

Size functions for comparing 3D models

S. Biasotti, D. Giorgi, M. Spagnuolo and B. Falcidieno

IMATI - CNR, Via De Marini 6, Genova, Italy

Abstract

This paper proposes an original framework to use size functions in the 3D context. Size functions are a mathematical tool, that have already shown its effectiveness for image retrieval and classification. They are here introduced for the first time to discriminate among 3D objects represented by triangle meshes, through the proposal of a method for defining size graphs independently of the underlying triangulation. We first derive a skeletal signature, which guarantees the topological coding and the geometric description of an object surface, then this signature is used as a size graph to compute discrete size functions. The attractive feature of size functions is that readily give a similarity measure between shapes. The result is the introduction of a new technique for 3D model retrieval, devised to capture both local and global properties of a shape. Finally, we demonstrate the potential of our approach in a set of experiments, and discuss the results with respect to existing techniques.

Key words: Skeletal graph, size graph, size function, matching distance, shape comparison, 3D shape retrieval

1 Introduction

Shape comparison plays a fundamental role in the Computer Vision and Computer Graphics fields. In the last decade, the advances in modelling, digitizing and visualizing 3D shapes have led to an explosion in the number of available 3D models, both on the Internet and in domain-specific databases. Examples are digital repositories recording cultural heritage [1] or archives of structural data of biological macromolecules [2]. Whilst it has become relatively easy to generate 3D information and interact with the geometry of shapes, it is

Email address: {`silvia,daniela,michi,bianca`}@ge.imati.cnr.it (S. Biasotti, D. Giorgi, M. Spagnuolo and B. Falcidieno).

URL: <http://www.ge.imati.cnr.it> (S. Biasotti, D. Giorgi, M. Spagnuolo and B. Falcidieno).

harder to structure, filter, organize and retrieve it. How to find and interpret 3D content has become the key issue, leading to the development of the first experimental search engines for 3D shapes, such as the 3D model search engine at Princeton University [3] or the 3D retrieval engine at Utrecht University [4].

From a high-level perspective, the main components of a retrieval system for 3D or 2D visual media are similar: a feature extraction module and an indexing system, that usually works off-line, and a matching module, that extracts online the most relevant items out of a collection, according to some metrics defined on the feature space. Human perception has been widely studied and supported a significant amount of work in Computer Vision related to the analysis and recognition of images [5,6]. The main differences between 2D pixel-based and 3D vector-based contexts arise in the feature extraction step and, unfortunately, most of the methods developed for images do not generalize directly to 3D shapes. This is mainly due to the different nature of the content: descriptors used for 2D images are concerned with color, textures, and properties that capture geometric details of the shapes segmented in the image. Most notably, feature extraction for image retrieval is intrinsically affected by the possible presence of information, which is only accidental in the image or due to occlusion and/or perspective distortion. On the other hand, the boundary of 3D models is represented in vector form and therefore does not need to be segmented from a background. Possessing the complete geometry of 3D models allows for more effective and reliable search tools, although the intrinsic complexity of 3D shapes still makes the understanding of their content an arduous problem.

In this context, we are working for the development of a 3D search platform, that provides a rich and flexible set of descriptors within the same framework. Shape properties and their relevance to the search are obviously dependent on the user and context, meaning that we are able to index an object with different features according to the context in which we are asked to describe it. We aim to provide a modular system for shape comparison and retrieval, which allows users to fit a descriptor to the shape idea they have in mind. *Size theory* offers the theoretical support for the development of such type of framework. In particular, in this paper we introduce a new technique for 3D shape retrieval, which builds on *size functions*, a mathematical descriptor, which can readily and efficiently be used to establish a similarity measure between shapes.

The theory of size functions has been developed since the beginning of the 1990s in order to get a new geometric-topological approach to shape discrimination [7,8]. The idea is to analyze a given shape by exploring the growth of a topological space S associated to the shape, according to the increasing values of a real function φ defined on it. Intuitively, size functions code

the topological evolution of S counting the number of connected components which remain disconnected passing from a lower level set of S to another. Since the growth of S is driven by the real function φ , size functions encode the geometrical properties of S captured by φ in the topological evolution of S . An introductory presentation can be found in [9,10]. In more recent years, a similar approach has been applied in a homological setting, independently leading to the introduction of persistent homology [11] and the definition of the Morse homology descriptor [12].

The most interesting aspect of size theory is that we can obtain a rich set of descriptors by simply changing the real function φ , while keeping the same matching framework. Another interesting aspect is the possibility to apply the theory not only to the shape itself, but also to an auxiliary space, associated to the shape, and enhancing its relevant characteristics [13]. This idea is exploited in the present paper.

Size functions have been extensively applied to content-based image retrieval and classification (see for example [8,14,15]), proving their effectiveness for describing and comparing objects. However, despite the general validity of the mathematical theory of size functions in spaces of any dimension, their application to 3D objects has never been performed before, and is not straightforward, as discussed in Section 3. The aim of this paper is to start exploiting and enhancing the potential of size functions for 3D shape comparison. The result is the definition of an effective technique for 3D shape description and retrieval, taking into account structure, topology and geometry. This paper is an evolution of the short paper presented in June 2006 at the Eurographics Symposium on Geometry Processing [16].

1.1 Paper contribution

The major contribution of the paper is the introduction of a method to apply size functions to the 3D shape matching context. The key idea is to associate a topological space S to a 3D shape M , where S is simpler than M in terms of dimensionality, while still preserves the main geometrical, structural and topological properties of the shape. This lower dimensional space S , or compact representation of M , is the input for the computation of size functions and the successive comparison process. We show the results obtained by using a 1D topological structure, called the *skeletal graph*, as an auxiliary space to study a 3D shape M .

The basic steps of the proposed method are:

- the definition of a suitable *skeletal graph* to describe 3D shapes, based on the construction of a centerline skeleton;

- the definition of a set of *measuring functions* on the skeletal graphs, which capture quantitative attributes of the shape;
- the computation of the size functions of the attributed skeletal graphs;
- the evaluation of the similarity between two models, through a suitable distance between their size functions.

Since the computation of size functions on 1D structures is very efficient, it is possible to use a varied set of attributed skeletal graphs – called *size graphs* – each reflecting the description of M with respect to some criteria.

For the construction of the skeletal graph, we adopt a schema grounded in Morse theory, based on the computation of the level sets of different real-valued functions (referred to in what follows as *mapping functions*) computed on the shape. The level sets drive the construction of the 1D centerline skeleton of the shape M , and also support the computation of several geometric attributes (referred to as *measuring functions*). The possibility of computing level sets of different mapping functions and the analysis of different geometric attributes allow us to build a set of descriptors, that capture a variety of shape properties, making our retrieval method a general and flexible framework. In other words, the system modularity allows the user to fit a descriptor to the specific nature of the object to be found, through the choice of the real functions (namely, mapping and measuring functions).

Moreover, since our size graphs are a skeletal representation of the shape, another interesting contribution of the paper is the introduction of a distance function for graph representations of 3D models. In this case, the distance between structural representations does not rely on the computation of exact or approximate (sub-)graph isomorphisms. Indeed, we replace the problem of graph matching (*NP*-hard in the classical formulation) with the simpler algebraic comparison of the size functions associated with the graphs.

The remainder of the paper is organized as follows. In Section 2 we briefly overview existing techniques for 3D object retrieval. After introducing size functions and discussing their use for 3D shapes in Section 3, we detail our approach to obtain informative size functions from 3D models in Section 4. The shape comparison process is outlined in Section 5. Relationships with related approaches in the literature are discussed in Section 6. Section 7 is devoted to discuss experimental results, and comparative remarks with other methods are also provided. Conclusions and suggestions on future work end the paper.

2 Related work

The majority of the methods proposed in the literature for 3D shape retrieval mainly focus on geometric descriptions, in the sense of considering the spatial distribution or extent of a shape in the 3D space [17–21]. For example, in [21] a rotation invariant descriptor is proposed, based on spherical harmonic decompositions. Models are beforehand voxelized and scaled so that the center of mass is placed at the origin of a volumetric grid. Then objects are sampled and represented through the vectors of coefficients of their spherical harmonics.

Other popular techniques use image-based descriptors [22,23]. For instance, the approach in [22] is based on the idea that objects are similar when they look similar from different viewing angles. A so-called *lightfield* descriptor is extracted from a set of silhouettes, which are obtained by projecting data using an array of surrounding cameras.

In all previous cases, 3D data are indexed using *numerical* descriptors referred to as *feature vectors*. Once these vectors are extracted, the similarity between two models can be inferred from the distance between their numerical descriptions in some metric space, usually the Euclidean one. From a practical point of view, the main advantage of these methods is that they are computationally efficient and do not make specific assumptions on the topology of the shape models, which might be even triangle soups.

Nevertheless, as shapes are mentally coded in terms of relevant parts and their spatial configuration, or structure, there is a growing consensus towards high-level descriptors, which merge a global topological analysis with local geometric attributes. This approach is followed for example in [24,25], where the notion of *barcodes* is introduced in the framework of persistent homology. The authors examine a complex X by first constructing a space strictly related to X , namely the tangent complex (that is, the closure of the space of all tangents to all points in X). This complex is filtered by the increasing values of a real function, defined as the curvature at a point along the tangent direction. The placement of topological events occurring during the filtration is coded by a set of intervals, representing the life-time of topological attributes. A suitable pseudo-metric between barcodes allows for a measure of the similarity between shapes. Examples are shown for curve point cloud data [24], geometric surfaces and surfaces of revolution [25].

The method presented in [26] addresses 3D shape similarity by using the Reeb graph in a multi-resolution fashion, and it makes use of graph-matching techniques for the retrieval. Another approach to shape retrieval has been proposed in [27]. In this work, a so-called *Extended Reeb Graph* representation is coupled with a geometric description of the shape subparts related to the

graph nodes. Different weights are associated to different shape aspects, either geometric or structural. The framework described is also suitable to identify partial correspondences of 3D shapes [28].

Similarly, the importance of structural descriptions for shape matching has been pointed out in [29–32]. The *relevance* of the skeletal edges incident in a node (e.g. edge length, diameters and average circumference of the skeleton loops) is coded in a vector [33,34] or in a geometric descriptor able to support either global [31] or partial [32] correspondence.

In all cases, the structural descriptors are coupled with different methods to compute or approximate subgraph isomorphisms [35], node correspondence [26,31], maximal cliques [36], or to evaluate graph spectra [37]. On the contrary, in our proposal the structural descriptors are compared in a completely different approach, that takes advantage of size theory.

Exhaustive surveys on 3D shape searching techniques can be found for example in [17,38,39].

3 Size functions

The main idea in size theory is to compare shape properties that are described by real functions, defined on topological spaces associated to the “objects” to be studied [7,9,40]. This leads to considering pairs (S, φ) , where S is a topological space and $\varphi : S \rightarrow \mathbb{R}$ is a continuous function. Every pair is called a *size pair*, while each function φ is called a *measuring function* and its purpose is to encode quantitative properties of the shape S .

Given a size pair (S, φ) , the size function

$$\ell_{(S,\varphi)} : \{(x, y) \in \mathbb{R}^2 : x < y\} \rightarrow \mathbb{N}$$

can be easily defined by setting $\ell_{(S,\varphi)}(x, y)$ equal to the number of connected components of the lower level set $S_y = \{P \in S : \varphi(P) \leq y\}$, containing at least one point of S_x .

