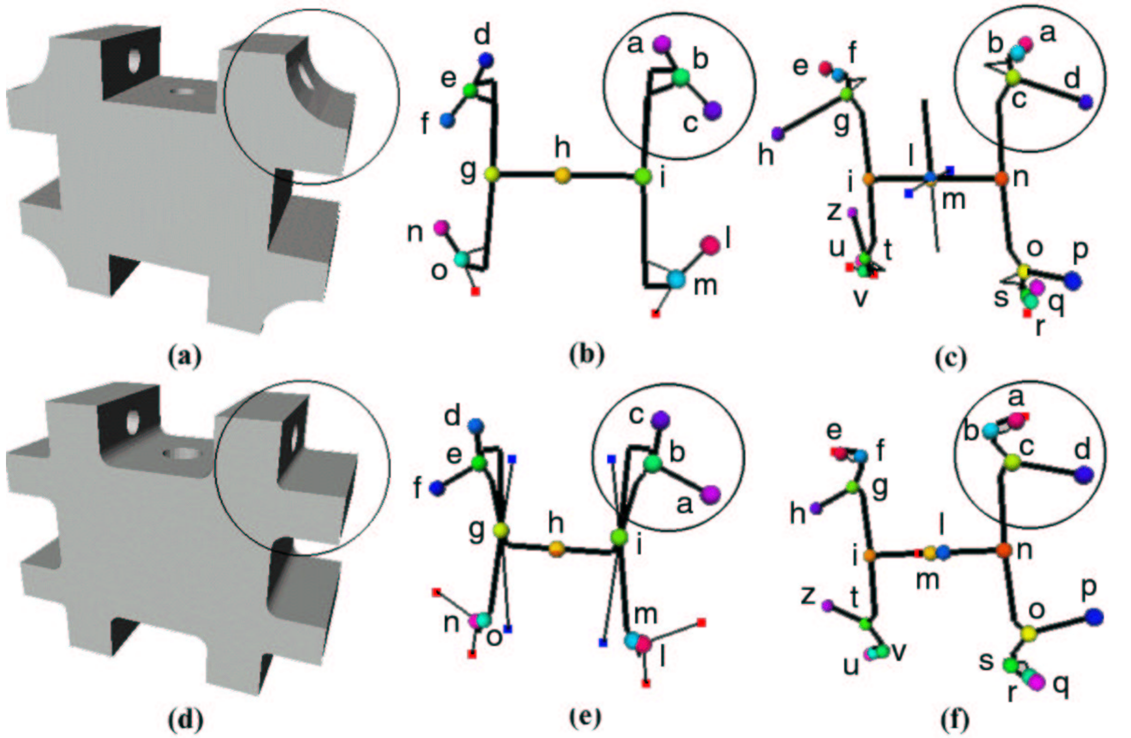


Figure 5.17: *Matching of the Reeb graphs of the objects (a) and (d) with respect to the distance from the barycentre (b) and (e) and the integral geodesic distance (c) and (f). Thick arcs and nodes with same color and label represent the graph mapping.*

to the shape comparison methodology an high discriminating power. Finally, has to be observed that mechanical models may differ from small features, number of holes or geometric aspects of their features like smoothness or sharpness, etc. The experiments on shape retrieval, provided in this section, prove the effectiveness of the structural descriptors (section 3.2.3) and the effectiveness of the graph matching algorithm (chapter 4), grouping objects with similar shape and also with respect to the small differences typical of the mechanical parts.

A statistical validation of the results obtained in the shape retrieval applications, is the precision/recall curve [Rij79]. The precision and recall descriptors attempt to measure the effectiveness of the retrieval methodology, measuring the ability of the system to retrieve relevant documents and discard non-relevant ones. Precision and recall are intuitively defined as:

- **recall** is the proportion of the relevant models retrieved in answer to a query, while the
- **precision** represents the proportion of retrieved models that are actually relevant.

In figure 5.19 is shown show the matching results obtained with the proposed graph comparison

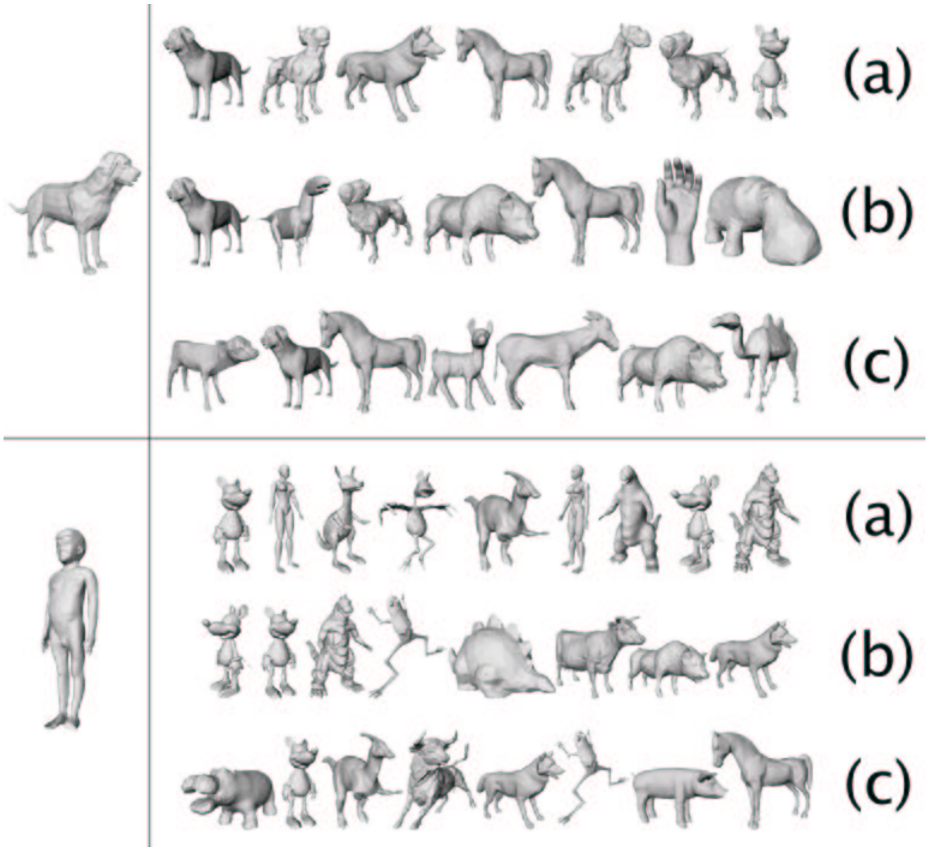


Figure 5.18: *Matching results for two query models in the shape repository with respect to the three Reeb graph representations: (a) distance from the barycentre, (b) integral geodesic distance and (c) distance from curvature extrema.*

