THE AUGMENTED MULTIRESOLUTION REEB GRAPH APPROACH FOR CONTENT-BASED RETRIEVAL OF 3D SHAPES

TONY TUNG and FRANCIS SCHMITT
GET/Télécom Paris, CNRS UMR 5141
46, rue Barrault
75634 Paris Cedex 13, France
{*tony.tung, francis.schmitt*}*@enst.fr*
http://www.tsi.enst.fr

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Communicated by (xxxxxxxxxx)

This article presents a 3D shape matching method for 3D mesh models applied to content-based search in database of 3D objects. The approach is based on the multiresolution Reeb graph (MRG) proposed by Hilaga et al [1]. MRG provides a rich representation of shapes able in particular to embed the object topology. In our framework, we consider 3D mesh models of various geometrical complexity, of different resolution, and when available with color texture map. The original approach, mainly based on the 3D object topology, is not accurate enough to obtain satisfying matching. Therefore we propose to reinforce the topological consistency conditions of the matching and to merge within the graph geometrical and visual information to improve matching and calculation of shape similarity between models. Besides, all these new attributes can be freely weighted to fit the user requirements for object retrieval. We obtain a flexible multiresolutional and multicriteria representation that we called augmented multiresolution Reeb graph (aMRG). The approach has been tested and compared with other methods. It reveals very performant for the retrieval and the classification of similar 3D shapes.

Keywords: 3D indexing; 3D mesh; Reeb graph; multiresolution representation; content-based retrieval; shape similarity.

1991 Mathematics Subject Classification: 22E46, 53C35, 57S20

1. Introduction

The strong development of numerical technologies has lead to efficient 3D acquisition of real objects and rendering of 3D methods. Nowadays 3D object databases appear in various areas for leisure (games, multimedia) as well as for scientific applications (medical, industrial part catalogues, cultural heritage, etc.). Large database can be nowadays quickly populated using 3D mesh acquisition and reconstruction tools which have become easy to use, and with new ergonomic 3D design tools which have become very popular. As database size is growing, tools to retrieve information become more and more important. 3D object indexing appears to be a useful and

very promising way to manage this new kind of data. Database indexing consists on defining a method able to perform comparisons between the database elements. Similarity retrieval is one of the main application: using a research "key", a subset of elements with the most similar keys are extracted from the database. The common guidelines of all 3D shape matching methods consist on creating a feature vector for each object from a database and propose a shape similarity measure to compare two feature vectors.

This article presents in Section 2 related work in the field of 3D objects retrieval in database. The study of the different approaches (global, local, using skeleton, etc.) has helped us to choose the most efficient methods and keep in sight our principal research interests. Section 3 is an overview of the multiresolution Reeb graph approach we have adopted [1]. We propose in Section 4 an extension of this method we named augmented multiresolution Reeb graph (or aMRG) [2]. Section 5 presents the similarity calculation between two aMRG. This approach is especially interesting for the variety and relevancy of information captured and the quality of the graph representation it provides based on multiresolution. Section 6 presents experimental results which emphasize the skills of this new proposed technique compared to the other tested methods.

2. Related Work

3D object indexing is a recent research area which has started around 1997. This field attracts more and more research groups. About 80% of the papers have been published in the last five years [3] and very different approaches have been proposed [4,5]. The 3D indexing problematic consists on describing 3D object shape in a compact manner (cf. Figure 1). The main goal is to perform shape recognition in large database. Therefore shape descriptors are used to produce feature vectors or signatures of 3D objects. These compact data stand for the objects and are the keys stored in database. In our study, 3D models are defined by their surface. 3D object surface is usually represented by a triangulated 3D mesh. If an object is initially defined by another representation, such as NURBS from CAD tools or 3D point clouds obtained by a laser scan, then the data must be first converted into a triangular mesh which is the most simple, easy-to-use, and popular representation of surfaces. 3D meshes are usually stored in database under the VRML format (Virtual Reality Modeling Language), most of them being at a low resolution [4,6]. 3D object databases including 3D meshes with their associated texture file are still rare, especially on the Internet [7,5].

The 3D model signatures can be efficiently used for attractive applications such as content-based search, comparison of models, classification, etc., if they satisfy the following set of criteria:

- compacity,
- speed and robustness of calculation,
- invariance to affine transformations (rotation, translation, scale factor),

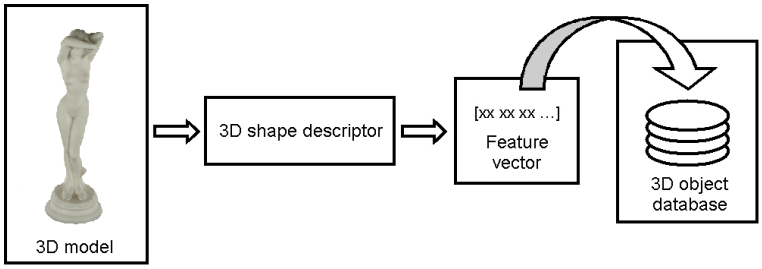

Fig. 1. 3D indexing consists on storing compact information related to a 3D model to optimise the searches in 3D object database.