An example of size function is shown in Figure 1. In this example we consider the size pair (S, φ) , where S is the curve represented by a continuous line in Figure 1 (a), and φ is the function “distance from the point P ”. The size function associated with (S, φ) is shown in Figure 1 (b). Here the domain of the size function is divided by solid lines, representing the discontinuity points of the size function. These discontinuity points divide the set $\{(x, y) \in \mathbb{R}^2 : x < y\}$ into regions where the size function is constant. The value displayed

in each region is the value taken by the size function in that region. For instance, for $\alpha \leq x < \beta$, the set S_x has two connected components which are contained in different connected components of S_y when $x < y < \beta$. Therefore $\ell_{(S,\varphi)}(x,y) = 2$ for $\alpha \leq x < \beta$ and $x < y < \beta$. When $\alpha \leq x < \beta$ and $y \geq \beta$, all the connected components of S_x are contained in the same connected component of S_y . Therefore $\ell_{(S,\varphi)}(x,y) = 1$ for $\alpha \leq x < \beta$ and $y \geq \beta$. Similarly, we find $\ell_{(S,\varphi)}(x,y) = 3$ for $\beta \leq x < \gamma$ and $x < y < \gamma$ and $\ell_{(S,\varphi)}(x,y) = 1$ when $\beta \leq x < \gamma$ and $y \geq \gamma$.

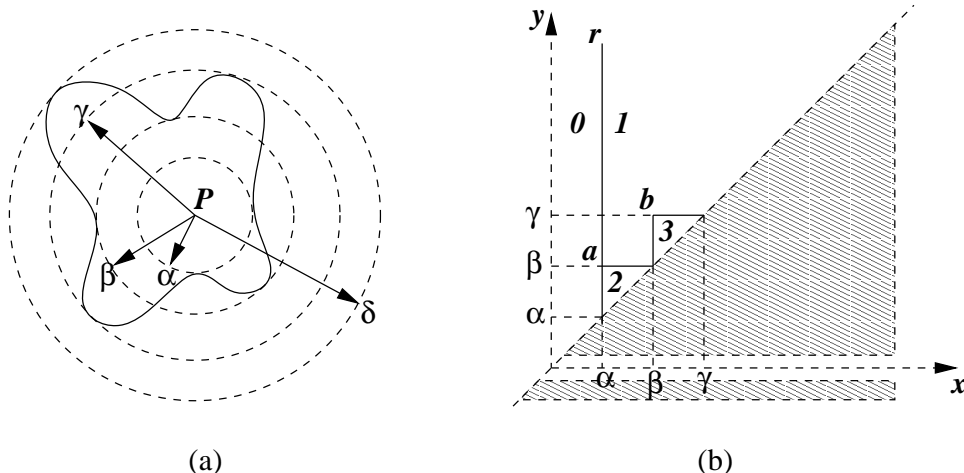


Fig. 1. (a) A size pair (S, φ) , where S is the curve represented by a continuous line and φ is the function “distance from the point P ”. (b) The size function of (S, φ) .

An important property of size functions is that they can always be seen as linear combinations of characteristic functions of triangles (possibly unbounded triangles with vertices at infinity), with a side lying on the diagonal $\{x = y\}$ and the other sides parallel to the coordinate axes [41]. For instance, the size function of Figure 1 is the sum of the characteristic functions of the triangles with right angles at vertices a and b , plus the characteristic function of the infinite triangle on the right of the line r . This suggests that the size function is completely determined by a, b, r . In fact, the property that size functions can be represented as collections of vertices (called *cornerpoints*) and lines (called *cornerlines*) always holds [42]. This provides a simple and concise representation for size functions in terms of points and lines in \mathbb{R}^2 , drastically reducing the required descriptive dimensionality. This representation also allows for the comparison of size functions using distances between sets of points and lines, as discussed in Section 5.

Another nice property useful for applications is that size functions inherit the invariance properties, if any, of the underlying measuring functions.

It is worth noticing that size functions actually coincide with the 0-th persistent Betti numbers introduced in [11], and, in the case of 0-th degree homology, the formal series of cornerpoints and cornerlines describing size func-

tions [41,42] substantially coincide with the persistence diagrams introduced in [43]. Analogously, the formal series representation of size functions satisfies a property of stability under perturbations of the data [44,45], and the same properties remains true for persistence diagrams [43].

The discrete counterpart of a size pair is given by a *size graph* (G, φ) , where $G = (V(G), E(G))$ is a finite graph, with $V(G)$ and $E(G)$ the set of vertices and edges respectively, and $\varphi : V(G) \rightarrow \mathbb{R}$ is a measuring function labeling the nodes of the graph [46,47].

3.1 Defining informative size functions for 3D shapes

In the context of shape modelling, 3D models are often represented as surfaces, defined by triangle meshes. In principle, the size graph might be made of the vertices and the edges of the triangle mesh. Beside the large size of the graph that would affect the computational cost, there is the problem of finding measuring functions on the mesh, which capture significant features of the 3D object.

The latter question is really non-trivial: while in principle the space dimension can be arbitrarily large, in practice the performance of size functions may be weaker if they are directly applied to the given space. As an example, let us consider the two different representations of the cactus shape shown in Figure 2 (a): a planar 1D contour sketching the profile (left) and a 2D surface (right). If we describe the 1D contour by means of the height function, we obtain a size function which is sufficiently informative about the shape (see Figure 2 (b-left)). The same description for the 2D surface produces a trivial size function (see Figure 2 (b-right)), since the number of connected components for the lower level set $\{P : \varphi(P) \leq y\}$ is always 1, for any value of y , $y \geq \min_P \varphi(P)$. The solution comes from the identification of the “right” set of shape properties, i.e. the “right” size pairs, that are suitable for the problem at hand.

4 3D size graphs

To overcome the limits of a straightforward application of size functions to 3D shapes, our idea is to associate with a 3D object a size graph (G^f, φ) , where G^f is a skeletal graph representing the 3D object, f is a real function driving the skeleton extraction and called the mapping function, and φ is a measuring function labelling each node of the graph with local geometrical properties of the original model. The replacement of a 3D shape with this auxiliary space

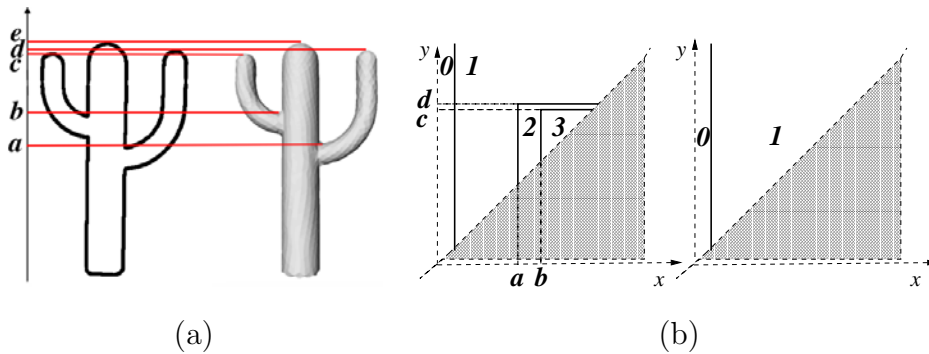


Fig. 2. (a) Two different representations of a cactus shape and (b) their corresponding size functions.

(G^f, φ) , which couples the structural information computed by the function f with the different information provided by the measuring function φ , produces non-trivial size functions. The use of (G^f, φ) reduces the dimensionality of the problem, meanwhile storing a rich set of information about the original object.

In the following, we describe in details the three ingredients that contribute to the definition of size graphs (G^f, φ) : the procedure for building the skeletal graph G^f starting from the level sets of a given function f ; the choice of an appropriate mapping function f ; the definition of a set of measuring functions φ to associate the most appropriate and effective geometric descriptors with the skeletal graph.

4.1 Constructing the skeletal graph

The construction of the centerline defining the skeletal graph G^f is based on the computation of the barycenters of the connected components of the level sets of a real-valued mapping function f defined on the shape under study. The idea is that the evolution of the level sets follows the shape, and their barycenters trace ideal skeletal lines that represent features captured by the level sets, that is, by f . In particular, the variability of the choice of f (see discussion in Section 4.2) differentiates our skeletal graph from the *Level Set Diagram* proposed in [48].

Differential topology results for surfaces [49] guarantee that this construction is well-defined, and most importantly, that the skeletal graphs G^f are topologically valid representations of the original shape, meaning that the connectivity of the shape is preserved by G^f . More precisely, this property derives from the fact that the construction of the skeletal graphs relies on the discretization of the Reeb graph theory defined in [50] and [51].

The example in Figure 3 illustrates the pipeline of the skeleton extraction.

Given a shape represented by a regular, closed triangle mesh M , we first scale M in the unit sphere centered in the origin of the Cartesian coordinates. Let $f : M \rightarrow \mathbb{R}$ be a real continuous mapping function, and let $[f_{min}, f_{max}]$ be the co-domain of f (Figure 3 (a)). We can choose nv non-critical values f_i , $f_i \in [f_{min}, f_{max}]$, $i = 1, \dots, nv$, such that $f_{min} < f_1 < \dots < f_{nv} < f_{max}$. Let $C(M) = \{C_i(M)\}_i$, $i = 1, \dots, nv$, be the set of the resulting level sets of the surface M , where $C_i(M) = f^{-1}(f_i(M))$. Since the f_i values are regular values, each contour (i.e., each connected component of the level sets) is a simple, closed line and $C(M)$ fully decomposes M into a set of regions (Figure 3 (b)). Let T be the partition of the interval $[f_{min}, f_{max}]$:

$$T = \{(f_i, f_{i+1}), i = 0, \dots, nv\} \cup \{f_{min} = f_0, f_1, \dots, f_{nv}, f_{nv+1} = f_{max}\}$$

induced by the f_i values. In other words, T is made of a set of contiguous open intervals together with the set of their extrema.

Two points $P, Q \in M$ are identified when the following two conditions hold:

- (1) $f(P)$ and $f(Q)$ belong to the same element t of the partition T ;
- (2) P and Q are in the same connected component of $f^{-1}(t)$, $t \in T$.

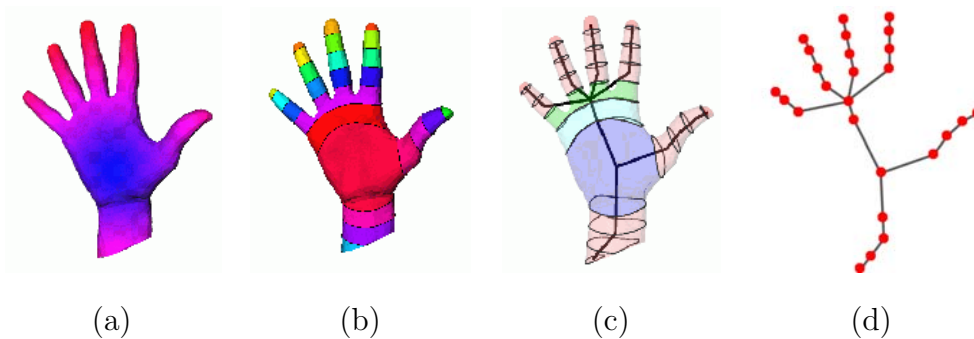


Fig. 3. Pipeline of the skeletal graph extraction. (a) Evaluation of the mapping function (the distance from the barycenter in this example) on the hand model in [52]. Red and blue colors respectively represent maximum and minimum values. (b) The shape segments after the insertion of the level sets. (c-d) The centerline skeleton, superimposed to the model in (c).

Therefore, all points belonging to either a region or a connected component of the level sets are equivalent and are identified as the same point of the quotient space obtained from this relation. Then, the quotient space can be represented as a traditional graph $G^f = (N, A)$ where the set of nodes and arcs are defined by

$$N = \{\text{classes of equivalence of regions}\}$$

$$A = \{\text{classes of equivalence of connected components of level sets}\}.$$

Denoting p and q the nodes that correspond to the regions R and R' , respectively, an arc connects p and q if the boundaries of R and R' share at least one connected component. The node p is adjacent to as many nodes as the number of connected components of the boundary of R and, conversely, an arc is defined for each connected component of the border of R .