method (chapter 4) with respect to different resolutions of the Reeb graphs used as structural descriptors. In particular, the Reeb graph has been extracted in a multi-resolution way, computing, respectively, 16, 32 and 64 subdivisions of the co-domain of the function on which is defined. The results in Figure 5.19(a) are obtained using the distance from the barycentre (DB), while Figure 5.19(b) shows the results with respect to the integral geodesic distance (IG). It is interesting to notice that the Reeb graph, with respect to the distance from the barycentre, performs better at a lower resolution while the one with respect to the integral geodesic distance improves when the number of subdivisions increases. This fact is not surprising because the distance from the barycentre induces a uniform slicing of the object that highlights more the main shape structure of the object, when the slicing is rougher. On the contrary, since contour levels of the integral geodesic distance concentrate on the object protrusions and cavities, it induces a non-uniform slicing and the resulting Reeb graph codes more handles and shape features when the number of contour levels increases. This fact emphasizes, once more, the different nature of the mapping functions: spatial based the first and shape based the second.

In Figure 5.20 the results obtained by applying the graph matching algorithm proposed in chapter 4 to the Reeb graph as structural shape descriptors, are compared with those obtained with the spherical harmonics method proposed by in Funkhouser et al. [KFR03] and discussed in section 2.2.2. Results of their method were obtained using the executables available at Kazhdan's home page[4]. When compared with the structural shape similarity method proposed in this thesis, the approach based on spherical harmonics globally performs very well and, in general, the object classes are correctly recognized. Nevertheless, has to be observed that the distinction provided by the Reeb graph structure is finer than that of the spherical harmonics and individuate the object parts that better overlap. For instance, the models in figure 5.21 do not belong to the same class of the shape repository used for these experiments but, being both elongated, they are not distinguished by the spherical harmonic descriptor while the Reeb graph correctly classifies them.

Related publications

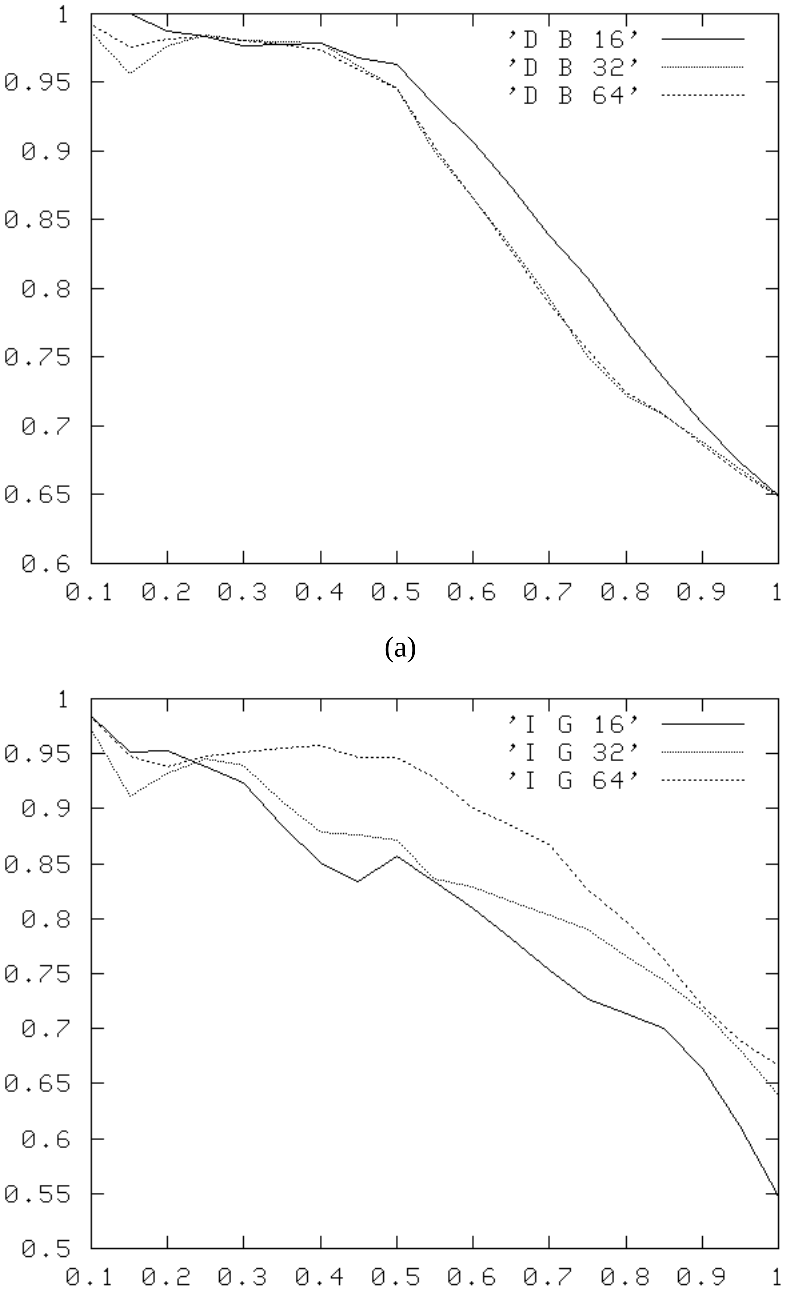
S. Marini, M. Spagnuolo, and B. Falcidieno. From exact to apporximate maximum common subgraph. In *Proceedings of the 5rd IAPR -TC-15 Workshop on Graph-based Representations in Pattern Recognition,* Lecture Notes in Computer Science, Poitiers (France), April 11-13 2005.

S. Biasotti and S. Marini. 3D object comparison based on shape descriptors. *International Journal of Computer Applications in Technology*, 23(2/3/4):57–69, 2005.

S. Biasotti, S. Marini, M. Mortara, G. Patané, M. Spagnuolo, and B. Falcidieno. 3D shape matching through topological structures. In I. Nyströn, G. Sanniti di Baja, and S. Svennson, editors, *Proceedings of the* 11^{th} *Discrete Geometry for Computer Imagery Conference*, volume 2886 of *Lecture Notes in Computer Science*, pages 194–203, Naples, 2003. Springer Verlag.

S. Biasotti, S. Marini, M. Mortara, and G. Patané. An overview on properties and efficacy of topological skeletons in shape modeling. In *Proceedings of Shape Modelling and Applications*, pages 245–254, Seoul, South Korea, June 2003. IEEE Press.

