

# 3D Shape Matching through Topological Structures

Silvia Biasotti, Simone Marini, Michela Mortara,  
Giuseppe Patanè, Michela Spagnuolo, and Bianca Falcidieno

Istituto di Matematica Applicata e Tecnologie Informatiche  
Consiglio Nazionale delle Ricerche  
{silvia,simone,michela,patane,spagnuolo,falcidieno}@ge.imati.cnr.it  
<http://www.ge.imati.cnr.it>

**Abstract.** This paper introduces a framework for the matching of 3D shapes represented by topological graphs. The method proposes as comparison algorithm an error tolerant graph isomorphism that includes a structured process for identifying matched areas on the input objects. Finally, we provide a series of experiments showing its capability to automatically compare complex objects starting from different skeletal representations used in Shape Modeling.

## 1 Introduction

Work on shape representation and comparison is based on a trade-off between conciseness and expressiveness of the chosen scheme. Shape representations have to keep in a compact and effective way the object topology and geometry; for instance, shocks graphs [7], component/max trees [9,19] in 2D imagining, and topological graphs of 3D meshes [2] and volumes [10] are popular tools for the abstraction of complex information. As shape descriptor we consider a topological representation that codes the relations among the surface features in a graph and serves as input for the matching algorithm. The matching method proposed in the paper is based on an error tolerant graph isomorphism which identifies a greedy approximation of the *maximal common subgraph* shared by the two input graphs and that gives information about the analogies and differences among the features of the compared objects. The paper is organized as follows: the graph matching algorithm for attributed directed acyclic graphs is introduced in section 2, where a possible similarity measure is also proposed. The definition of a particular class of topological graphs, i.e. the Reeb graphs, is outlined in section 3, while a series of experiments for comparing complex objects and conclusions are given in section 4.

## 2 Matching of Topological Structures

The problem of comparing topological structures has been approached in several ways. In [6], multi-resolution Reeb graphs are extracted and compared in order

to estimate the similarity between 3D models; the graph matching algorithm is based on a coarse to fine strategy that preserves the consistency of the graph structure. In [21] the shock graph is used as signature of the shape of 2D objects. In order to perform the graph matching the shock graphs are transformed in rooted tree and then compared. Analogously, in [10] a skeletal graph is computed from a volumetric object and the corresponding graphs are compared using the methodology presented in [21].

In this paper we propose an algorithm to compare objects described by topological structures, where the input shape is represented by an attributed, directed and acyclic graph as formalized in definition 1.

**Definition 1.** A *M-graph*  $G$  is given by a quadruple  $G = (V, E, \mu_V, \mu_E)$ , where  $V$  is a set of nodes,  $E \subseteq V \times V$  is the set of the graph edges,  $\mu_V : V \rightarrow A_V$  and  $\mu_E : E \rightarrow A_E$  are the node and the edge attribute functions, with  $A_V, A_E$  sets of node and edge attributes of  $G$ . The set of M-graphs is denoted by  $M_{Gset}$ . A *subgraph*  $S$  of  $G$  is a quadruple  $(V_S, E_S, \mu_{V_S}, \mu_{E_S})$ , where  $V_S \subseteq V, E_S \subseteq E \cap (V_S \times V_S)$ ,  $\mu_{V_S}$  and  $\mu_{E_S}$  are induced by  $\mu_V$  and  $\mu_E$ , respectively.

Since the graph is directed, each node  $v \in V$  identifies a subgraph  $S$  of  $G$ , where  $V_S$  contains the  $v$  itself and all the nodes for which  $v$  is an ancestor. This property is used during the graph comparison process in order to match not only nodes, but subgraphs too.

The isomorphism notion defines an equivalence relationship among M-graphs, [15]. Since the demand for two graphs to be isomorphic is a strong condition, generally unsuitable for similarity tasks, we relax this hypothesis and propose a weaker notion of isomorphism. Starting from the properties of weak isomorphism presented in [15], the definition of an error tolerant graph isomorphism is introduced as a set of graph editing operations that makes the two M-graphs isomorphic.