The graph G^f can be embedded in \mathbb{R}^3 by associating with each node the position of the barycenter of the corresponding region, and visualized as a centerline skeleton of M (Figure 3 (c-d)).

G^f can also be extracted in a multi-resolution fashion, by iteratively refining the interval $[f_{min}, f_{max}]$, and extracting a sequence of graph structures. The power of this approach is clear: the surface shape can be processed at different levels of detail, and the estimation of its features is automatically provided. An analogous approach has been proposed in [26], where the intervals are iteratively halved; see Section 6 for comparative remarks.

If a closed manifold is considered, Morse theory guarantees that the number of non-homologous loops in the manifold corresponds to the number of cycles in the skeletal graph. From an implementation point of view, when dealing with a mesh M , we ensure to keep this topological consistency by using the Euler formula in each region, in order to check that the region does not contain any through hole. This control ensures that we accept only skeletal structures that have as many cycles as the number of tunnels of the model.

From the computational point of view, the graph construction begins with the computation and insertion of the set of level sets $C(M)$ into the triangle mesh. This step requires $O(\max(m, n \log(n)))$ operations, where m is the number of vertices added inserting $C(M)$ and n is the number of vertices of the original triangulation. A node is created for each region, and an arc is associated to each boundary component. Then the inserted isocontours are used as constraints during the graph extraction process. The latter is based on a region growing strategy, that visits all triangles. Hence, the cost for the region detection and arc completion step is $O(m+n)$. Therefore, the complexity of the whole process is given by the maximum of the costs of the two algorithm phases, that is $O(\max(m + n, n \log(n)))$.

4.2 *Choosing the mapping functions*

In order to make the skeletal graph G^f valid and effective for shape description, the mapping function f should be independent of object rotation, translation and uniform scaling. Given these general constraints, the choice of the mapping functions is arbitrary and should be driven by the suitability of the mapping function to describe the shape, according to the geometric properties that we

would like to enhance. For example, functions based on the Euclidean distance from a point P naturally highlights the distribution of the object with respect to P , while those based on the geodesic distance¹ are independent of the spatial embedding. Indeed, the resulting skeletal graphs follow the protrusions until they reach their tips, corresponding to the maxima of the functions.

In our experiments, we have used four different mapping functions, namely the distance from the barycenter, the distance from the center of the bounding sphere, the integral geodesic distance [26] and the topological distance from curvature extrema [53]. The geometric characterization of these mapping functions is discussed after their definition below.

Let v be a vertex of the mesh; the following two distance functions f^1, f^2 are considered:

- $f^1(v) = |v - P_B|_E$, see Figure 4 (a);
- $f^2(v) = |v - P_S|_E$, see Figure 4 (b);

here $|\cdot|_E$ is the Euclidean length, and P_B and P_S are the barycenter of the object and the center of its bounding sphere, respectively. These two points are suitable choices as reference points, because they are easy to calculate and independent of the embedding of the shape in the space. The level sets, which induce the surface decomposition into regions, correspond to the intersection of the mesh with a collection of spheres centered in the barycenter and in the center of the bounding sphere, respectively, and with varying radius. Since the barycenter or the center of the bounding sphere and the sphere/mesh intersection are independent of translation, rotation and uniform scaling of the object, the same properties are reflected in the resulting graph configurations.

The next mapping function considered is the normalized integral geodesic distance, as discretized in [26]:

- $f^3(v) = \sum_i g(v, b_i) \cdot area(b_i)$, see Figure 4 (c);

here $\{b_i\} = \{b_0, \dots, b_k\}$ is an almost uniform sampling of the vertices of M , $g(v, p_i)$ is the geodesic distance of point v from point b_i , and $area(b_i)$ is the area of the neighborhood of b_i . In practical cases, when the triangle mesh is not uniform, the use of Dijkstra’s algorithm [54] may alter the approximation of the geodesic distance and the choice of the basis points.

The fourth mapping function is based on the geodesic distance from curvature extrema, and is similar to the discrete topological distance originally proposed in [53]:

¹ Here, the geodesic distance between two given surface points p and q is the minimal length of all surface curves joining p and q .

- $f^4(v) = -\min_{i=1,\dots,n}\{g(v, p_i)\}$, see Figure 4 (d);

here $\{p_1, \dots, p_n\}$ is a set of vertices that represent mesh regions having high curvature values, and g denotes the geodesic distance. In the evaluation of f^4 , the geodesic distance is approximated through Dijkstra’s algorithm.

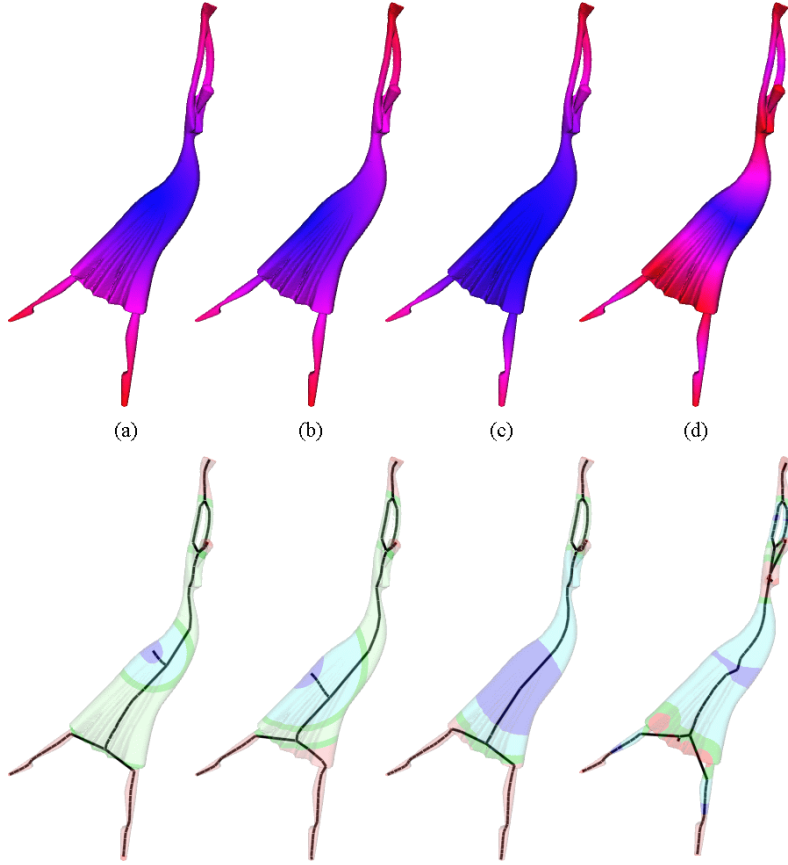


Fig. 4. (Top) From left to right, the evaluation of the mapping functions f^1 , f^2 , f^3 and f^4 on the same model from [52]. (Bottom) The corresponding skeletal graphs, obtained by inserting fifteen level sets in the meshes.

Since the geodesic distance relies neither on a local coordinate system nor on surface embedding, in both cases of f^3 and f^4 the graph configuration derived is invariant to translation, rotation and uniform scaling of the model.

As anticipated, important aspects to evaluate are the kind of features that the description should highlight, and the type of matching wished. In our example, f^1 and f^2 naturally emphasize the distribution of the object with respect to a point. Therefore these functions are rotation invariant, but sensitive to pose variations, where by pose variation we mean a different spatial arrangement, or embedding, of the articulations of a model. On the contrary, functions f^3 and f^4 are pose invariant, because they depend on the shape distribution with respect to the geodesic center of the surface (f^3) or the high curvature points (f^4). In all cases, the shape is described as a configuration of protrusions and

hollows, but the geodesic distances do not discriminate between objects in different poses, while Euclidean distances do. Therefore, geodesic distances are best suited for retrieving articulated objects disregarding the pose, while Euclidean ones allow us to distinguish among articulated models in different poses.

4.3 Measuring functions

Measuring functions in size theory have the role of capturing relevant geometric properties of the shape, to be used as distinctive properties to discriminate among shapes. Once the skeletal graph G^f has been extracted, the *size graph* (G^f, φ) is obtained by defining the measuring function $\varphi : V(G^f) \rightarrow \mathbb{R}$ on the set $V(G^f)$ of nodes of G^f .

For each node $v_R \in V(G^f)$ corresponding to a region R , the value of $\varphi(v_R)$ is defined as a property characterizing the region R or its boundary, as depicted in Figure 5. Hence, we analyze features of the original 3D object, storing them in our compact representation. Moreover, besides geometric attributes of the model, we also consider features describing the spatial embedding of the shape in the 3D space. The set of proposed measuring functions is detailed in the following.

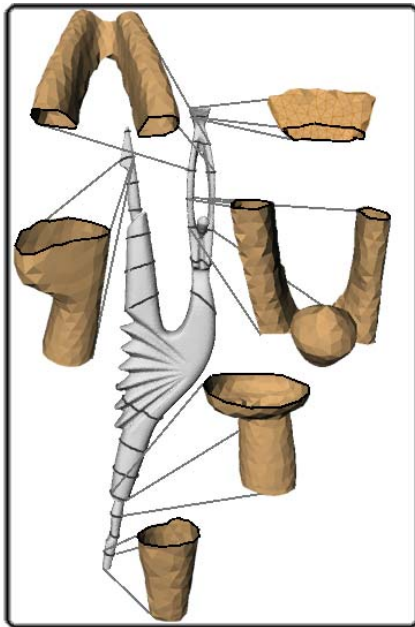


Fig. 5. Some segments associated to the size graph nodes (graph obtained by segmenting the mesh with the insertion of seven level sets).

For each node $v_R \in G^f$ associated to a region R , a first measuring function $\varphi(v_R)$ can be defined as:

- the *area* of the region R ,

that is to say the sum of the areas of the faces in the triangulation which belong to R .

Three other measuring functions are given by:

- the *minimum* (r_{min}), *the maximum* (r_{max}) and *the average* (r_{av}) *radius* of R , i.e., the minimum, maximum and average distance of the barycenter C of the triangles in R from the vertices of the region, see Figure 6 (a).

Since the boundary $B_M(R)$ of a region R is made of closed contours, the interior of R is well defined and it is possible to associate to each boundary component a so-called outgoing direction. An outgoing direction is classified as ascending or descending, according to the behavior of the mapping function f across the corresponding boundary component: the direction is ascending (resp. descending) if the value of f increases (resp. decreases) walking from the inside to the outside of the region. Let us now denote $B_M^+(R)$ (resp. $B_M^-(R)$) the set of connected components of $B_M(R)$ such that the outgoing directions for the mapping function f are ascending (resp. descending), see Figure 6 (c). For each node v_R corresponding to a region R , two measuring functions can now be defined as:

- the *length* of $B_M^+(R)$ (resp. $B_M^-(R)$).

If $B_M^+(R)$ (resp. $B_M^-(R)$) is made up of multiple components $\{B_i^+\}$ (resp. $\{B_i^-\}$), then $\varphi(v_R)$ is the sum of the length of each of the B_i^+ s (resp. B_i^- s). If all outgoing directions of f for $B_M(R)$ are ascending (resp. descending), we label the region R as a minimum (resp. maximum) for the function f and assume the length of the border $B_M^+(R)$ (resp. $B_M^-(R)$) to be zero, that is $\varphi(v_R) = 0$.

The next measuring functions evaluate the lateral area of the pseudo-cone whose basis is the boundary component B_i^+ (resp. B_i^-) and whose vertex is the barycenter of B_i^+ (resp. B_i^-). Two measuring functions are indeed defined as

- the sum of the *pseudo-cone areas* computed for each B_i^+ in $B_M^+(R)$, (resp. B_i^- in $B_M^-(R)$), see Figure 6 (b).