[4] http://www.cs.princeton.edu/ mkazhdan/



(a)

(b)

Figure 5.19: *Multi-resolution matching approach with respect to the distance from the barycentre (a) and the integral geodesic distance (b).*

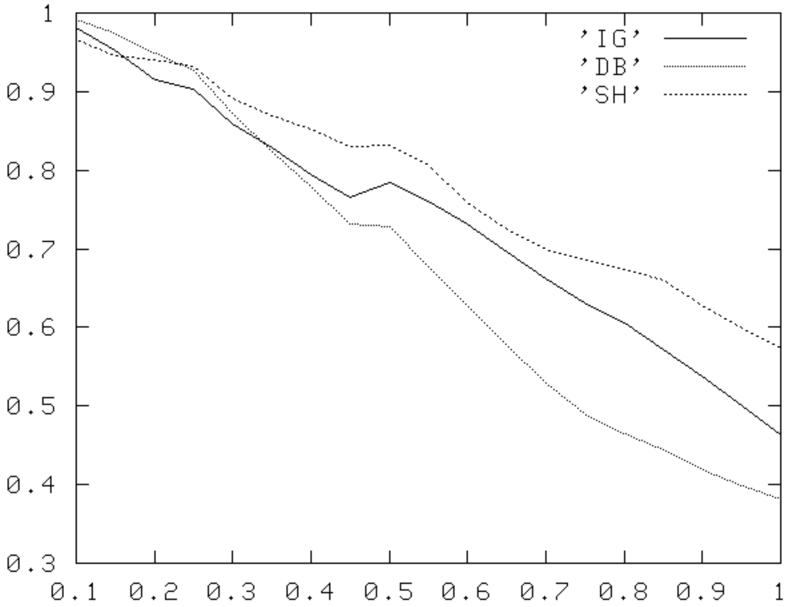


Figure 5.20: *The precision/recall curve of the graph matching with respect to the distance from the barycentre, the integral geodesic distance and the spherical harmonics method for our database, over 200 models of CAD and free form objects.*

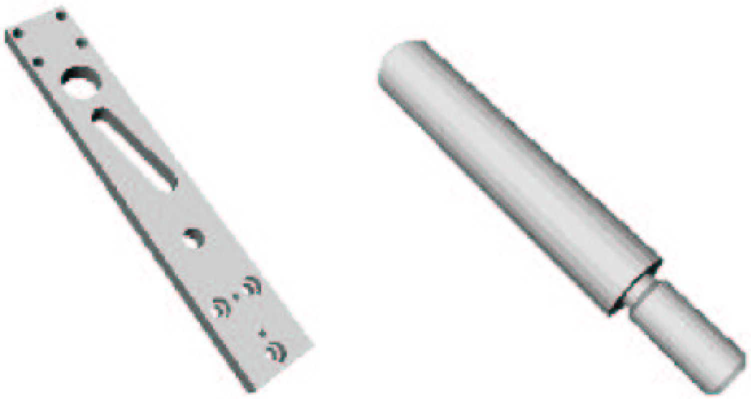


Figure 5.21: *Two CAD models.*

Conclusion and Future Work

The first issue addressed by this thesis, is the critical analysis of geometric and structural shape descriptors with respect to a set of properties aimed at highlighting conceptual and practical differences among the shape descriptors. The thesis begins discussing two methodologies for shape representation raised from a psychological analysis of the shape perception: resemblance between the object and its representation and representation of differences among objects (sections 1.1.1 and 1.1.2 respectively). Moreover, also the similarity judgments on perceptual stimuli, analyzed by psychologists, have been discussed with respect to the similarity distance properties commonly used in the computer graphics community. First, has been observed that the human behavior applied to the similarity judgment satisfy properties different from the ones satisfied by the similarity distance function commonly used for comparing 3D objects. Nevertheless semi-metric, pseudo-metric, metric and ultra-metric are easy to prove and provide a good approximation of the human perception of similarity. Anyway the most representative theories based on the violation of the distance axioms have been surveyed and discussed (section 1.2.2). Interesting aspects of the critical analysis concerning geometrical and structural shape descriptors are summarized in the following discussion, nevertheless more detailed results have been presented in the chapters 2 and 3 respectively. First of all it is important to remark that most of the 3D structural descriptors mentioned in this thesis, have to be extracted from 2-manifold models (mainly polygonal meshes), while the geometrical descriptors (discussed in the thesis) do not need such a restriction: for example, they can be obtained also from polygon soup models. This first distinction is important because makes geometrical descriptors suitable form a wide class of models, possibly retrieved from the Internet. The same models can also be used to extract structural descriptors, but through a (often not simple) preprocessing step aimed at transform them into 2-manifold models. The aspects of shape captured by the descriptors differ for the two class of geometric and structural descriptors, and also among different descriptors belonging to the same class. Has to be observed that even if geometric descriptors, excellently capture the salient feature of the objects, these features are not explicitly represented in the descriptor. This fact makes very hard the automatic reasoning about relevant features. On the other hand, structural descriptors explicitly encode the features they capture, making them suitable for shape reasoning. Nevertheless structural descriptors lack in capturing merely geometric features (e.g. sharpness, flatness, roundness, etc.) and thus they have to be endowed with attributes describing the geometric aspects of features. For example a geometrical descriptor can discriminate between chairs and other objects (e.g. tables), but it is not easy to use to deduce if a chair has the arms or not, as well as, if a chair has four leg or a pedestal. On the other hand given a structural descriptor of a chair it is possible to analyze it in order to extract the arms or the legs of the chair if any. Often the geometric shape descriptor encode the information captured from an object as a vector or a matrix. This descriptors can be very concise and suitable for applications dealing with shape retrieval of a huge number of objects. In this case the suitability cames also from the fact that, vector or matrix norm between descriptors, can be used as similarity distance function among objects. Also structural descriptors provide a concise encoding of shape objects, because their complexity reflect the structural complexity of the object. Usually few dozens of elements are enough to describe/represent an object. While structural descriptors are extracted from the object through a deterministic process, some geometric descriptors are obtained by statistical processes that make them not unique. This implies that the same object may result different from itself in application contexts dealing with similarity evaluation. The same problem occurs when structural and geometric descriptors dependent on the orientation of the coordinate system of the object. In this cases a pre-processing phase is necessary to calculate the transformations needed to move the object into a canonical position before the extraction of the descriptor. Nevertheless, for some application contexts it is reasonable to consider a special coordinate system where a direction is more relevant than the others. Two examples are the terrain analysis and the human virtual environments analysis, where the eight function is the privileged direction. In this case, among the shape descriptors not invariant with respect to rotations transformations, the Reeb graph based on the height function (described in section 3.2.3.1) is the most suitable descriptor, because it is independent by the rotations along the axis parallel to the height direction. Scalability controls the level of detail of the features described/represented by the shape descriptor. Both geometrical and structural descriptors allow multi-resolution description/representation of the object, but only structural descriptors provide multi-scale capabilities. This is because, as already mentioned, they explicitly encode the salient features of the object, thus the size of the descriptor can be reduced or organized by discarding not relevant features or selecting the most relevant ones. Finally, the meaning of the information encoded, makes the descriptors suitable for different application contexts dealing with similarity evaluation. Saliency and conciseness provide a good base for shape retrieval: saliency affects the quality of the obtained results and conciseness improve the computational performance when the number of models involved in the retrieval task is large. Also scalability affects the quality of the results of a retrieval task, especially when an intelligent search among the models necessitate of a multi-scale-based reasoning. Application tasks dealing with recognition, classification, composition and editing necessitate to recognize sub-part correspondence among the shape models. For these reason, structural descriptors are the best suitable tools, because comparison between models is obtained through a one-to-one mapping between the elements of the descriptors (usually graph nodes and edges). Structural descriptors are also the best choice for animation tasks: this is because, due to their structural nature, they allow the selection of local parts of the object that can be deformed during the animation process, without affecting the rest of the object.