**Definition 2.** Let  $G$  and  $G'$  be two M-graphs and  $\Delta = (\delta_1, \dots, \delta_n)$  a sequence of graph editing operations (where a graph edit operation,  $\delta_i$ , is an addition, a deletion or an attribute modification of nodes and edges), then:

- the **edited graph**  $\Delta(G)$  is the graph  $\Delta(G) = \delta_n(\delta_{n-1}(\dots(\delta_1(G))\dots))$ ;
- an **error tolerant graph isomorphism** is a couple  $\psi = (\Delta, \psi_\Delta)$ , where  $\Delta$  is a sequence of editing operations such that there exists an isomorphism  $\psi_\Delta$  between  $\Delta(G)$  and  $G'$ .

Given two M-graphs (named *input* and *model graph*, respectively), we construct an error tolerant isomorphism originating a subgraph (possibly not connected) in the input graph that is “mapped” onto an isomorphic subgraph (with respect to the graph edit operations) of the model graph. The error tolerant isomorphism is seen as a transformation process from the input graph to the model graph where the track of the editing operations explicits the differences between the two objects and is used to check the effectiveness of the matching process.

## 2.1 Error Tolerant Graph Isomorphism

The construction of a graph isomorphism is intrinsically a NP-problem and, in the scientific literature, most of methods compare graphs by exploiting statistic [12], genetic algorithms [8], fuzzy approaches [13], optimization theory [4] or a search in the space of states [15].

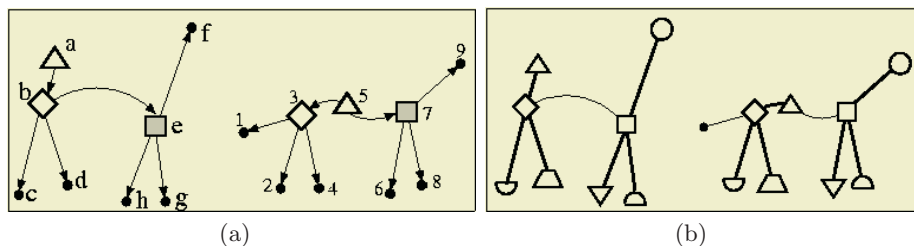
In this paper a heuristic algorithm has been devised to assess and quantify an error tolerant graph isomorphism. Let  $S$  be the subgraph identified by  $v \in V$ ,  $v$  is described by the vector  $s_v = \langle (in(v), out(v), in\_sum, out\_sum, n, s\_sum) \rangle$ , where  $in()$  (resp.  $out()$ ) is the indegree (resp. outdegree) of  $v$ ,  $in\_sum$  (resp.  $out\_sum$ ) the indegree (resp. outdegree) node sum of  $S$ ,  $n$  the cardinality of  $V_S$  and  $s\_sum$  the sum of the attributes of  $E_S$ . The proposed algorithm is a search in the *state spaces* defined by the input graph  $G_1 = (V_1, E_1, \mu_{V_1}, \mu_{E_1})$  and the model graph  $G_2 = (V_2, E_2, \mu_{V_2}, \mu_{E_2})$ , where a *state* is the ordered set of node pairs  $St = \{(v_j, v_k) \mid v_j \in V_1, v_k \in V_2, (v_j, v_k) \text{ is candidate to be mapped by } \psi_\Delta\}$ . The state ordering is based on a *cost function* associated to each pair of nodes and computed by the real-valued function  $C(v_j, v_k)$ , which is the Euclidean distance between the vectors  $s_{v_j}, s_{v_k}$  and works as a node similarity measure. Therefore, if  $\mu_{V_i}, \mu_{E_i}$  are the attribute functions of  $G_i, i = 1, 2$ , we require that there exist two totally ordered sets  $A_V$  and  $A_E$ , such that  $A_{V_i} \subseteq A_V$ , and  $A_{E_i} \subseteq A_E$ .

The algorithm maps nodes and edges of  $G_1$  and  $G_2$  moving from an *initial state*  $St_0$  to a *final state*  $St_f$ , where  $St_f = \emptyset$ , through a sequence of intermediate states  $St_i$ .

**The initial state.** The initial state  $St_0$  is set using a heuristics based on the information associated to the nodes. The nodes of  $G_i, i = 1, 2$  are ordered with respect to  $\mu_{V_i}$  and a set of candidate node pairs is chosen as follows. Firstly, the pairs having similar attribute value are selected. Then, the set of candidate pairs is refined by considering only those with relevant nodes: since each node generates a subgraph  $S$ , the bigger  $S$  is, the more the node is relevant. Finally, the set of candidates is ordered with respect to the cost function  $C$ . In figure 1 an example is shown on two directed acyclic M-graphs: the initial node candidates, selected with respect to the node attributes and the node relevance, are marked with the same symbols (see figure 1(a)). Leaves are not considered because the associated information is negligible.