- invariance to the mesh connectivity,
- robustness with regards to small mesh artefacts (deformation, decimation),
- relevance of stored information.

The approaches based only on curvature histograms [8] or EGI (Extended Gaussian Image) and its variants [9,10] describe very locally the geometry of the objects and thus are very sensitive to mesh modifications (noise on the vertex location, vertex or edge decimation, etc.). Furthermore they do not keep any topological information (mesh connectivity). Conversely, global approaches such as cord histograms [11], shape distributions [12], or weighted point sets [13] produce a too coarse description of the shape of 3D objects to be a sharp retrieval tool. Similar comments can be done for Fourier based descriptors and moments calculations [14,15]. Therefore using a skeletal or graph representation appears as very attractive as they permit to obtain an intuitive description of the shapes and to keep their topology. Unfortunately, the weakness of these methods lies in the extraction of the skeleton which is either time consuming, or too sensitive to noise on the object surface, or need a source point which favour a direction. In addition, most of the time no matching scheme is provided [16,17,18], or no multiresolution approach [19,20] well adapted for retrieval in large database. The multiresolutional Reeb graph (MRG) [1] appears to be a very promising starting point of research, in particular because it is an efficient mean to fuse and exploit different techniques as we will show in the next sections. The Reeb Graph of a 3D object is represented by a skeleton. It is built using a function μ based on the mesh connectivity. The surface of the object is divided in regions according to the values of μ, and a node is associated to each region. The graph structure is then obtained by linking the nodes of the connected regions. Afterwards the MRG is constructed hierarchically from the finest level of resolution to the coarsest level, which is the root of the graph. Similarity calculation between MRGs is done reversibly starting from the root and descending down to the finest level. Keeping advantage of the multiresolutional representation offered by the MRG, we add topological, geometrical and visual (colour and texture) information to the graph nodes to obtain a hierarchical description of the models, which

is both global and local and which we call augmented multiresolution Reeb graph (aMRG). As a consequence, 3D object indexing in database consists on coding in feature vectors all the relevant information associated to the nodes of the Reeb graph. Tests were performed upon databases made of a mixed set of low and high resolution 3D models [7,5,6].

3. Multiresolution Reeb graph

According to the Morse theory, a continuous function defined on a closed surface characterizes the topology of the surface on its critical points [21]. A Reeb graph is obtained assuming such a function μ calculated over the 3D object surface. Multiresolution results from the dichotomic discretization of the function values and from the hierarchical collection of the Reeb graphs obtained at each resolution.

3.0.1. *Reeb graph definition*

The Reeb graph is a graphical representation of the connectivity of a surface between its critical points. Reeb graphs were initially used to represent the skeleton of a compact manifold [22]. Let assume a real continuous function $\mu : S \to \mathbb{R}$ defined on the surface S of a 3D object. The Reeb graph is by definition the quotient space of the graph of μ on S, defined by the following equivalence relation between $\mathbf{X} \in S$ and $\mathbf{Y} \in S$:

$$\mathbf{X} \sim \mathbf{Y} \iff \begin{cases} \mu(\mathbf{X}) = \mu(\mathbf{Y}) \\ \\ \mathbf{X} \text{ and } \mathbf{Y} \text{ are in a same connected component of } \mu^{-1}(\mu(\mathbf{X})) \end{cases} \tag{1}$$

Example:
Let consider the set $S \in \mathbb{R} \times \mathbb{R}, \mathbf{v} \in S$ and μ such as $\mu(\mathbf{v}(x, y)) = y$.
$\forall \mathbf{X_1} = (x_1, y_1) \in S$ and $\mathbf{X_2} = (x_2, y_2) \in S$:

- $\mu(\mathbf{X_1}) = \mu(\mathbf{X_2})$ if and only if $y_1 = y_2$,
- $\mu^{-1}(\mu(\mathbf{X_1})) = \mu^{-1}(y_1)$ is the restriction of S to $S \cap (\mathbb{R} \times y1)$, that is to the "section" at ordinate y_1.

Two points will be in the same equivalence class if and only if they have the same ordinate, and if they are in the same connected component of the corresponding

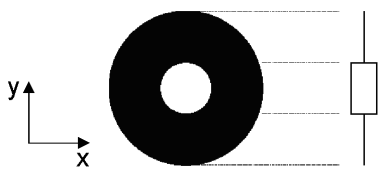

Fig. 2. Example of 2D Reeb graph.

Fig. 3. Example of μ functions: height function (left), distance to center of mass (middle), geodesic distance integral (right).

surface section (cf. Figure 2). The principal advantage of the Reeb graph relies on its ability to simply represent the topology of a set. We note that resulting graph totally depends on the choice of the μ function.