An algorithm for the computation of the maximum common subgraph between two directed acyclic graphs and its correctness proof have been presented. The algorithm is tailored to encapsulate heuristic techniques that may yield to an approximation of the maximum common subgraph only, but that make the algorithm suitable for applications where graph structural noise may compromise the results and computational costs have to be optimized, both in terms of time consumed and occupied memory space. Different heuristic techniques have been described and their effectiveness has been shown and discussed with respect to shape structural descriptors of 3D objects. The algorithm proposed in this thesis has been applied for measuring similarity and recognizing sub-part correspondence between 3D shapes. Main research contribution is the new similarity matching mechanism to compare 3D shapes coupling geometry and topology. Since this method computes an approximation of the maximal common subgraph of two structural shape descriptors, it is particularly suitable for sub-part shape correspondence. In addition, it is flexible, because it can be applied to any skeletal structure with the same properties of the structural descriptor (attributed, directed and acyclic), and tunable, as it can be used in a multi-step query approach, to progressively refine the set of geometrically similar candidates. No existing shape descriptor satisfies all the ‘ideal’ requirements for shape retrieval. In fact, it has been shown that curve skeletons may be topologically non equivalent to the original shape and, both curve skeletons and Reeb graphs, may depend on shape details. On the contrary spherical harmonics are more stable for matching, but there is no correspondence between the descriptor and the shape of the object subparts. Furthermore, it has been shown that matching methods based on skeletal-based descriptors are better suitable for tasks for which it is fundamental to decompose the shape in salient portions, while other approaches, such as those based on shape distributions and spherical harmonics, better performs in retrieval tasks if partial matching and reasoning about subparts differences is not needed.

Future Works

This work can be extended and improved in several directions. By the point of view of the structural shape descriptors, the choice of the mapping function in the Reeb graph representation determines the characteristics of the resulting shape descriptor and, usually, each function highlights one shape property at time. It is interesting to investigate how to simultaneously use and integrate different mapping functions. This approach is promising and goes into the direction of developing tools to automatically annotate the shape semantic. In fact, we are trying to understand where and how the semantics can be encapsulated in the digital representation of shapes to move toward semantically capable digital representations of shapes, able to support the emerging categories of applications dealing with digital shapes. In this sense, it is interesting to formalize the application domain through a specific ontology which provides the rules for associating semantics to shape or shape parts (e.g., in Industrial Manufacturing, a slot is defined by two parallel faces which are both adjacent to a horizontal face) [IST].

The comparison algorithm developed may be applied to application contexts that are more general than those presented in chapter 5. Therefore, it would be interesting to test the approach proposed for graph comparison to other application fields such as virtual human analysis, bio-medicine and robotics. Furthermore, the graph isomorphism induced by the proposed matching approach can be exploited to deduce rules for shape editing, in order to topologically and structurally align two different shapes. Also the definition of new distance measures that are independent of the matching approach and their possible combination with other distances will be investigated. With reference to the application context of the shape retrieval, Will be consider a multi-step approach where a set of different filters, for example coarse filters, shape harmonics and structural descriptors, are used to progressively refine the set of geometrically similar candidates. In this way we will obtain a multi-modal query mechanism that could provide a combination of various measures of shape similarity, corresponding to function, form and structure analysis of $3D$ shapes.

From the number of application previously listed, it should be clear that structural shape comparison is a rich, interesting, and rapidly evolving discipline. The work in this thesis suggests that further research in this field is likely to be fruitful and it is reasonable that it will increase during next years, as digital shapes will become more and more popular.

Bibliography

[Abd98] A. M. Abdulkader. *Parallel Algorithms for Labelled Graph Matching*. PhD thesis, Colorado School of Mines, 1998.

[ABS03] M. Attene, S. Biasotti, and M. Spagnuolo. Shape understanding by contour driven retiling. *The Visual Computer*, 19(2-3):127–138, 2003.

[AC02] A. Angelidis and M.-P. Cani. Adaptive implicit modeling using subdivision curves and surfaces as skeletons. In *Proceedings of the* 7^{th} *ACM Symposium on Solid Modeling and Applications*, pages 45–52, Saarbrucken, Germany, 2002. ACM Press.

[AE98] U. Axen and H. Edelsbrunner. Auditory Morse Analysis of Triangulated Manifolds. In H.-C. Hege and K. Polthier, editors, *Mathematical Visualization*, pages 223–236. Springer-Verlag, Heidelberg, 1998.

[AP88] F. G. Ashby and N. A. Perrin. Toward a unified theory of similarity and recognition. *Psychological review*, 95(1):124–150, 1988.

[AT86] F. G. Ashby and J. T. Townsend. Varieties of perceptual independence. *Psychological review*, 93:154–179, 1986.

[BBB$^+$97] J. Bloomenthal, C. Bajaj, J. Blinn, M.-P. Cani-Gascuel, A. Rockwood, B. Wyvill, and G. Wyvill, editors. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., San Francisco, California, 1997.

[BFM$^+$96] Julio E. Barros, James C. French, Worthy N. Martin, Patrick M. Kelly, and T. Michael Cannon. Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 392–403, 1996.

[BFS00] S. Biasotti, B. Falcidieno, and M Spagnuolo. Extended Reeb Graphs for surface understanding and description. In G. Borgefors and G. Sanniti di Baja, editors, *Proceedings of the* 9^{th} *Discrete Geometry for Computer Imagery Conference*, volume 1953 of *Lecture Notes in Computer Science*, pages 185–197, Uppsala, 2000. Springer Verlag.