**Choice of the best candidate node pair.** The heuristic used to choose the best candidate pair  $(v_j, v_k)$  involves both the cost of the node pair  $C(v_j, v_k)$  and the information carried by the two nodes. The relevance related to a candidate pair is given by the minimum between the cardinality of the subgraphs induced by the two nodes. Therefore, the chosen candidate is the node pair of minimum cost, with respect to  $C$  which has a relevance greater or equal to the average relevance value in the state. This choice generates a priority list for the construction of the isomorphism node mapping. In figure 1(a), the first best candidate pair is denoted by a grey square.

**From the state  $St_i$  to the state  $St_{i+1}$ .** The candidate node pair  $(v_j, v_k)$  having highest priority is removed from  $St_i$  and becomes a component of the node mapping isomorphism  $\psi_\Delta$ . New candidate pairs are obtained from  $v_j$  and



**Fig. 1.** Example of the matching method: initial state and best candidate pair (a) and final matching (b).

$v_k$ : they are all the possible node pairs obtained by combining the child nodes of  $v_j$  and  $v_k$ . Then, the state  $i + 1$  is the state  $i$  enriched by the new candidate node pairs obtained from  $v_j$  and  $v_k$ . According to figure 1(a), when the node pair  $(e, 7)$  is added to  $\psi_\Delta$ , the new state is obtained from the previous one adding the new node pairs  $(h, 6)$ ,  $(h, 8)$ ,  $(h, 9)$ ,  $(g, 6)$ ,  $(g, 8)$ ,  $(g, 9)$ ,  $(f, 6)$ ,  $(f, 8)$ ,  $(f, 9)$  and removing  $(e, 7)$ .

**End of the search.** When the set of candidate pairs is empty the final state is reached. The node mapping that identifies the common subgraph (and consequently the edge mapping) is complete and the set of graph editing operations  $\Delta$  (see definition 2) is computed. Results are shown in figure 1(b): node mapping is highlighted with the same symbols; bold arcs represent the edge matching induced.

According to the notation proposed in [3], a graph  $S_M$  is a common subgraph of  $G_1$  and  $G_2$  if there exists a subgraph isomorphism between  $S_M$  and  $G_1$ ,  $S_M$  and  $G_2$ . Moreover,  $G_1$  can be transformed into  $G_2$  by adding to  $\Delta$  a *deletion* operation for each edge in  $E_1$  not belonging to  $S_M$  and an *addition* operation is inserted into  $\Delta$  for each edge in  $E_2$  not belonging to  $S_M$ . Furthermore, *modify* operations are added to  $\Delta$  for each couple of edges belonging to the edge mapping, that do not have the same attributes.

## 2.2 Similarity Measure

To compare two M-graphs  $G_1$  and  $G_2$ , a *distance metric*  $d : M_{Gset} \times M_{Gset} \rightarrow [0, 1]$  that satisfies the properties of uniqueness, non-negativity, identity, symmetry and triangular inequality, is necessary, [22]. Then, the *similarity measure*,  $s(G_1, G_2)$  between  $G_1$  and  $G_2$  is defined by  $s(G_1, G_2) = 1 - d(G_1, G_2)$ .

The bigger the common subgraph  $S_M$  of  $G_1$  and  $G_2$  defined by the error tolerant isomorphism is, the bigger the similarity measure  $s(G_1, G_2)$  should be. A possible choice [3] is  $d_B(G_1, G_2) = 1 - \frac{|maxcs(G_1, G_2)|}{\max(|G_1|, |G_2|)}$ , where  $maxcs(G_1, G_2)$  is the maximal common subgraph of  $G_1$  and  $G_2$  and  $|\dots|$  is the number of nodes of a graph. As required by our statements such a distance depends on the size of the subgraph; however, it does not take into account the attribute value of edges and nodes. This implies that each node of the subgraph has the same weight, despite its relevance in the graph. Our aim is to compute