When all outgoing directions of f for $B_M(R)$ are ascending (resp. descending) we set $\varphi(v_R) = 0$.

A further description of the object can be given by considering its location in space. As previously discussed, each node v_R is located in the barycenter (x_R, y_R, z_R) of the corresponding region R . In addition, for each model, we compute the minimal bounding box following the algorithm in [55], which

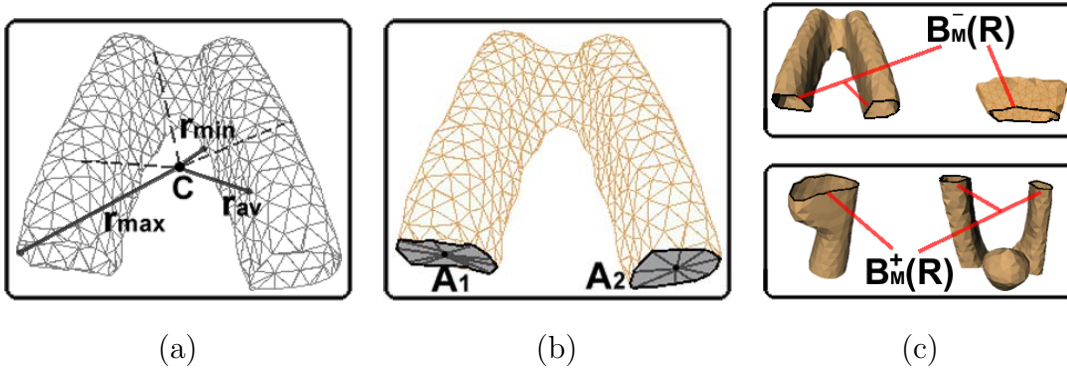


Fig. 6. Details about the geometric meaning of some measuring functions. (a) Minimum, maximum and average radius of a region. (b) A_1 and A_2 represent two pseudo-conic areas. (c) Length of upper and lower (with respect to a chosen f) boundary components.

provides a bounding box oriented towards the principal direction of the object (as shown in Figure 7). The faces F_i , $i = 1, \dots, 6$, of the bounding box are ordered according to their increasing areas, thus obtaining the ordered set of pairs $\{(F_1, F_2), (F_3, F_4), (F_5, F_6)\}$, where each pair corresponds to a pair of opposite faces, having the same area. Then, the center P_i is evaluated for each face F_i , $i = 1, \dots, 6$, see Figure 7 (b), and a set of measuring functions $\{\varphi_{P_i}\}$ can be constructed by computing the Euclidean distance $\varphi_{P_i}(v_R) = |v_R - P_i|_E$. The invariance with respect to axial symmetry is obtained replacing the set $\{\varphi_{P_i}\}$ by:

- the set of measuring functions $\{\varphi_{P_i, P_{i+1}}\}$, with $i = 1, 3, 5$, and such that $\varphi_{P_i, P_{i+1}}(v_R) = \min\{|v_R - P_i|_E, |v_R - P_{i+1}|_E\}$,

where the minimum distance is considered between the node v_R and the centers of two opposite faces.

5 3D model comparison

After a size graph (G^f, φ) has been obtained, with G^f the skeletal graph and φ the measuring function labeling its nodes, the definition of the size function of the size graph follows the classical one. Denoting by G_y^f the subgraph of G^f obtained by erasing all vertices of G^f at which the measuring function φ takes a value strictly greater than y , and all edges that connect those vertices to other vertices, the size function of the size graph (G^f, φ) is defined by setting $\ell_{(G^f, \varphi)}(x, y)$ equal to the number of connected components of G_y^f , containing at least a vertex of G_x^f .

In order to compute size functions, we follow the algorithm introduced in [47]. The algorithm is based on a technique, called Δ^* -reduction, that permits a

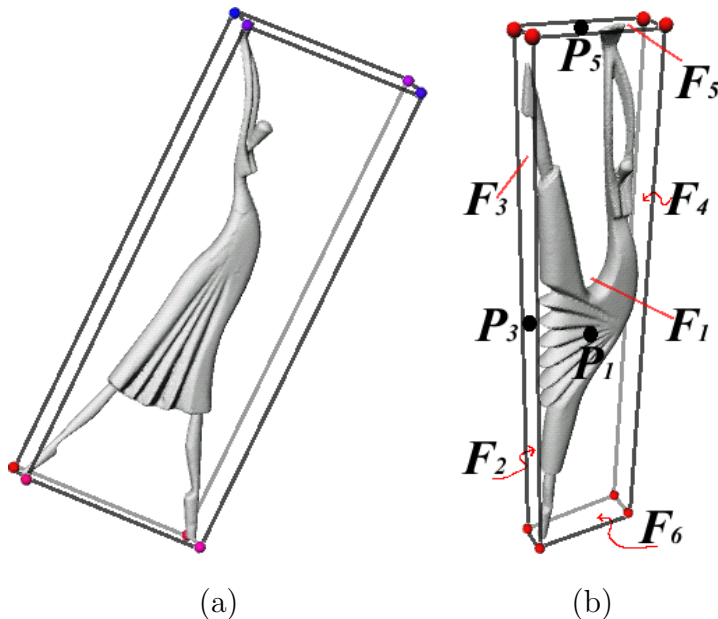


Fig. 7. (a,b) Minimal bounding boxes of two models.

simplification of the size graph, without changing the corresponding size function. This procedure is defined by recursively applying three different graph editing moves, see [47] for details. The process works in a finite number of operations, and its output is a graph that has the simple structure of a tree. The value of Δ^* -reduction relies not only on reducing the number of vertices and edges in the graph, but also on permitting faster computation of size functions [47]. In fact, Δ^* -reduction permits direct computation of the cornerpoints and cornerlines, that completely describe the size functions (see Section 3). To achieve this, one has to orient the reduced graph obtained through the process of Δ^* -reduction by orienting each edge from the vertex with higher value to the other one. The resulting configuration is an arborescence, i.e. an oriented tree in which no two edges are directed to the same vertex. Finally, the cornerpoints and cornerlines are computed from this arborescence by applying the following recursive procedure: (i) choose the highest leaf v and erase it together with the corresponding edge wv ; (ii) put a cornerpoint at $(\varphi(v), \varphi(w))$; (iii) if just one vertex u is left, then draw the cornerline $x = \varphi(u)$ and stop, otherwise repeat from (i).

A possible implementation for this algorithm is based on the union-find structure [56], so that the computational complexity for the whole procedure is $O(n \log n + m \cdot \alpha(2m + n, n))$, where n and m are the number of vertices and edges in the size graph, respectively, and α is the inverse of the Ackermann function [57].

Two size functions represented by formal series of cornerpoints and cornerlines can be compared by using different metrics [58], as for example the Hausdorff metric. In this way, the complex issue of shape matching is translated into the

simpler (algebraic) problem of comparing formal series.

In this paper, we use the *matching distance* between size functions, which is detailed in [40]. Informally speaking, the matching distance between two reduced size functions l_1 and l_2 , respectively represented by the sequences (a_i) and (b_i) of cornerpoints and cornerlines, can be defined as:

$$d_{match}(l_1, l_2) := \min_{\sigma} \max_i d(a_i, b_{\sigma(i)})$$

where i varies in the set \mathbb{N} of natural numbers, σ varies among all the bijections from \mathbb{N} to \mathbb{N} , and the pseudo-metric² d between two points p and p' takes the smaller between the cost of moving p to p' and the cost of moving p and p' onto the diagonal $\{x = y\}$, with costs induced by the max-norm [40]. Notice that d_{match} is actually a metric. An example is shown in Figure 8 (right), where an optimal matching between cornerpoints and cornerlines of two size functions is shown. Here the matching distance is given by the cost of moving the cornerpoint denoted by b onto the diagonal.

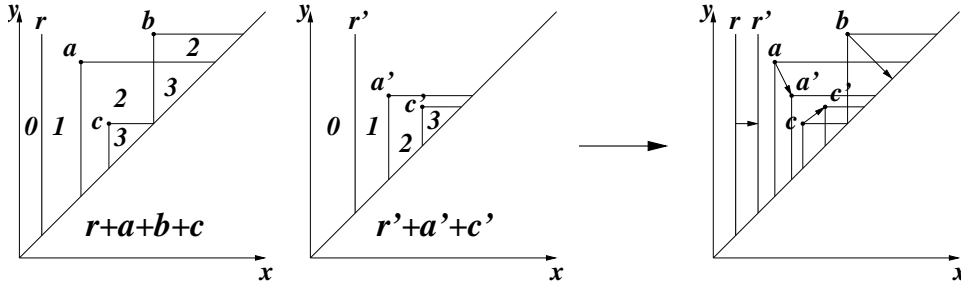


Fig. 8. Two size functions described by formal series of cornerpoints and cornerlines and compared by means of the matching distance.

The stability of the matching distance and its suitability for shape comparison have been discussed from the theoretical point of view in [44,45,40]. In particular, it has been proven that the matching distance between size functions is continuous with respect to the measuring functions, guaranteeing a property of robustness to perturbations of the data, that is useful in practical applications. Moreover, in [45] it has been shown that the matching distance between size functions produces a sharp lower bound for the *natural pseudo-distance* between size pairs [59], thus guaranteeing a link between the comparison of size functions and the comparison of shapes.

Since only a finite number of cornerpoints may occur in the discrete case, computing the matching distance between discrete size functions corresponds to solving an optimal matching problem between finite point sets. The com-

² A pseudo-metric fulfills the properties of a metric of being non-negative, being symmetric and satisfying the triangle inequality, but allows the distance between two different points to be zero.

putational cost is $O(k^{2.5})$, where k is the number of cornerpoints considered for each size function.

To summarize, Figure 9 shows an example of the pipeline of our comparison technique, from the graph extraction to the evaluation of the distance between the models. Figure 9 (a) shows two models from [52], representing two instances of a dancer. The first step is the extraction of the skeletal graph, as in Figure 9 (b), where the integral geodesic distance is used to drive the extraction. Next, we evaluate the geometric attributes of the nodes, the average radius of surface regions in this example. Hence, once the size graphs have been constructed, we compute their size functions, as in Figure 9 (c), and evaluate their matching distance. Notice how the similarity between the two models is reflected in a similar structure for the corresponding size functions, and results in a small value for the matching distance.

6 Comparison with existing techniques

In this section we compare our technique with existing approaches. We take into account both phases of shape description and similarity evaluation.

From a general perspective, a shape descriptor closely related to ours is given by the barcodes in [24,25], see also Section 2. In particular, beside the mathematical relationship between persistent homology theory and size theory [60], it is interesting to note that both approaches study a shape by constructing a new space strictly related to the original shape: the centerline skeleton in our approach, the tangent complex for the barcodes.

As for the graph-based representation we propose, our algorithm for centerline extraction provides a shape segmentation, that enables us to easily couple the regions of the surface with a set of geometric attributes, as detailed in Section 4.3. Thus, our size graph may be seen as a centerline combining both topological information and geometric attributes. It is worth observing that also another skeletal centerline that roughly approximates the medial line of a shape (like those proposed in [33,30]) could in principle be chosen as the basis for the size graph. However, in our framework we have tried to keep each module of the system as flexible as possible, as discussed in Section 7.3. For the same reason, our technique also differs from the structural descriptors based on the medial centerline of the shape, such as [31], that provides a more rigid coding based on the distance transform.