[BFS02] S. Biasotti, B. Falcidieno, and M. Spagnuolo. Shape abstraction using computational topology techniques. In U. Cugini and M. Wozny, editors, *From Geometric Modeling to Shape Modeling*, pages 209–222. Kluwer Academic Publishers, 2002.

[Bia04] S. Biasotti. *Computational topology Topology Methods for Shape Modelling Applications*. PhD thesis, University of Genova, May 2004.

[Bie87] I. Biederman. Recognition-by-Components: A theory of human image understanding. *Phicological Review*, 94:115–147, 1987.

[Bie95] I. Biederman. Visual object recognition. In S. Kosslyn and D. Osherson, editors, *An invitation to Cognitive Science*, volume 2, chapter 4, pages 121–165. MIT Press, 1995.

[BL99] J. Bloomental and C. Lim. Skeletal methods of shape manipulation. In *Proceedings of Shape Modelling and Applications '99*, pages 44–49, Aizu-Wakamatsu, Japan, March 1999. IEEE Press.

[Blu67] H. Blum. A transformation for extracting new descriptors of shape. In W. Whaten-Dunn, editor, *Proceedings of the Symp. Models for Perception of Speech and Visual form*, pages 362–380. Cambridge MA: MIT Press, 1967.

[BM05] S. Biasotti and S. Marini. 3D object comparison based on shape descriptors. *International Journal of Computer Applications in Technology*, 23(2/3/4):57–69, 2005.

[BMM$^+$03] S. Biasotti, S. Marini, M. Mortara, G. Patané, M. Spagnuolo, and B. Falcidieno. 3D shape matching through topological structures. In I. Nyströn, G. Sanniti di Baja, and S. Svennson, editors, *Proceedings of the* 11^{th} *Discrete Geometry for Computer Imagery Conference*, volume 2886 of *Lecture Notes in Computer Science*, pages 194–203, Naples, 2003. Springer Verlag.

[BMMP03] S. Biasotti, S. Marini, M. Mortara, and G. Patané. An overview on properties and efficacy of topological skeletons in shape modelling. In *Proceedings of Shape Modelling and Applications*, pages 245–254, Seoul, South Korea, May 2003. IEEE Press.

[BMS00] S. Biasotti, M. Mortara, and M. Spagnuolo. Surface compression and reconstruction using Reeb Graphs and Shape Analysis. In *Proceedings of the* 16^{th} *Spring Conference on Computer Graphics*, pages 174–185, Budmerice, Bratislava, May 2000.

[BRS03] D. Bespalov, W. C. Regli, and A. Shokoufandeh. Reeb Graph Based Shape Retrieval for CAD. In *Proceedings of the 2003 ASME Design Engineering Technical Conferences*, Chicago, Illinois, September 2003. ASME.

[BS91] J. Bloomenthal and K. Shoemake. Convolution surfaces. In *ACM Computer Graphics, (Proc. of SIGGRAPH '91)*, volume 25, 4, pages 251–256. ACM Press, 1991.

[BS98] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19:255–259, 1998.

[BSRS03] D. Bespalov, A. Shokoufandeh, W. C. Regli, and W. Sun. Scale-space representation of 3D models and topological matching. In *Proceedings of the* 8^{th} *ACM Symposium on Solid Modeling and Applications*, pages 208–215, Seattle, Washington, June 2003. ACM Press.

[BSRS04] Dmitriy Bespalov, Ali Shokoufandeh, William C. Regli, and Wei Sun. Local feature extraction using scale-space decomposition. In *ASME Design Engineering Technical*

Conferences, Computers and Information in Engineering Conference Conference (DETC 2004-57702). ASME Pres, Sep 2004.

[BW00] D. Brodsky and B. Watson. Model simplification through refinement. In *Proceedings of Graphics Interface,*, pages 221–228, 2000.

[Cal85] C.R. Calladine. Gaussian curvature and shell structures. *The Mathematics of Surfaces*, pages 179–196, 1985.

[CC70] J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of ”eckart-young” decomposition. *Psychometrika*, 35:283–319, 1970.

[CC72] J. D. Carroll and J. J. Chang. Idoscal (individual differences in orientation scaling): A generalization of lndscal allowing idiosyncratic reference systems as well as analytic approximation to indscal. In *Psychometric Society*, Princeton, NJ., 1972.

[CFSV04] Luigi P. Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1367–1372, October 2004.

[CG99] Scott Cohen and Leonidas Guibas. The earth mover’s distance under transformation sets. In *ICCV ’99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 1076. IEEE Computer Society, 1999.

[CGC03] D. Conte, C. Guidobaldi, and C.Sansone. A comparison of three maximum common subgraph algorithms on a large database of labeled graphs. In E. Hancock and M. Vento, editors, *Proc. of IAPR Workshop GbRPR 2003*, volume 2726 of *Lecture Notes in Computer Science*, pages 130–141. Springer-Verlag Berlin Heidelberg, 2003.

[Cho02] Y. Choe. Second order isomorphism: A reinterpretation and its implications in brain and cognitive sciences. In W. D. Gray and D. Shunn, editors, *Proceedings of the 24th Annual Conference of the Cognitive Science Society.*, pages 190–195, 2002.

[CK01] C. M. Cyr and B. B. Kimia. 3D object recognition using shape similarity-based Aspect Graph. In *Proceedings of the International Conference on Computer Vision*, pages 254–261. IEEE Press, 2001.

[COT$^+$03] D.Y. Chen, M. Ouhyoung, X.P. Tian, Y.T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. In *Proceedings of the 23^{th} European Conference on Computer Graphics (Eurographics ’03)*, volume 22, 3 of *Computer Graphics Forum*, pages 223–232. BLACKWELL PUBL LTD, 108 COWLEY RD, OXFORD OX4 1JF, OXON, ENGLAND, 2003.

[CRC$^+$02] J. Corney, H. Rea, D. Clark, J. Pritchard, M. Breaks, and R. MacLeod. Coarse filters for shape matching. *IEEE Computer Graphics and Applications*, 22(3):65–74, 2002.

[CSA03] H. Carr, J. Snoeyink, and U. Axen. Computing Contour Trees in all dimensions. *Computational Gemetry*, 24:75–94, 2003.

[CW74] J. D. Carroll and M. Wish. Models and methods for three-way multidimensional scaling. In D. H. Krantz, R. C. Atkinson, R. D. Lute, and P. Suppes, editors, *Contemporary developments in mathematical psychology*, volume 2, pages 57–105, New York, 1974.