the distance on the edges of  $S_M$  (where an arc  $e$  belongs to  $S_M$  iff the nodes it connects do) and to adapt  $d_B$  to our purposes, correcting the contribution of each edge in  $S_M$  by considering the difference between the attributes of the corresponding edge of  $G_1$  and  $G_2$ . More formally, our distance measure is defined as:  $d(G_1, G_2) = 1 - \frac{\sum_{e \in S_M} (1 - \frac{|\mu_{E(a)} - \mu_{E(b)}|}{\alpha})}{\max(|G_1|, |G_2|)}$ , where  $\alpha$  represents the maximal attribute value on the edges of the graphs in the database and  $e = \psi_\Delta(a) = \psi_\Delta(b)$ ,  $a \in E_1$ ,  $b \in E_2$ . Clearly, if the edges of  $G_1$  mapped in  $S_M$  have the same attribute value of those ones in  $G_2$ , our metric corresponds to  $d_B$ . Furthermore the non-negativity, uniqueness, identify and symmetry properties immediately follow from the definition; if  $S_M$  is the maximal common subgraph, also the triangular inequality can be demonstrated, through a procedure similar to that proposed in [3], and  $d$  is a distance.

### 2.3 Computational Complexity

The isomorphism construction involves the generation of the node description represented by the vector  $s_v$  previously described, the computation of the initial state, the transition among states and the choice of a candidate. Denoting with  $N$  the number of nodes of a graph, the generation of the vector  $s_v$  has a linear cost in  $N$ : each node or edge is read exactly once. The ordering of nodes with respect to  $\mu_V$  requires  $O(N^2)$  operations, while the generation of the initial candidates is quadratic. Also the transition between two states is  $O(N^2)$ . The removal of the chosen candidate is performed in linear time, but the generation of the new candidates has a quadratic cost. The choice of the best candidate from the current state is linear in the number of the candidates but, in the worst case, the number of candidates is quadratic in the number of nodes.

## 3 Topological Structures

Among the possible shape descriptors of 3D objects that may be coded as a M-graph, we detail in this section a discrete representation of the Reeb graph [18] discussing its application to shape retrieval. Let  $f : M \rightarrow \mathbb{R}$  be a real mapping function defined on a surface  $M$ , and let  $[f_{min}, f_{max}]$  be the variation interval of  $f$  on the surface  $M$ , and  $f_{min} < f_1 < \dots < f_h < f_{max}$  be the distribution of the values of the contour levels of  $M$ , which are supposed to be all non degenerate contours. In addition, let  $I = \{(f_{min}, f_1), (f_i, f_{i+1}), (f_h, f_{max}) \mid i = 1, \dots, h-1\} \cup \{f_{min}, f_1, \dots, f_h, f_{max}\}$  be the partition of the interval  $[f_{min}, f_{max}]$  provided by the set of the  $h + 1$  interior parts and the function values of the contour levels.

**Definition 3.** *An extended Reeb equivalence between two points  $P, Q \in M$  is given by the following conditions:*

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

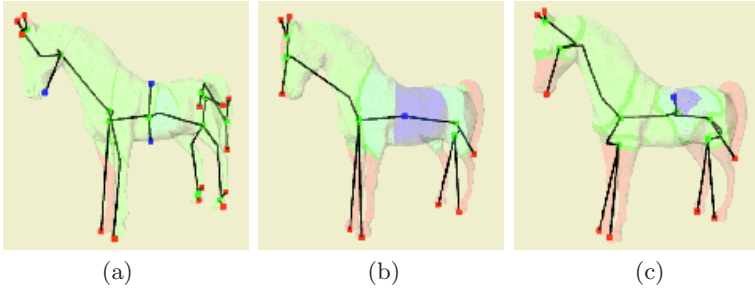
By the quotient relation in definition 3, all the points belonging to  $R := f^{-1}(f(t))$ , for some  $t \in I$ , are Reeb-equivalent in the extended sense and they are collapsed into the same point of the quotient space, called *Extended Reeb (ER) quotient space*. Moreover the *ER space*, which is an abstract sub-space of  $M$  and is independent of the geometry, may be represented as a traditional graph which is called the *Extended Reeb Graph, (ERG)*. The *ERG* with respect to the function  $f$  is a M-graph; in fact, the nodes correspond to critical points of  $f$ , i.e. the points of  $M$  where the gradient of  $f$  vanishes, while the node attributes are provided by their classification (as minima, maxima or saddles) and some geometric information such as the space position, the value of  $f$ , etc.. In addition, the function  $f$  induces a natural orientation on an edge  $e = (v_1, v_2)$  by considering as relation  $f(v_1) < f(v_2)$ , while the edge attributes are defined as  $\mu_E(e) = f(v_2) - f(v_1)$ . However, additional geometric information, such as cross-section length, area and volume of the portion of the corresponding part of the surface, may be stored for each edge. Finally, due to the monotonicity of  $f$  along an edge, the *ERG* is acyclic.