With respect to similar approaches based on Reeb graphs [61,27], the nodes of the Reeb graph correspond only to critical points, while we take into account not only the configuration of critical level sets, but also the intermediate

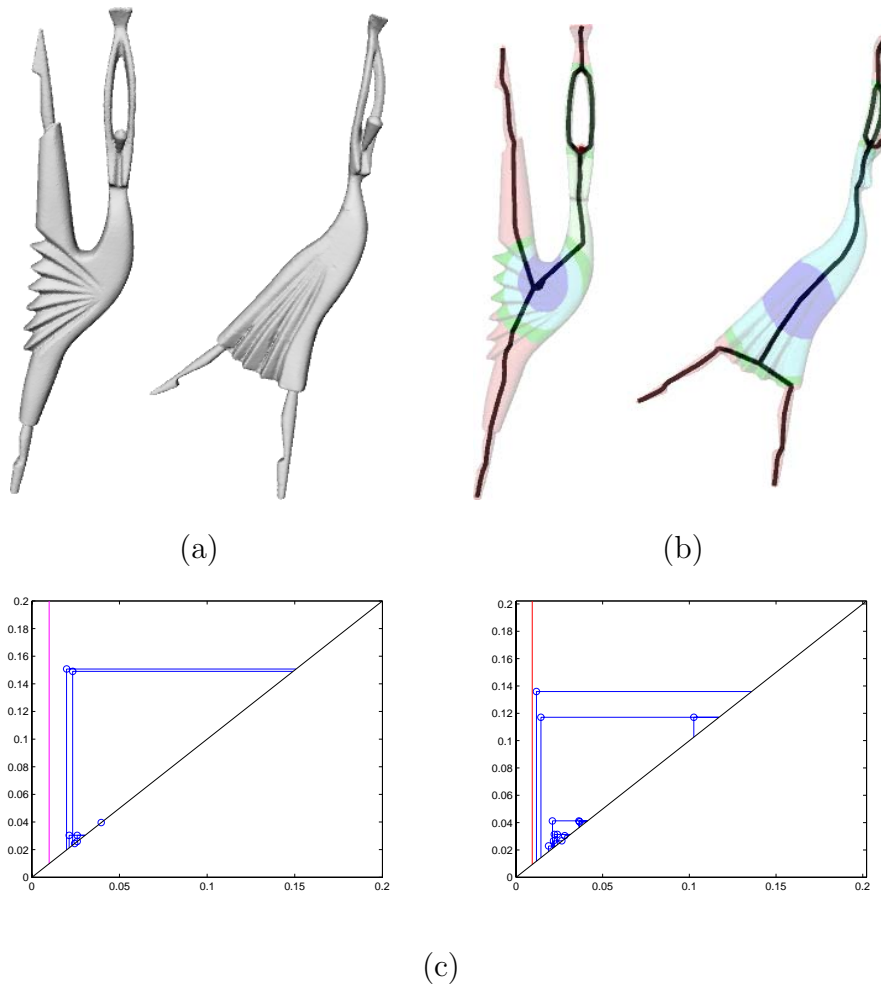


Fig. 9. (a) Two models from [52]. (b) The skeletal graphs with respect to the integral geodesic distance (fifteen level sets inserted). (c) The corresponding size functions, when the measuring function is the average region radius. Notice the two points far from the diagonal, which are very close one to the other in the size function on the left and correspond to a similar structure in the size function depicted on the right.

non-critical level sets. Our centerline also differs from the *Multiresolution Reeb Graph*, described in [26] and extended in [62], both in the way the skeleton is extracted – our graph is not necessarily multi-resolution, and is also topologically consistent, being the number of loops in the graph equal to the number of tunnels – and in the way the geometric attributes are chosen, see discussion in Section 4.3. In particular, our measurements mainly describe the geometry of the surface slices obtained during the contouring phase, and extend the attributes proposed in [26], where the ratio of the area and the length of the model sub-part with respect to the whole model are considered. Moreover, our measuring functions estimate the properties of the surface slices, unlike the attributes proposed in [62], that are mainly devoted to a volumetric description of the slices. In this way, it is possible to avoid the ambiguity that arises when dealing with volumetric slices with non-planar bases.

Finally, with regard to the similarity evaluation framework, previous works on graph-based representations usually rely on graph matching strategies or graph spectra evaluations. Graphs are compared involving error tolerant graph or subgraph isomorphisms, that take into account editing operations, and are often well suited for partial shape matching [26,27]. On the contrary, in our approach the comparison strategy is based on the computation of the matching distance between the size functions associated to the graph representations, which in turn corresponds to the comparison of finite point sets.

7 Experimental results

This section aims to provide a qualitative analysis of the shape comparison approach proposed, and a quantitative evaluation of the performance of our method in the retrieval context.

We first introduce in Section 7.1 the set of models which constitute our database. In Section 7.2 the robustness of our algorithm for shape description and comparison is discussed with the support of some examples. Section 7.3 investigates the crucial step of the choice of mapping and measuring functions, discussing their role in the retrieval context. Finally, we use our dataset to empirically compare the proposed method with four popular shape descriptors; retrieval results are reported in Section 7.4.

7.1 *The database*

The type and classification of the models in a database are crucial when testing a retrieval method, and it is difficult to separate the influence of the data set from the performance [63]. One of the most popular performance indicators is the precision-recall diagram [64,65], where recall indicates the ratio of the models in a given class returned within the top matches, and precision indicates the ratio of the top matches which are members of the correct class. Since the precision-recall representation is sensitive to the dimension of the classes [66], it is reasonable to require the number of elements of each class to be uniform when a retrieval method is compared with others. In addition, a detailed analysis of the pros and cons of a method may be done only if the classes of the database are well defined and the shape characteristics of the models belonging to the same class are distinguishable from the others. A further important requirement should be a uniform quality of the models. As observed in [63], when a large data set is composed of several smaller data sets to get a higher size, the main difficulty relies in the internal coherence of the large data set with respect to the coherence of its constituents.

Following these observations, we have constructed a database of regular triangle meshes with five classes of twenty elements, plus twenty unclassified manufactured models, see Figure 10. Many of the models represent articulated objects, reflecting a variety of complex poses. Moreover, shape classes show internal variations sufficient to make the retrieval task complex. This level of classification corresponds to the notion of a *basic level* or *entry level* categorization [5].

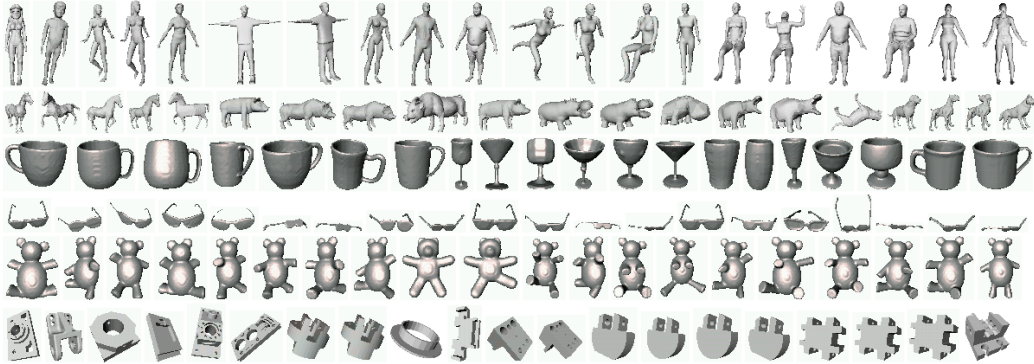


Fig. 10. Our testing models.

The original models of our database were collected from several web repositories. Most of the CAD models come from the National Design Repository at Drexel University [67]. Other models belong to the AIM@SHAPE repository [52] and the Princeton Shape Benchmark [3], while several human models in different poses come from the CAESAR Data Samples [68]. Since our graph extraction technique requires topologically correct meshes, when possible, we pre-processed and repaired models having many connected components by gluing them. Moreover, we also considered the database at the McGill 3D Shape Benchmark [69] that offers about 420 volume models, classified in 19 classes. Since the original models were voxelized and the resulting isosurfaces were too rough, we smoothed them using a Laplacian smoothing filter and selected 60 models from the “cups”, “teddy” and “glasses” classes, that present a uniform quality. In order to make results independent of scale operations, we uniformly scaled every model in a unit sphere, centered in the origin of the Cartesian coordinates.

The experiments reported in this section are performed through a uniform subdivision of the mapping function image $[f_{min}, f_{max}]$. In particular, the size graphs are obtained by inserting fifteen contour levels in correspondence of fifteen function values.

The normalization of the models in the same unit sphere and the level sets insertion with the same frequency implies that features are compared in a scale-independent manner, and according to their relative relevance in the

models under evaluation. Moreover, the rougher the slicing, the more the size graph recovers the main structure of the object. On the contrary, in case of a dense slicing, the size graph becomes more detailed and also small features may be recognized. The choice of a rough or a dense slicing depends on the application target, if the structure or the feature details must be recognized. However, when considering a dense slicing, small undesired features and arcs could appear in the graph: in these cases, a pruning strategy like that proposed in [50] allows us to discard irrelevant details.

7.2 Robustness

In this section we empirically show that our retrieval method is robust to noise and presence of small detail features.

Figure 11 (a) shows a model and its simplified version (having 10% of the vertices of the original model). The difference in the models neither significantly alters the attribute centerline skeleton (Figure 11 (b)) nor affect the corresponding size function (Figure 11 (c)), and the resulting matching distance is consistent. In practice, the method reveals to be robust from different perspectives:

- (1) *size graphs*: the attributed skeletal graph representation is stable under small variations of the mapping function f (Figure 11 (b)). In fact, it is possible to adopt a refinement of the slicing strategy, that guarantees that all features having size (in terms of the variation of the mapping function f) greater than a given threshold are detected, discarding the features whose size is smaller, see discussions in [51];
- (2) *size functions*: small changes of the size graph do not significantly alter the size functions (that can almost be superimposed in Figure 11 (c));
- (3) *matching*: the stability of matching distance between size functions under small perturbations of the data is confirmed by the example shown, where the matching distance between the size functions computed for the two models is very small.

The results of a second check for robustness are shown in Table 1, where the distances between six different objects in our database (four humans and two manufactured models) are reported. In particular, two of the human models are simplified versions of the other two ones, while the manufactured models differ for some small features. As expected, the comparison framework satisfies the identity property, guaranteeing that a model has a null distance from itself. In addition, the distance between a model and its slightly modified version is smaller than the distance between two different objects in the same class (see for example the woman and the man models) and significantly smaller than

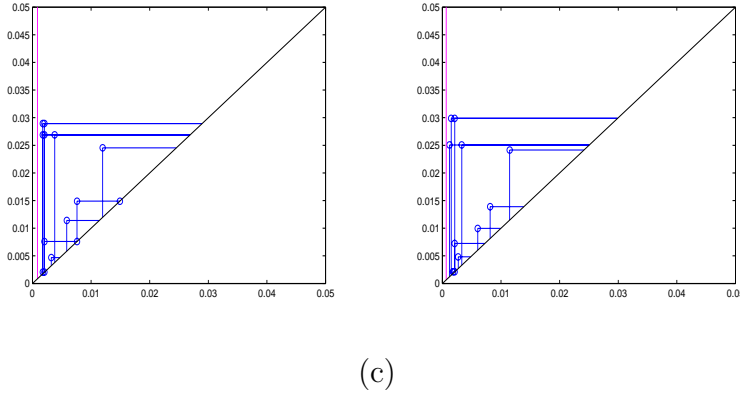
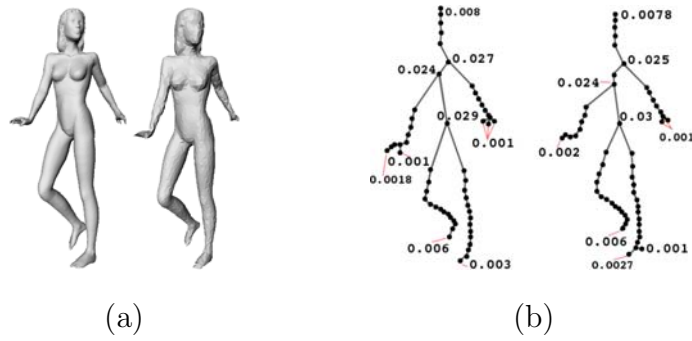


Fig. 11. Changes due to a simplification of a model do not significantly alter the centerline skeleton and its size function. (a) A model and its simplified version; (b) their size graphs with some attributes highlighted, using the integral geodesic distance f^3 and the measuring function $area$; (c) the corresponding size functions, which show very small variations.

the distance between objects belonging to different classes (e.g. a human and a manufactured model). We remark here that these distances are used to rank the results of database queries and aggregate the models that share analogous properties, although they cannot be interpreted as “absolute” values, due to their dependence on the values of the measuring function.