[Dav83] M. L. Davison, editor. *Multidimensional scaling*. Wiley, New York, 1983.

[DSD$^+$04] M. Fatih Demirci, Ali Shokoufandeh, Sven Dickinson, Yakov Keselman, and Lars Bretzner. Many-to-many feature matching using spherical codng of directed graphs. In *Proceedings of the European Conference on Computer Vision.*, Prague, May 2004.

[EDB97] S. Edelman and S. Duvdevani-Bar. Similarity, connectionism, and the problem of representation in vision. *Neural Computation*, 9(4):701–721, 1997.

[Ede96] S. Edelman. Representation is representation of similarities. Technical Report CS96-08, Center for Biological and Computational Learning, Dept. of Brain and Cognitives Sciences, 1996.

[FGN89] T. Fan, G.Medioni, and R. Nevatia. Recognizing 3D objects using surface description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1140–1157, November 1989.

[FK97] A. Fomenko and T. L. Kunii. *Topological Modelling for Visualization*. Springer Verlag, 1997.

[FKS$^+$04] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin. Modeling by example. *ACM Transactions on Computer Graphics, (Proc. of SIGGRAPH 2004)*, 2004.

[FMK$^+$03] Thomas Funkhouser, Patrick Min, Misha Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3d models. *ACM Transactions on Graphics*, 22(1):83–101, 2003.

[FS98a] R. Fagin and L. Stockmeyer. Relaxing the triangle inequality in pattern matching. *International Journal of Computer Vision*, 28(3):219–231, 1998.

[FS98b] B. Falcidieno and M. Spagnuolo. A shape abstraction paradigm for modeling geometry and semantics. In F.-E. Wolter and N. M. Patrikalakis, editors, *Proceedings of the Conference on Computer Graphics International 1998 (CGI-98)*, pages 646–657, Los Alamitos, California, June 22–26 1998. IEEE Computer Society.

[Ger39] A. Gershun. The light field. *Translated by P. Moon and G. Timoshenko in Journal of Mathematics and Physics*, 18:51–151, 1939.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 1979.

[GK00] P. Giblin and B. B. Kimia. A formal classification of 3D Medial Axis points and their local geometry. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition*, volume 1, pages 566–573, 2000.

[GK03] P. J. Giblin and B. B. Kimia. On the local form and transitions of Symmetry Sets, Medial Axes, and Shocks. *International Journal of Computer Vision*, 54(1-3):143–157, 2003.

[GP74] V. Guillemin and A. Pollack. *Differential Topology*. Englewood Cliffs, NJ: Prentice-Hall, 1974.

[GS99] N. Gagvani and D. Silver. Parameter controlled volume thinning. *Graphical Models and Image Processing*, 61(3):149–164, 1999.

[HA03] F. Hétroy and D. Attali. Topological quadrangulations of closed triangulated surfaces using the Reeb Graph. *Graphical Models*, 65(1):131–148, 2003.

[Hag00] M. Hagedoorn. *Pattern matching using similarity measures.* PhD thesis, University of Utrecht (The Netherlands), 2000.

[Hat01] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2001.

[HKDH04] D. Huber, A. Kapuria, R. Donamukkala, and Martial Hebert. Parts-based 3D object classification. In *Proceedings of the International Conference on Computer Vision*, 2004.

[Hor69] C. B. Horan. Multidimensional scaling: Combining observa- tions when individuals have different perceptual structures. *Psychometrika*, 34:139–165, 1969.

[HR85] D. D. Hoffman and W. Richards. Parts of recognition. *Cognition*, pages 65–96, 1985.

[HSKK01] M. Hilaga, Y. Shinagawa, T. Komura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *ACM Computer Graphics, (Proc. of SIGGRAPH 2001)*, pages 203–212, Los Angeles, 2001. ACM Press.

[htt] http://www.google.com.

[HV99] M. Hagedoorn and R. Veltkamp. Metric pattern spaces. Technical Report UU-CS-1999-03, University of Utrecht (The Netherlands), 1999.

[IST] EC IST. FP6 Network of Excellence: AIM@SHAPE. http://www.aimatshape.net. Starting date january 2004 - Duration 48 months.

[Jam90] W. James. *Principles of psychology*. Holt, New York, 1890.

[JH97] A. E. Johnson and M. Hebert. Recognizing objects by matching oriented points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 684–689, 1997.

[JH99] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, 1999.

[Joh97] Andrew Edie Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University., 1997.

[Jon99] R. Jones. Connected Filtering and Segmentation Using Component Trees. *Computer Vision and Image Understanding: CVIU*, 75(3):215–228, September 1999.

[KFR03] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In L. Kobbelt, P. Schröder, and H. Hoppe, editors, *Proceedings of Symposium in Geometry Processing*, pages 156–165, Aachen, Germany, June 2003.

[Kru64a] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.

[Kru64b] J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:115–129, 1964.

[Kru78] C. L. Krumhansl. Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density. *Psychological Review*, 85:445–463, 1978.

[KT75] D. H. Krantz and A. Tversky. Similarity of rectangles: An analysis of subjective similarity. *Journal of Mathematical Psychology*, 12:4–34, 1975.

[KTZ95] B. Kimia, A. Tannenbaum, and S. Zucker. Shapes, shocks, and deformations, I: The components of shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15:189–224, 1995.

[KvD76] J.J. Koenderink and A.J. van Doorn. The singularities of the visual mapping. *Biological Cybernetics*, 24(1):51–59, 1976.

[KvD79] J.J. Koenderink and A.J. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32(4):211–216, 1979.

[LH96] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of ACM SIGGRAPH*, pages 31–42,, New Orleans, USA, 1996.

[LV99] F. Lazarus and A. Verroust. Level set diagrams of polyhedral objects. In *Proceedings of the 5^{th} ACM Symposium on Solid Modeling and Applications*, pages 130–140, Ann Arbor, 1999. ACM Press.

[Mal94] T. Maldonato. *Reale e Virtuale*. Feltrinelli, MIlano, Italy, 1994.

[Mar82] D. Marr. *Vision - A computational investigation into the human representation and processing of visual information*. W. H. Freeman, San Francisco, 1982.

[Mas67] W. Massey. *Algebraic Topology: An Introduction*. Brace & World, Inc, 1967.

[MB98] B. T. Messmer and H. Bunke. A new algorithm for error tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):493–504, May 1998.

[Mes97] B. T. Messmer. *Graph Matching Algorithms and Applications*. PhD thesis, University of Bern, Bern, Switzerland, 1997.

[Mil63] J. Milnor. *Morse Theory*. Princeton University Press, New Jersey, 1963.