Likewise the Reeb graph [18,16], under the hypothesis  $I$  is a sufficiently dense partition of the domain of  $f$  [1], the *ERG* representation identifies the main topological properties of  $M$  independently of the chosen  $f$ . However, the application domain restricts the choice of  $f$ ; for instance, a suitable mapping function  $f$  has to be independent of rotation, translation, uniform scaling of the object and user's choices. These requirements prevent the use for matching purposes of the height function [1], and the centerline representation [11,5] which respectively depend on the orientation and on the selection of a seed point. The family of continuous or Morse functions is a natural set for identifying  $f$ , and in the following we present an overview of possible choices of  $f$  for coding triangular meshes without boundary.

**ERG with respect to the Euclidean distance from a point.** Differential topology and Morse theory guarantees that the distance functions of the surface points from a given point  $p$  of the Euclidean space are appropriate for extracting a Reeb graph. Such a point could belong to the mesh or not, even though a reasonable choice seems to be the barycenter of the object [2] which is easily calculated and, due to its linear dependence on all the vertices, is stable to small perturbations. In figure 2(a), an example of this graph representation is given.

**ERG with respect to the integral geodesic distance.** A different mapping function has been defined by Hilaga et al. [6], where the notion of integral geodesic distance has been introduced for matching purposes. In particular, for each vertex of a mesh  $M$ , the value of the function  $f$  is given by:  $f(v) = \sum_i g(v, b_i) \cdot area(b_i)$ , where  $g(v, b_i)$  represents the geodesic distance between  $v$  and  $b_i$ ,  $\{b_i\}_i$  are the base vertices for the Dijkstra's algorithm that are scattered almost equally on the surface, and  $area(b_i)$  is the area of the neighborhood of  $b_i$  (see figure 2(b)).

**Geodesic distance from curvature extrema.** The strategy proposed in [17] extracts the M-graph of a surface represented by a simplicial complex. More precisely, once computed a multi-resolutive Gaussian curvature on the input



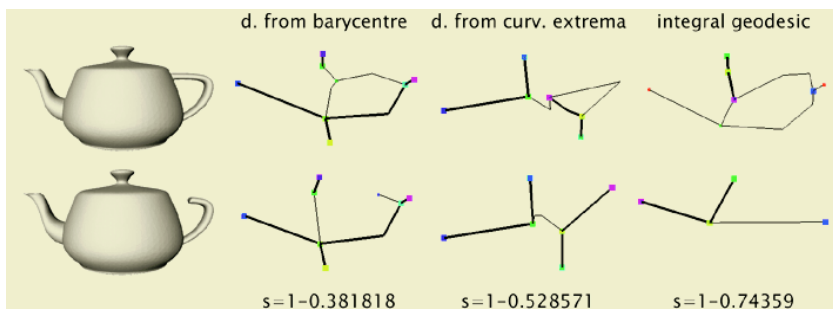
**Fig. 2.** The Reeb graph with respect to the distance from the barycenter (a), the integral geodesic distance (b), and the curvature extrema distance (c). The blue, red and green nodes represent minima, maxima and saddles, respectively.

mesh [14], for each high curvature region  $R_i$ ,  $i = 1, \dots, n$ , a seed vertex  $p_i$  is selected. Starting at the same time from all the representative vertices, rings made of vertices of increasing neighborhoods are computed in parallel until the whole surface is covered. Rings growing from different seed points collide and join where two distinct protrusions depart, thus identifying a branching zone. A graph is drawn according to the ring expansion: terminal nodes are identified by the seed points, while union or split of topological rings give branching nodes. Arcs join consecutive nodes of the graph, (see figure 2(c)). Experimental results have shown that this framework works on shapes of arbitrary genus.

## 4 Experimental Results and Concluding Remarks

The behavior of the different *ERG* representations has to be taken into account during the similarity analysis: in fact each function emphasizes different aspects of the object shape. For instance, the geodesic distance distribution on a human model does not change if the legs and the arms are stretched rather than curled up, since the geodesic distance from the body does not change, while the Euclidean distance from the center of mass does. Therefore, if the aim is to distinguish between different poses of the same object, the *ERG* with respect to the distance from the barycenter should be preferred.