7.3 Modularity: the role of mapping and measuring functions

One of the attractive features of our approach is its flexibility, due to its geometric-topological nature and its capability to produce results which reflect human perception, by choosing the shape descriptors in a suitable way. In fact, the core idea of our method is the analysis of properties of real functions describing the shape under study. Real functions are involved for both the extraction of the skeletal graphs and the definition of the set of measuring functions, which serve for the computation of size functions. The role of the real functions is to take into account only the shape properties of the object which are relevant to the problem at hand, while disregarding the irrelevant ones, as well as to impose the desired invariance properties. Indeed, imposing







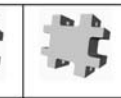
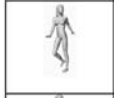
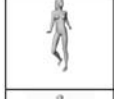
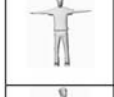
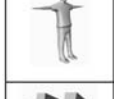
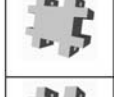

						
	0.000	0.003	0.021	0.019	0.286	0.241
	0.003	0.000	0.021	0.020	0.286	0.241
	0.021	0.021	0.000	0.010	0.286	0.241
	0.019	0.020	0.010	0.000	0.286	0.241
	0.286	0.286	0.286	0.286	0.000	0.079
	0.241	0.241	0.241	0.241	0.079	0.000

Table 1

Values for the matching distances between six different models in our database. The size graphs have been obtained using the integral geodesic distance f^3 and the region area.

invariance with respect to a transformation group simply means requiring the mapping and measuring functions to be invariant with respect to that group. Therefore, the added value of this approach to shape analysis relies on the possibility of adopting different functions as shape descriptors, according to the properties and invariants that one wishes to capture. When changing the functions, the resulting configurations can give insights on the shape from different perspectives. As a first example, if we aim to distinguish different positions in space, we should use mapping and measuring functions which emphasize the spatial distribution of the object shape, for example the distance from the barycenter, and the distances from the points on the bounding boxes. In Figure 12 (a) we see that, when searching for a standing-up human among the human models, their use produces a query response consisting of standing-up humans sharing a similar pose. Otherwise, if our idea is to put more emphasis, for example, on the “fatness” of the model rather than on its spatial pose, we can run the system selecting the region area as the measuring function. The results can be seen in Figure 12 (b), where the same query as before, a fat man, returns a fat man as the closer model, followed by a series of males. On the contrary, if we are interested in retrieving models showing a variety of poses, the integral geodesic distance reveals to be the best choice for the mapping function, since the centerline representation based on the integral geodesic distance does not distinguish, e.g., straight legs from legs with bent knees (see Figure 12 (c) where the measuring function is r_{av}). Finally, in Figure 12 (d) the search for a pig in the class of animals with respect to the topological

distance from curvature extrema and r_{max} produces a response that reflects both topological and geometric properties, according to perceptual similarity among the animals.

























Query	1 st match	2 nd match	3 rd match	4 th match	5 th match
(a) 					
(b) 					
(c) 					
(d) 					

Fig. 12. Query results according to different choices of the mapping and the measuring functions. (a) Emphasis on the spatial position, using f^1 and φ_{P_5, P_6} ; (b) emphasis on the fatness, with f^1 and the *area* of the regions; (c) humans in different poses are recognized, using f^3 and r_{av} ; (d) results on the class of four-limbs reflecting perceptual similarity, obtained by f^4 and r_{max} .

These results suggest that our approach could also be used as either a finer tool, after a rough filter has been used, or an instrument to refine queries in a retrieval pipeline, according to the user needs. The idea is that the interaction between the user and the system would allow tailoring the system response to the precise shape domain a user has in mind. This may be particularly true when considering our approach. While, in general, it may be difficult for the user to select the best combination of features to reach his goal, using our technique allows him to readily indicate the shape idea he has in mind, through the selection of a set of a features (i.e., mapping and measuring functions), which have a clear and intuitive geometric (and perceptual) significance.

7.4 Performance evaluation: a comparative study

In this section we use our database to compare the retrieval performance of the approach proposed, with respect to three popular shape descriptors, namely the spherical harmonic descriptor³ [21], the view-based approach⁴ [22] and the Multiresolution Reeb graph [26].

The results about size functions reported in this section are obtained selecting, during a training phase, the most performing combination of features (mapping functions f and measuring functions φ) to compute the size graph (G^f, φ) . The training phase has been performed using a smaller (20 items) database, containing a fraction of the models currently used as well as some additional models. The performance of different combinations has been evaluated in terms of the area of the region below the precision–recall curve, considering that larger areas indicate better results. In particular, the most performing pair (G^f, φ) is obtained when f is the integral geodesic distance, and φ the minimum radius r_{min} (see Section 4.2 and 4.3). Some comparative results are summarized in Figure 13. An intuitive explanation for these results is that the integral geodesic distance is suitable to deal with articulated objects, while the minimum radius of the surface regions provides an informative description of the local object shape, which is also stable to small perturbations. The reader is referred to Section 4.2, 4.3 and 7.3 for a more detailed analysis of the behavior of different descriptors, and to Section 8.2 for future research directions on how to automatically choice and combine the descriptors.

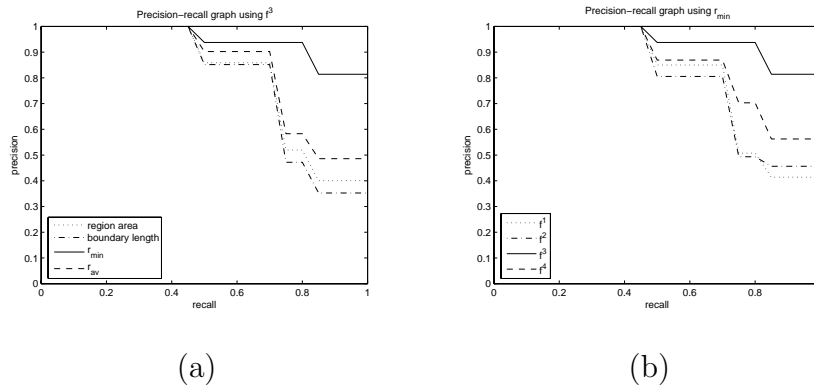


Fig. 13. Precision-recall diagrams on the training dataset, involving different skeletal representations and attributes. (a) The mapping function to extract the centerline is the normalized integral geodesic distance [26]; the measuring function varies in a set of four attributes. (b) The mapping functions varies, while the measuring function is always the minimum radius of a region.

³ <http://www.cs.jhu.edu/~misha/>

⁴ <http://3d.csie.ntu.edu.tw/~dynamic/3DRetrieval/index.html>

In what follows, we show some results on both the whole database and the single classes.

As a first performance parameter, we consider *percentage recall*. For a given number N , this parameter corresponds to the percentage of models in the same class of the query retrieved within the first N items. In particular, the recall histogram in Figure 14 are obtained computing percentage recall for the rank thresholds $N = 10, 20, \dots, 120$. Results are averaged over the whole database, and indicate that almost 80% of relevant items are retrieved within top 25% of the database (that is, within the first 30 models; remember that each class contains 20 elements). Moreover, in many cases the values obtained using size functions appears shifted up in the histogram, in comparison with respect to competitors, meaning a better performance.

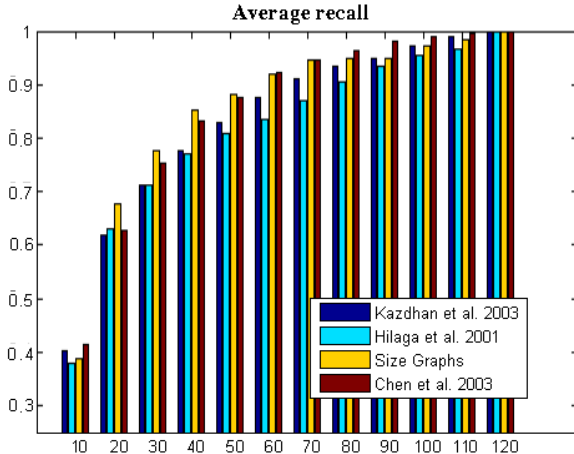


Fig. 14. Comparison with existing retrieval methods. Recall histograms: the values are averaged on the whole database.

Figure 15 (a) compares the *average rank* for the whole database obtained using size functions with the values obtained by the other techniques. The average rank is obtained by running each model of the database as a query and computing the retrieval rank of all members in the class of the query. The value obtained with size functions is the lowest one; notice that for this indicator lower values indicate better performance.

Another measure we use to assess the retrieval performance is the *last place ranking*, defined in [70] as

$$L_n = 1 - \frac{Rank_l - n}{N - n},$$

where $Rank_l$ indicates the rank at which the last relevant object is found, n is the number of relevant items and N is the size of the whole collection. The values computed are reported in Figure 15 (b). This measure gives an estimate of the number of items retrieved a user has to search in order to

have a reasonable expectation of finding all relevant items. The higher this measure within the interval $[0, 1]$, the lower the number of items to check, meaning better results.

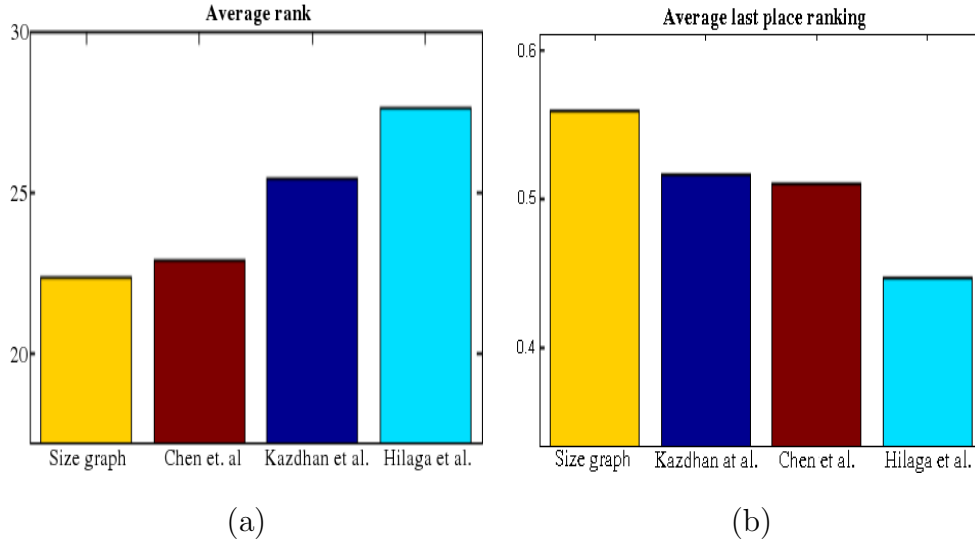


Fig. 15. Comparison with existing retrieval methods. (a) average rank and (c) last place ranking.