[MP02] M. Mortara and G. Patané. Shape-covering for skeleton extraction. *International Journal of Shape Modelling*, 8(2):245–252, 2002.

[MPS$^+$03] M. Mortara, G. Patané, M. Spagnuolo, B. Falcidieno, and J. Rossignac. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica*, 38(2):227–248, 2003.

[MSF05] S. Marini, M. Spagnuolo, and B. Falcidieno. From exact to apporximate maximum common subgraph. In *Proceedings of the 5rd IAPR -TC-15 Workshop on Graph-based Representations in Pattern Recognition*, Lecture Notes in Computer Science, Poitiers (France), April 11-13 2005.

[Nac84] L. R. Nackman. Two-dimensional Critical Point Configuration Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(4):442–450, 1984.

[NK01] M. Novotni and R. Klein. A geometric approach to 3D object comparison. In *Proceedings of Shape Modelling and Applications*, pages 167–175, Genova, Italy, May 2001. IEEE Press.

[oCE87] Longaman Dictionary of Contemporary English. Longman Edition, 1987.

[OFCD01] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In *Proceedings of Shape Modelling and Applications*, pages 154–166, Genova, Italy, May 2001. IEEE Press.

[OFCD02] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics (TOG)*, 21(4):807–832, October 2002.

[Pen86] Alex P. Pentland. Perceptual organization and the representation of natural form. *Artif. Intell.*, 28(3):293–331, 1986.

[PSZ99] M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching hierarchial structures using association graphs. *IEEE Transanctions on Pattern Analysis and Machine Intelligence*, 21(11):1105–1119, 1999.

[RC77] R. C. Read and D. G. Corneil. The graph isomorphism disease. *Journal of Graph Theory*, 1:339–363, 1977.

[RCSM03] S. Ruiz-Correa, L. Shapiro, and M. Meila. A new paradigm for recognizing 3D object shapes from range data. In *Proceedings of the International Conference on Computer Vision*, pages 1126–1133, October 2003.

[Ree46] G. Reeb. Sur les points singulièrs d'une forme de Pfaff complètement intégrable ou d'une fonction numèrique. *Comptes Rendu de l'Academie des Sciences*, 222:847–849, 1946.

[Rij79] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth, 1979.

[Sac86] Oliver Sacks. *The Man Who Mistook His Wife For a Hat*. Pan Macmillan Ltd, Pan Macmillan, 20 New Harf Road, London N1 9RR, 1986.

[SC70] R. N. Shepard and S. Chipman. Second-order isomorphism of internal representations: Shape of states. *Cognitive Psychology*, 1:1–17, 1970.

[SD01] A. Shokoufandeh and S. Dickinson. A unified framework for indexing and matching hierarchical shape structures. In *Proceedings of 4th International Workshop on Visual Form*, Capri, Italy, May 28–30 2001.

[She62a] R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function I. *Psychometrika*, 27:125–140, 1962.

[She62b] R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function II. *Psychometrika*, 27:219–246, 1962.

[SJ99] Santini S and R. Jain. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):871–883, September 1999.

[SK91] Y. Shinagawa and T. L. Kunii. Constructing a Reeb Graph automatically from cross sections. *IEEE Computer Graphics and Applications*, 11(6):44–51, November 1991.

[Ski45] B. F. Skinner. The operational analysis of psychological terms. *Psychological Review*, 52:270–277, 1945.

[SKK91] Y. Shinagawa, T. L. Kunii, and Y. L. Kergosien. Surface coding based on Morse Theory. *IEEE Computer Graphics and Applications*, 11(5):66–78, September 1991.

[SL01] D. Shattuck and R. Leahy. Automated graph based analysis and correction of cortical volume topology. *IEEE Transactions on Medical Imaging*, 20(11):1167–1177, 2001.

[SMF94] P. Suppes, M.Pavel, and J. Falmagne. Reppresentations and models in psychology. *Ann. Rev. Psychol.*, 45:517–544, 1994.

[SOG98] P. Salembier, A. Oliveras, and L. Garrido. Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4):555–570, April 1998.

[Spa66] E. Spanier. *Algebraic Topology*. McGraw Hill, 1966.

[SPB96] E. C. Sherbrooke, N. M. Patrikalakis, and E. Brisson. An algorithm for the Medial Axis Transform of 3D polyhedral solids. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):44–61, 1996.

[SSC03] V. Singh, D. Silver, and N. Cornea. Real-time volume manipulation. In *Proceedings of 3rd International Workshop on volume*, 2003.

[SSDZ98] K. Siddiqi, A. Shokoufandeh, S.J. Dickenson, and S.W. Zucker. Shock graphs and shape matching. In *Proceedings of the* 5^{th} *International Conference on Computer Vision*, pages 222–229, 1998.

[SSGD03] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Proceedings of Shape Modelling and Applications*, pages 130–139, Seoul, South Korea, June 2003. IEEE Press.

[STK94] K. Sididqi, K. Tresness, and B. B. Kimia. Parts of visual forms: Echological and psychophysical aspects. Technical Report TR LEMS-104, Laboratory for Engineering, Brown University, 1994.

[TG78] A. Tversky and I. Gati. Studies of similarity. In E. Rosch and B. Lloyd, editors, *Cognition and Categorization*, chapter 4, pages 79–98. Erlbaum, Hillsdale, NJ, 1978.

[TG82] A. Tversky and I. Gati. Similarity, separability, and the triangle inequality. *Psychological Review*, 89:123–154, 1982.

[TH84] A. Tversky and K. Hemenway. Objects, parts and categories. *Journal of Experimental Psychology*, 113:169–193, 1984.

[TIS+95] S. Takahashi, T. Ikeda, Y. Shinagawa, T. L. Kunii, and M. Ueda. Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data. *Computer Graphics Forum*, 14(3):181–192, September 1995.

[TO02] G. Turk and J. O'Brien. Modelling with implicit surfaces that interpolate. *ACM Transactions on Graphics*, 21(4):855–873, 2002.

[Tor58] W. S. Torgerson. *Theory and methods of scaling*. Wiley, New York, 1958.

[TS04] T. Tung and F. Schmitt. Augmented reeb graphs for content-based retrieval of 3d mesh models. In *Proceedings of Shape Modelling and Applications*, Genova, Italy, June 2004. IEEE Press.

[Tuc72] L. R. Tucker. Relations between multidimensional scaling and three-mode factor analysis. *Psychometrika*, 37:3–28, 1972.

[Tve77] A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.

[Ull76] E. K. Ullman. An algorithm for subgraph isomorphism. *Journal of ACM*, 23(1):31–42, 1976.