Figure 3 highlights how the choice of  $f$  influences the matching results. In fact, a teapot has been slightly modified and the graphs result much different. The graph obtained by distance from the barycentre function 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 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 the modification of the teapot handle results in a new curvature extreme generating a new maximum critical point.



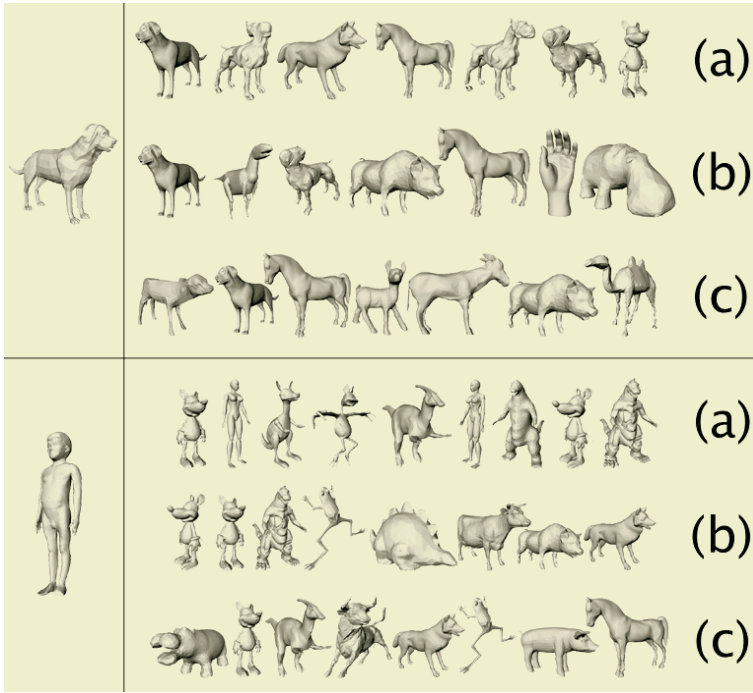
**Fig. 3.** Matching between the teapot and its modified version for the three *ERG* structures and corresponding similarity values.

In our experimental results the *ERG* structure, obtained from a uniform partition  $I$  of the interval  $[f_{min}, f_{max}]$ , highlights that the main structure of the object is better detected through a rough subdivision of it, while smaller features are located when the number of sub-intervals of  $I$  increases. On the contrary, the representation provided by the geodesic distance from curvature extrema depends on the choice of the base points, which are individuated using the multi-resolution strategy proposed in [14]. Finally, we observe the function  $f$  in our *ERG* representation is always non-negative. Then, to compare the shape features according to their relevance on the model, we adopt the following “normalized” *ERG* extraction: for each model the partition  $I$  is given by  $I = I_P \cap [f_{min}, f_{max}]$ , where  $I_P = \{(\frac{i*f_{max}}{m}, (\frac{(i+1)*f_{max}}{m}) \mid i = 0 \dots m - 1\} \cup \{\frac{i*f_{max}}{m} \mid i = 0 \dots m\}$  is a partition of the interval  $[0, f_{max}]$  and  $m$  is an integer chosen by the user.

Experimental results of our matching method are shown in figure 4, where the top objects retrieved by our matching algorithm on 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. For both, all the proposed *ERG* representations were 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.

We can conclude that the graph comparison reflects the intuitive notion of similarity and groups the objects in a number 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 entities and of configuration: in this case the query graph itself is easily contained into other graph representations of the models in our database. The framework for graph matching proposed in this paper is valid not only for Extended Reeb graphs but also for each graph-like representation which can be related to a M-graph. In particular, for the digital context, the component tree [9], the max-tree [19] and the topological graph proposed in [20], seem to be natural candidates. Moreover, even if the adopted matching approach is mainly based on the topological information





**Fig. 4.** Matching results for two query models in our database with respect to the three *ERG* representations: (a) distance from the barycentre, (b) integral geodesic distance and (c) distance from curvature extrema.

stored in the M-graph, as a future development we are planning to consider a greater number of geometric attributes which should improve the results so far obtained. Further improvements of the matching algorithm can be also obtained by considering the sequence of the editing operations of the input and model graphs and using them for partial mapping or metamorphosis purposes.