Finally, Figure 16 shows the standard precision–recall diagrams computed over the 5 classes in the database. The curves with different colors correspond to the different techniques we have tested. Remember that curves shifted upwards and to the right indicate a superior retrieval performance. For example, it is worth noticing the good performance realized by size functions for the classes of humans and glasses. These articulated models are well suited to enhance the features of our method, since the object in these classes share a common structure both for connectivity (e.g. a head, two arms and two legs for humans) and attributes (large sections near the barycenter, smaller ones for protrusions). The worst performance is realized when dealing with the class of animals with four limbs. The reason is that the models in this class are very heterogeneous. Our method takes into account, besides the structural properties – having four limbs – also the geometrical properties of the shape, which show too strong variations. In this particular case, better results could be obtained by considering a finer level of classification, rather than a basic one.

8 Concluding remarks

We have proposed an original framework to extend the use of size functions in the 3D context. More precisely, we have derived a signature to be extracted from 3D models, which guarantees the topological coding and the geometrical

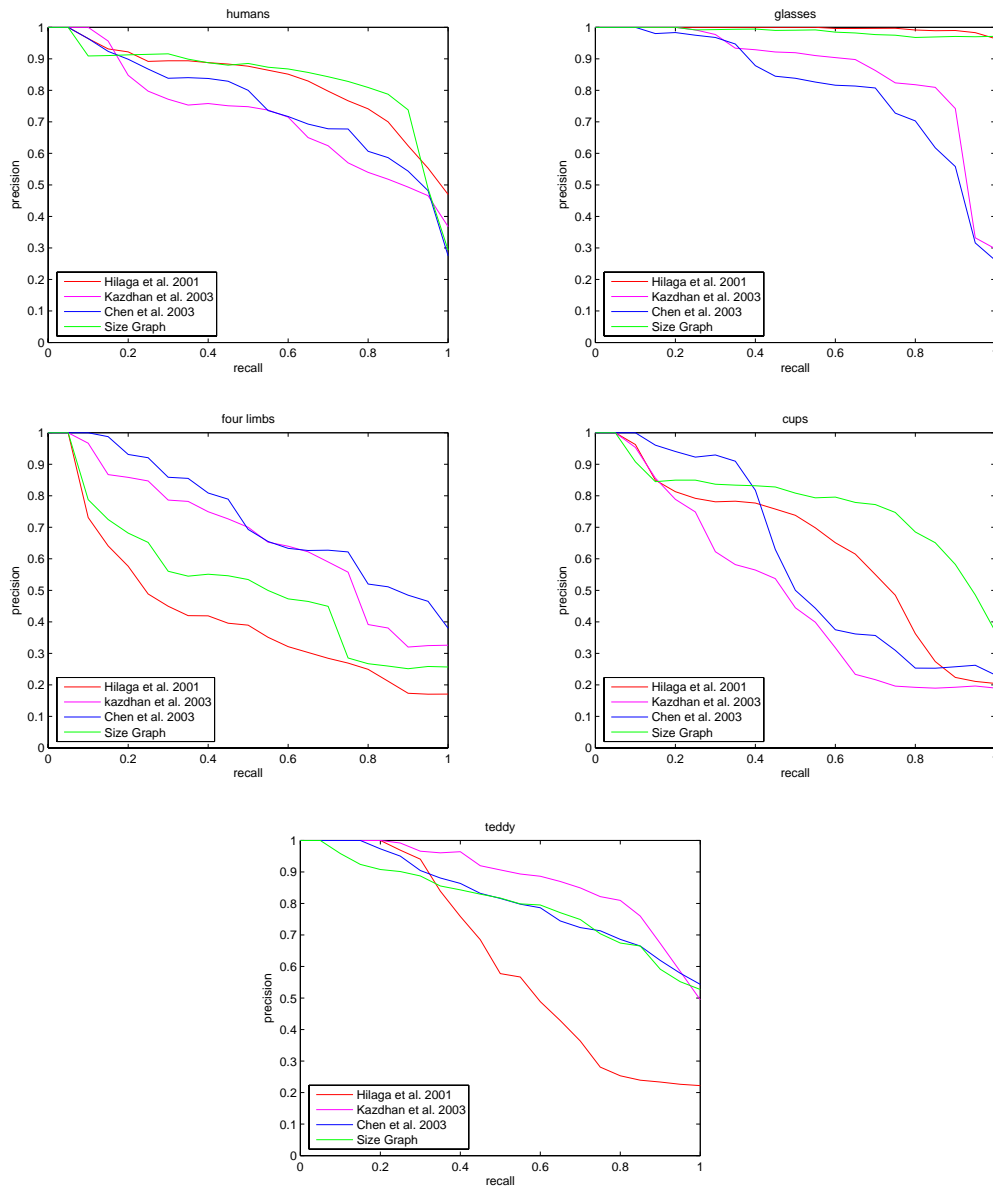


Fig. 16. Precision–recall diagrams for the different classes in our database.

description, and which is computationally efficient. This representation has been used as a size graph for computing size functions, making the use of size functions in the 3D domain efficient and effective, as shown in a series of experiments.

8.1 Properties

The proposed shape descriptor presents many desirable properties:

- (1) quick to compute: the computation of 120 size functions for the 120 mod-

els in the database requires 1.53 second on a 1.73GHz laptop PC-M; the off-line step of computing the size graphs requires 1 minute and 12 seconds;

- (2) concise to store: less than 1k storage per model (see details in Table 2);
- (3) easy and quick to compare: evaluating 120×120 matching distances between size functions requires 8.55 seconds;
- (4) invariant under similarity transformations: imposing the desired invariance simply means requiring the same invariance for the mapping and measuring functions, without any change in the mathematical and computational framework;
- (5) robust against noise and small extra features, as shown in Figure 11 and Table 1;
- (6) able to discriminate among shapes at many scales, conveying information about global and local properties of the shape, as shown by the experimental results.

Models	#vertices	#faces	Size	#N	#CP	SF
vase [52]	896,338	1,792,672	73MB	45	9	< 1k
Happy Buddha [71]	543,652	1,087,716	45MB	27	12	< 1k
armadillo [52]	165,951	331,898	15MB	41	12	< 1k
hand [52]	136,663	273,060	13MB	56	13	< 1k
Bunny [71]	32,872	65,740	3.1MB	18	4	< 1k
dancer [52]	24,998	49,996	2.2MB	22	6	< 1k
dancer2 [52]	26,358	52,716	2.6MB	20	6	< 1k

Table 2

Statistics for some models, relating the dimension of the original model with the number of nodes ($\#N$) of the size graph G^f , the number of cornerpoints ($\#CP$) of the corresponding size function and the storage size of the final descriptor (SF).

The added value of our approach relies on the fact that we provide a modular framework, based on the idea of describing shapes by geometrical-topological properties of real functions. A suite of descriptors is thus available, which can be fit to the problem at hand, and helps the user to tune the retrieval system to be on his/her wavelength.

As for the limitations of our approach, currently the method to derive the size graph can process only manifold meshes; therefore, in the current version, the method cannot be applied to polygon soups and point cloud models. However, the possibility of performing not only a geometric comparison but also a structural analysis largely compensates for this limitation.

In this sense, our framework exploits an abstract shape description, which complements existing geometry-oriented methods.

8.2 *Directions of future research*

We are currently researching into the development of new measuring functions, in order to analyze different kind of shape features of three-dimensional models. We believe that capturing a larger amount of information would increase the retrieval performance, allowing for a better discrimination of objects, and the rejection of some of the false matchings which can be observed. Moreover, we are also investigating how the choice of the mapping functions for the graph extraction determines the characteristics of the resulting graph configuration. The experimental results have shown that this approach is promising, and goes in the direction of developing tools to automatically annotate the shape semantic, and to encapsulate it in a digital shape representation.

A next step in our research project is the investigation into a method to automatically select the most performing descriptors, and to combine their information into a single dissimilarity score. A statistical framework has been recently proposed in [72], where a *number of false alarms* is computed to merge the contributions of different families of measuring functions, with applications to image retrieval.

The use of multi-variate measuring functions is also a current research theme [73].

Acknowledgements

The authors thank Prof. M. Ferri and the Vision Mathematics Group at the University of Bologna for fruitful discussions. Thanks are also due to G. Barequet and S. Har-Peled for providing us their code for minimal bounding-box extraction.

This work has been developed in the CNR research activity (ICT-P04) and partially supported by the European Network of Excellence “AIM@SHAPE” (contract number 506766), and the Italian National Project SHALOM, funded by the Italian Ministry of Research (contract number RBIN04HWR8).

References

- [1] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewics, D. Koller, L. Pereira, M. Gintzon, S. Anderson, J. Davis, J. Gensberg, J. Shade, D. Fulk, The Digital Michelangelo Project: 3D scanning of large statues, in: SIGGRAPH 2000: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, ACM Press, 2000, pp. 131–144.
- [2] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, P. E. Bourneet, The protein data bank, *Nucleic Acid Research* 28 (1997) 235–242.
- [3] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The Princeton Shape Benchmark, in: Proceedings of SMI 2004: International Conference on Shape Modeling and Applications, IEEE Computer Society, 2004, pp. 167–178.
- [4] <http://www.cs.uu.nl/centers/give/imaging/3Drecog/3Dmatching.html>.
- [5] I. Biederman, Recognition-by-components: A theory of human image understanding, *Psychological Review* 94 (1987) 115–147.
- [6] D. Marr, *Vision - A computational investigation into the human representation and processing of visual information: An invitation to Cognitive Science*, W. H. Freeman, 1982.
- [7] P. Frosini, A distance for similarity classes of submanifolds of a Euclidean space, *Bulletin of the Australian Mathematical Society* 42 (1990) 407–416.
- [8] A. Verri, C. Uras, P. Frosini, M. Ferri, On the use of size functions for shape analysis, *Biological Cybernetics* 70 (1993) 99–107.
- [9] P. Frosini, C. Landi, Size theory as a topological tool for Computer Vision, *Pattern Recognition and Image Analysis* 9 (1999) 596–603.
- [10] T. Kaczynski, K. Mischaikow, M. Mrozek, *Computational Homology*, Vol. 157 of Applied Mathematical Sciences, Springer-Verlag, 2004.
- [11] H. Edelsbrunner, D. Letscher, A. Zomorodian, Topological persistence and simplification, *Discrete Computational Geometry* 28 (2002) 511–533.
- [12] M. Allili, D. Corriveau, D. Ziou, Morse homology descriptor for shape characterization, in: ICPR2004: Proceedings of the 17th International Conference on Pattern Recognition, Vol. 4, 2004, pp. 27–30.
- [13] C. Uras, A. Verri, Computing size functions from edge maps, *International Journal of Computer Vision* 23 (2) (1997) 169–183.
- [14] A. Cerri, M. Ferri, D. Giorgi, Retrieval of trademark images by means of size functions, *Graphical Models* 68 (5) (2006) 451–471.