[Vel01] R. C. Veltkamp. Shape matching: Similarity measures and algorithms. In *Proceedings of Shape Modelling and Applications*, pages 188–197, Genova, Italy, May 2001. IEEE Press.

[VH00] R. Veltkamp and M. Hagedoorn. Shape similarity measures, properties, and constructions. Technical Report UU-CS-2000-37, University of Utrecht (The Netherlands), 2000.

[VL00] A. Verroust and F. Lazarus. Extracting skeletal curves from 3D scattered data. *The Visual Computer*, 16(1):15–25, 2000.

[Vra04] D. V. Vranič. *3D Model Retrieval*. PhD thesis, University of Leipzig, Leipzig, Germany, 2004.

[VSR01] D. V. Vranič, D. Saupe, and J. Richter. Tools for 3D-Object retrieval. In *Proceedings of the IEEE 2001 Workshop on Multimedia Signal Processing*, pages 293–298, Cannes, France, October 2001. IEEE Press.

[VT03] R. C. Veltkamp and J. W. Tangelder. Polyhedral model retrieval using weighted point sets. In *Proceedings of Shape Modelling and Applications*, pages 119–128, Seoul, South Korea, 2003. IEEE Press.

[VV99] Jules Vleugels and Remco C. Veltkamp. Efficient image retrieval through vantage objects. In *Visual Information and Information Systems*, pages 575–584, 1999.

[VV02] J. Vleugels and R. Veltkamp. Efficient image retrieval through vantage objects. *Pattern Recognition*, 35(1):69–80, 2002.

[WDSB00] Z. J. Wood, M. Desbrun, P. Schröder, and D. Breen. Semi-regular mesh extraction from volumes. In *Proc. of the* 11^{th} *Annual IEEE Visualization Conference (VIS) 2000*, pages 275–282, 2000.

[WSS^{+}98] J. T. L. Wang, B. A. Shapiro, D. Shasha, K. Zhang, and K. M. Currey. An algorithm for finding the largest approximately common substructures of two trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20:889–895, 1998.

[ZL02] D. S. Zhang and G. Lu. A comparative study of fourier descriptors for shape representation and retrieval. In *Proceedings of the 5th Asian Conference on Computer Vision (ACCV)*, pages 652–657, January 2002.

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

Appendix A

Basic Definitions on Graphs

Definition A.0.1 (Attributed graph) *An attributed graph $\mathcal{G}$ is given by a quadruple $G = (V, E, \mu_V, \mu_E)$, where V is a set of nodes, E is the set of the graph edges, $\mu_V : V \to A_V$ and $\mu_E : E \to A_E$ are the node and the edge attribute functions, with A_V, A_E sets of node and edge attributes of G. The set of attributed graphs is denoted by M_{Gset}.*

Definition A.0.2 (Attributed sub-graph) *A subgraph S of G is a quadruple $(V_S, E_S, \mu_{V_S}, \mu_{E_S})$, where $V_S \subseteq V$, $E_S \subseteq E$, μ_{V_S} and μ_{E_S} are induced by μ_V and μ_E, respectively.*

Definition A.0.3 (path) *a path between two vertices $n_1, n_2 \in V$ is a non-empty sequence of k different vertices $v_0, v_1, \ldots, v_k$ where $v_0 = n_1$ and $v_k = n_2$ and $(v_i, v_{i+1}) \in E$, $i = 0, \ldots, k-1$. Finally, a graph G is said to be* acyclic *when there are no cycles between its edges, independently of whether the graph G is directed or not.*

Definition A.0.4 (graph isomorphism) *is a bijective function $f : \mathcal{V}_1 \to \mathcal{V}_2$ such that*

1. $\mu^{\mathcal{V}_1}(v) = \mu^{\mathcal{V}_2}(f(v)),\ v \in \mathcal{V}_1$.
2. *for all the edges $e_1 = (v_1, v_1') \in \mathcal{E}_1$, there exists an edge $e_2 = (f(v_1), f(v_1')) \in \mathcal{E}_2$ such that $\mu^{\mathcal{E}1}(e_1) = \mu^{\mathcal{E}_2}(e_2)$. Moreover, for all the edges $e_2 = (v_2, v_2') \in \mathcal{E}_2$, there exists an edge $e_1 = (f^{-1}(v_2), f^{-1}(v_2')) \in \mathcal{E}_1$ such that $\mu^{\mathcal{E}_1}(e_1) = \mu^{\mathcal{E}_2}(e_2)$.*

If a graph is not attributed, the condition 1 and the equality between the edge attributes in the condition 2, are not necessary.

Definition A.0.5 (subgraph isomorphism) *If $f : \mathcal{V}_1 \to \mathcal{V}_1'$ is a graph isomorphism between $\mathcal{G}_1$ and $\mathcal{G}'$, and $\mathcal{G}'$ is a subgraph of $\mathcal{G}_2$, then f is called a subgraph isomorphism from $\mathcal{G}_1$ to $\mathcal{G}'$.*

Definition A.0.6 (common subgraph) *A common subgraph of $\mathcal{G}_1$ and $\mathcal{G}_2$ is a graph $\mathcal{G}$ such that there exists a subgraph isomorphism from $\mathcal{G}$ to $\mathcal{G}_1$ and from $\mathcal{G}$ to $\mathcal{G}_2$.*

Definition A.0.7 (maximum common subgraph) *A maximum common subgraph of $\mathcal{G}_1$ and $\mathcal{G}_2$, denoted as $\mathcal{M}CS_{\mathcal{G}_1,\mathcal{G}_2}$ is a common subgraph $\mathcal{G}$ such that there exists no other common subgraph having more nodes than $\mathcal{G}$. The $\mathcal{M}CS_{\mathcal{G}_1,\mathcal{G}_2}$ in not necessarily unique.*

Definition A.0.8 (error tolerant graph isomorphism) *Let $\mathcal{G}$ and $\mathcal{G}'$ be two attributed graphs as proposed in definition A.0.1 and $\Delta = (\delta_1, \ldots, \delta_n)$ a sequence of graph editing operations, where a graph edit operation, δ_i, is an addition, a deletion or an attribute modification of nodes and edges, then:*

- *the* edited graph $\Delta(\mathcal{G})$ *is the graph* $\Delta(\mathcal{G}) = \delta_n(\delta_{n-1}(\ldots (\delta_1(\mathcal{G}))\ldots))$*;*
- *an* error tolerant graph isomorphism *is a couple (Δ, ψ), where Δ is a sequence of editing operations such that there exists an graph isomorphism ψ between $\Delta(\mathcal{G})$ and $\mathcal{G}'$.*