### Acknowledgements

This work has been partially supported by the National Project “MACROGeo: Metodi Algoritmici e Computazionali per la Rappresentazione di Oggetti Geometrici”, FIRB grant.

### References

1. M. Attene, S. Biasotti and M. Spagnuolo. Shape understanding by contour driven retiling. *The Visual Computer*, **19**(2-3):128–137, 2003.
2. S. Biasotti, S. Marini, M. Mortara and G. Patané. An overview on properties and efficacy of topological skeletons in shape modelling. *Proc. of Shape Modelling and Applications 2003*, IEEE Press, Seoul, pp. 245–254, 2003.

3. H. Bunke and K. Shearer A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, **19**:255–259, 1998.
4. S. Gold and A. Rangarajan. A Graduated Assignment Algorithm for Graph Matching. *IEEE Trans. on Patt. Anal. Mach. Intell.*, **18**(4):377–388, April 1996.
5. F. Hétroy and D. Attali. Topological Quadrangulations of Closed Triangulated Surfaces using the Reeb Graph. *Graphical Models*, **65**, pp. 131–148, 2003.
6. M. Hilaga, Y. Shinagawa, T. Komura and T. L. Kunii. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. *ACM Computer Graphics, (Proc. of SIGGRAPH 2001)*, Los Angeles, pp. 203–212, 2001.
7. B. Kimia, A. Tannenbaum and S. Zucker. Shapes, shocks, and deformations, I. *Computer Vision* 15:189–224, 1995
8. K. G. Koo and P. N. Suganthan. Multiple Relational Graphs Mapping Using Genetic Algorithms. *Proc. of Congr. on Evolutionary Comp.*, pp. 727–733, 2001.
9. R. Jones. Connected Filtering and Segmentation Using Component Trees. *Computer Vision and Image Understanding*, **75**(3):215–228, 1999
10. H. Sundar, D. Silver, N. Gagvani and S. Dickinson. Skeleton Based Shape Matching and retrieval. *Proc. of Shape Modelling and Applications 2003*, IEEE Press, Seoul, pp. 130–139, 2003.
11. F. Lazarus and A. Verroust. Level Set Diagrams of Polyhedral Objects. *ACM Solid Modeling '99*, Ann Arbor, Michigan, pp. 130–140, 1999.
12. B. Luo and E. R. Hancock. Symbolic Graph Matching using the EM Algorithm and Singular Value Decomposition. *Proc. of Int. Conf. on Pattern Recognition*, Vol. 2, pp. 2141–2144, 2000.
13. S. Medasani, R. Krishnapuram and Y. Choi. Graph Matching by Relaxation of Fuzzy Assignments. *IEEE Trans. on Fuzzy Systems*, **9**(1):173–182, February 2001.
14. M. Mortara, G. Patané, M. Spagnuolo, B. Falcidieno and J. Rossignac. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica*, Special Issue on Shape Algorithmics, Springer-Verlag. to appear.
15. B. T. Messmer, H. Bunke. A New Algorithm for Error Tolerant Subgraph Isomorphism Detection. *IEEE Trans. Patt. Anal. Mach. Intell.*, **20**(5):493–504, 1998.
16. J. Milnor. *Morse Theory*. Princeton University Press, New Jersey, 1963.
17. M. Mortara and G. Patané. Shape-Covering for Skeleton Extraction. *Int. J. of Shape Modelling* , **8**(2):245–252, 2002.
18. G. Reeb. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes Rendu Acad. Sciences*, **222**:847–849, 1946.
19. P. Salembier, A. Oliveras, L. Garrido Anti-extensive connected operators for image and sequence processing. *IEEE Trans. on Image Processing*, **7**(4):555–570, 1998.
20. D. Shattuck, R. Leahy, Automated graph based analysis and correction of cortical volume topology. *IEEE Trans. on Medical Imaging*, **20**(11):1167–1177, 2001
21. K. Siddiqi, A. Shokoufandeh, S.J. Dickenson and S.W. Zucker Shock graphs and shape matching. *Proc. of 6th Int. Conf on Computer Vision*, pp. 222–229, 1998.
22. R. C. Veltkamp and M. Hagendoorn. State-of-Art in Shape Matching. IN *Principles of Visual Information Retrieval*, M. Lew (Eds.) Springer-Verlag, pp. 87–119, 2000.