- [15] I. Stanganelli, A. Brucale, L. Calori, R. Gori, A. Lovato, S. Magi, B. Kopf, R. Bacchilega, V. Rapisarda, A. Testori, P. A. Ascierio, E. Simeone, M. Ferri, Computer-aided diagnosis of melanocytic lesions, *Anticancer Research* 25 (6C) (2005) 4577–4582.
- [16] S. Biasotti, D. Giorgi, M. Spagnuolo, B. Falcidieno, Size functions for 3D shape retrieval, in: *SGP 2006: Proceedings of the 2006 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, 2006, pp. 239–242.
- [17] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, D. V. Vranić, Feature-based similarity search in 3D object databases, *ACM Computing Surveys* 37 (4) (2005) 345–387.
- [18] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions, *ACM Transactions on Graphics* 21 (4) (2002) 807–832.
- [19] M. Novotni, R. Klein, A geometric approach to 3D object comparison, in: *Proceedings of SMI 2001: International Conference on Shape Modelling and Applications*, IEEE Computer Society, 2001, pp. 167–175.
- [20] J. Corney, H. Rea, D. Clark, J. Pritchard, M. Breaks, R. MacLeod, Coarse filters for shape matching, *Computer Graphics and Applications* 3 (22) (2002) 65–74.
- [21] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3D shape descriptors, in: *Proceedings of SGP 2003: Eurographics Symposium on Geometry Processing*, 2003, pp. 156–165.
- [22] D. Chen, M. Ouhyoung, X. Tian, Y. Shen, On visual similarity based 3D model retrieval, *Computer Graphics and Applications* (22) (2003) 223–232.
- [23] A. Del Bimbo, P. Pala, Content-based retrieval of 3D models., *ACM Transactions on Multimedia Computing, Communications and Applications* 2 (1) (2006) 20–43.
- [24] G. Collins, A. Zomorodian, A. Carlsson, L. J. Guibas, A barcode shape descriptor for curve point cloud data, *Computers and Graphics* 28 (2004) 881–894.
- [25] G. Carlsson, A. Zomorodian, A. Collins, L. J. Guibas, Persistence barcodes for shapes, *International Journal of Shape Modeling* 11 (2) (2005) 149–187.
- [26] M. Hilaga, Y. Shinagawa, T. Kohmura, T. L. Kunii, Topology matching for fully automatic similarity estimation of 3D shapes, in: *SIGGRAPH 2001: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press, 2001, pp. 203–212.
- [27] S. Biasotti, S. Marini, M. Mortara, G. Patané, M. Spagnuolo, B. Falcidieno, 3D shape matching through topological structures, *Lecture Notes in Computer Science* 2886 (2003) 194–203.
- [28] S. Biasotti, M. Marini, M. Spagnuolo, B. Falcidieno, Sub-part correspondence by structural descriptors of 3D shapes, *Computer Aided Design* 38 (9) (2006) 1002–1019.

- [29] E. Zuckerberger, A. Tal, S. Shlafman, Polyedral surface decomposition with applications, *Computers & Graphics* 26 (2002) 733–743.
- [30] N. D. Cornea, M. F. Demirci, D. Silver, A. Shokoufandeh, S. J. Dickinson, P. B. Kantor, 3D object retrieval using many-to-many matching of curve skeletons, in: *Proceedings of SMI 2005: International Conference on Shape Modeling and Applications*, IEEE Computer Society, 2005, pp. 368–373.
- [31] J. Zhang, K. Siddiqi, D. Macrini, A. Shokoufandeh, S. Dickinson, Retrieving articulated 3-D models using medial surfaces and their graph spectra, in: *Proceedings of EMMCVPR 2005: International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2005.
- [32] F. Demirci, A. Shokoufandeh, Y. Keselman, L. Bretzner, S. Dickinson, Object recognition as many-to-many feature matching, *International Journal of Computer Vision* 69 (2) (2006) 203–222.
- [33] H. Sundar, D. Silver, N. Gagvani, S. Dickinson, Skeleton based shape matching and retrieval, in: *Proceedings of SMI 2003: International Conference on Shape Modeling and Applications*, IEEE Computer Society, 2003, pp. 130–139.
- [34] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Shape-based searching for product lifecycle applications, *International Journal of Computer Vision* 37 (13) (2005) 1435–1446.
- [35] B. T. Messmer, H. Bunke, A new algorithm for error tolerant subgraph isomorphism detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (5) (1998) 493–504.
- [36] B. T. Messmer, H. Bunke, Matching free trees, maximal cliques, and monotone game dynamics, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (11) (2002) 1535–1541.
- [37] R. C. Wilson, E. R. Hancock, B. Luo, Pattern vectors from algebraic graph theory, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (7) (2005) 1112–1124.
- [38] J. W. H. Tangelder, R. C. Veltkamp, A survey of content based 3D shape retrieval methods, in: *Proceedings of SMI 2003: International Conference on Shape Modeling and Applications*, IEEE Computer Society, 2004, pp. 145–156.
- [39] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Three-dimensional shape searching: state-of-the-art review and future trends, *Computer-Aided Design* 37 (5) (2005) 509–530.
- [40] M. d’Amico, P. Frosini, C. Landi, Using matching distance in size theory: A survey, *International Journal of Imaging Systems and Technology* 16 (5) (2006) 154–161.
- [41] P. Frosini, C. Landi, New pseudo-distances for the size function space, in: R. A. Melter, A. Y. Wu, L. J. Latecki (Eds.), *Vision Geometry VI, Proceedings of SPIE*, Vol. 3168, 1997, pp. 52–60.

- [42] P. Frosini, C. Landi, Size functions and formal series, *Applicable Algebra in Engineering, Communication and Computing* 12 (2001) 327–349.
- [43] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, Stability of persistence diagrams, *Discrete and Computational Geometry* 37 (1) (2007) 103–120.
- [44] M. d’Amico, P. Frosini, C. Landi, Optimal matching between reduced size functions, *Tech. Rep. 35*, DISMI, Univ. di Modena e Reggio Emilia (2003).
- [45] M. d’Amico, P. Frosini, C. Landi, Natural pseudo-distance and optimal matching between reduced size functions, *Tech. Rep. 66*, DISMI, Univ. di Modena e Reggio Emilia (2005).
- [46] P. Frosini, Discrete computation of size functions, *Journal of Combinatorics and Information System Science* 17 (1992) 232–250.
- [47] M. d’Amico, A new optimal algorithm for computing size function of shapes, in: *CVPRIP Algorithms III: Proceedings International Conference on Computer Vision, Pattern Recognition and Image Processing*, 2000, pp. 107–110.
- [48] F. Lazarus, A. Verroust, Level set diagrams of polyhedral objects, in: W. Bronsvoort, D. Anderson (Eds.), *SMA 1999: Proceedings of the 5th ACM Symposium on Solid Modeling and Applications 1999*, ACM Press, 1999, pp. 130–140.
- [49] A. Gramain, *Topologie des surfaces*, Presses Universitaires de France, 1971.
- [50] S. Biasotti, *Computational topology methods for shape modelling applications*, Ph.D. thesis, Università degli Studi di Genova (May 2004).
- [51] S. Biasotti, B. Falcidieno, M. Spagnuolo, *Topological Data Structures for Surfaces: An Introduction for Geographical Information Science*, John Wiley & Sons, 2004, Ch. Surface Shape Understanding based on Extended Reeb Graphs, pp. 87–103.
- [52] <http://shapes.aim-at-shape.net>.
- [53] M. Mortara, G. Patané, Shape-covering for skeleton extraction, *International Journal of Shape Modelling* 8 (2) (2002) 245–252.
- [54] T. H. Cormen, C. E. Leiserson, R. L. Rivest, *Introduction to Algorithms*, The MIT Press, Cambridge, MA, 1994.
- [55] G. Barequet, S. Har-Peled, Efficiently approximating the minimum-volume bounding box of a point set in three dimensions, *Journal of Algorithms* 38 (2001) 91–109.
- [56] R. E. Tarjan, J. von Leeuwen, Worst-case analysis of set union algorithms., *Journal of the Association for Computing Machinery* 31(2) (1984) 245–281.
- [57] W. Ackermann, Zum hilbertschen aufbau der reellen zahlen, *Mathematische Annalen* 99 (1928) 118–133.

- [58] C. L. P. Donatini, P. Frosini, Deformation energy for size functions, in: E. R. Hancock, M. Pelillo (Eds.), *Proceedings of EMMCVPR 1999: International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 1999, pp. 44–53.
- [59] P. Donatini, P. Frosini, Lower bounds for natural pseudodistances via size functions, *Archives of Inequalities and Applications* 2 (2004) 1–12.
- [60] S. Biasotti, L. De Floriani, B. Falcidieno, P. Frosini, D. Giorgi, C. Landi, L. Papaleo, M. Spagnuolo, Geometrical-topological properties of real functions for describing shapes, Tech. Rep. 5, IMATI (2006).
- [61] K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, V. Pascucci, Loops in reeb graphs of 2-manifolds, *Discrete and Computational Geometry* 32 (2004) 231–244.
- [62] T. Tung, F. Schmitt, The Augmented Multiresolution Reeb Graph approach for content-based retrieval of 3D shapes, *International Journal of Shape Modelling* 11 (1) (2005) 91–120.
- [63] A. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-Based Image Retrieval at the end of the early years., *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22 (12) (2000) 1349–1380.
- [64] C. J. V. Rijsbergen, *Information Retrieval*, Butterworth, 1979.
- [65] G. Salton, *Introduction to modern information retrieval*, McGraw, 1983.
- [66] D. P. Huijsmans, N. Sebe, How to complete performance graphs in Content-Based Image Retrieval: Add generality and normalize scope, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005) 245–251.
- [67] <http://www.designrepository.org>.
- [68] <http://www.hec.afrl.af.mil/HECP/Card1b.shtml> #caesarsamples.
- [69] <http://www.cim.mcgill.ca/shape/benchMark/>.
- [70] J. Eakins, J. Boardman, M. Graham, Similarity retrieval of trademark images, *Multimedia* 5 (2) (1998) 53–63.
- [71] <http://graphics.stanford.edu/data/3Dscanrep/>.
- [72] A. Cerri, D. Giorgi, P. Musé, F. Sur, F. Tomassini, Shape recognition via an a contrario model for size functions, in: *Proceedings of ICIAR 2006: International Conference on Image Analysis and Recognition*, 2006.
- [73] S. Biasotti, A. Cerri, P. Frosini, D. Giorgi, C. Landi, Multidimensional size functions for shape comparison., Tech. Rep. 4, IMATI (2007).

Silvia Biasotti is contract researcher at the Institute of Applied Mathematics and Information Technology of the National Research Council in Genova (Italy). She graduated in Mathematics in 1998 and obtained a doctoral degree in Mathematics and Applications in 2004, at the University of Genoa. Since 1998, she has been on the staff of the Shape Modelling Group (ex Computer Graphics Group) at the Department of Genova of the Institute of Applied Mathematics and Information Technology. Her research interests include computational topology, shape abstraction and skeleton representation of polyhedral surface.

Daniela Giorgi is a post-doctoral fellow at the Institute of Applied Mathematics and Information Technology of the National Research Council in Genova (Italy). She received a degree cum laude in Mathematics from the University of Bologna (Italy), in 2002, and a Phd degree in Applied Mathematics at the University of Padua, in 2006. Her research interests are in Pattern Recognition and topological methods for shape analysis and comparison.

Michela Spagnuolo got the Laurea Degree cum laude from the Dept. of Mathematics, University of Genova, 1989, and a Doctoral Degree in Computer Science Engineering, at the INSA, Lyon, France, 1997. She has studied techniques of geometric reasoning for the extraction of shape features from discrete surface models and geometric models for coding uncertainty in data samples (fuzzy-based modelling). She is currently senior researcher at IMATI-CNR Genova, and her research interests are related to shape-based approaches to modeling digital shapes and computational topology techniques for shape analysis.

Bianca Falcidieno is a Research Director of the CNR and the chair of the department of Genova of the Institute of Applied Mathematics and Information Technology. For many years she has been responsible for the Geometric Modelling and Computer Graphics Research Group, now Shape Modelling Group, working in the field of Applied Mathematics and Computer Science, with applications in Computer Graphics, Geographic Information Systems, and Industrial Design. In her activity, she supervised many young researchers and PhD students in Italy and abroad, in these topics. Editor in chief of the International Journal Shape Modelling (a World Scientific journal), she is the author of more than 200 scientific refereed papers and books. Bianca Falcidieno is the coordinator of the FP6 Network of Excellence AIM@SHAPE.