3.1. *The μ function*

In our framework, 3D models are defined by their surface represented as a 3D triangular mesh with vertices located in a Cartesian frame $R(x, y, z)$. Different continuous functions μ can be applied to construct a Reeb graph [20]. They all have different properties:

- Assuming a point $\mathbf{v}(x, y, z)$ on the surface S of an object, the height function defined as $\mu(\mathbf{v}(x, y, z)) = z$ is well adapted for models spread along the vertical axis as human being representations. This function is easy to compute but by definition is totally dependent of the object orientation. As a consequence, it is not suited for objects for which the points dispersal is perpendicular to the axis z.
- The function defined as the geodesic distance from curvature extrema is based on growing regions from local Gaussian curvatures on seed vertices. The results depend on the position of the seeds and require local curvature calculation which is not always exact, especially for the objects coming from the Internet which have often a mesh of a poor geometrical quality [4,6].
- The centroid function defined as the distance of a surface point $\mathbf{v}$ to the center of mass $\mathbf{G}$ of the object $\mu(\mathbf{v}) = d(\mathbf{G}, \mathbf{v})$, where d is the Euclidean distance,is simple to calculate and stable against small noise. However, this definition produces a Reeb graph which is not flexible enough with regards to our study, as explained next.
- We keep the function μ proposed in Ref. 1, which is defined as the integral of the geodesic distance $g(\mathbf{v}, \mathbf{p})$ from $\mathbf{v}$ to all the other points $\mathbf{p}$ of the surface (cf.

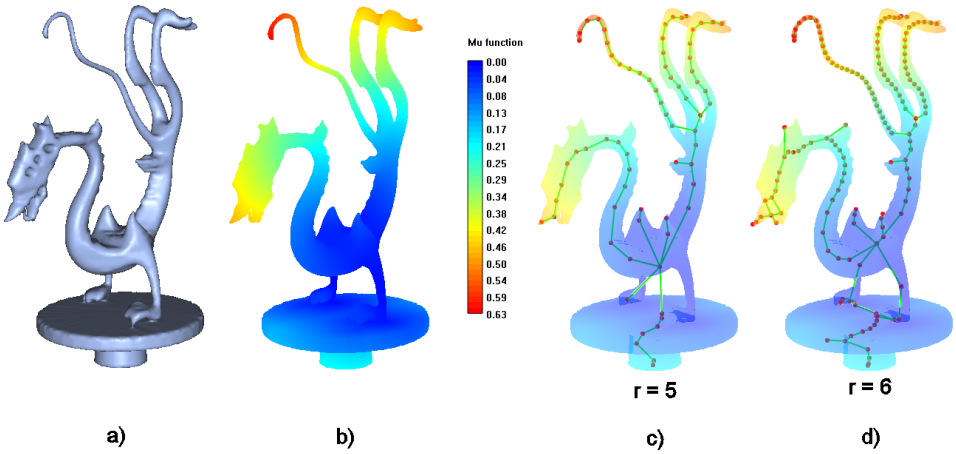

Fig. 4. Reeb graph: a) 3D model, b) function μ_N, c) and d) Reeb graphs at resolution $r = 5$ and $r = 6$ respectively.

Figure 3):

$$\mu(\mathbf{v}) = \int_{\mathbf{p} \in S} g(\mathbf{v}, \mathbf{p}) dS. \tag{2}$$

This function μ has the property to be invariant to rotations. Its integral formulation on the whole object surface provides a good stability to local noise on the mesh and gives a measure of the eccentricity of the object surface points. A point with a great value of μ is far from the center of the object and from the opposite side. A point with a minimal value of μ is close to the center of the object. Therefore, compared to the centroid function, this function has the ability to characterize intuitively better the topology of a shape. Arms of a human model will always have the same μ values independently of their position, which is not the case with the centroid function (cf. Figure 3).

Dividing the function μ by $max_{\mathbf{p} \in S} \mu(\mathbf{p})$ returns the normalize function μ_N which is invariant to scale transformation. The geodesic calculation is computationally costly, so the shortest path algorithm of Dijkstra is used as its time complexity is $\mathcal{O}(Nlog(N)$ where N is the number of vertices of the mesh [1]. The corresponding Reeb graph is then obtained by iteratively partitioning the object surface into regular intervals of μ_N values and by linking regions which are connected. For each interval, a node is associated to each different set of connected triangles (cf. Figure 4).

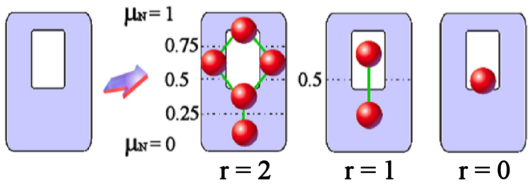

Fig. 5. MRG construction by merging the intervals of μ_N partitioning the object surface. Nodes from a level of resolution $r+1$ merge in parent nodes at the level of resolution r (from left to right: $r = 2, 1$ and 0).

3.2. *Multiresolution aspect*

To construct a Reeb graph of R levels of resolution, μ_N is subdivided into 2^R intervals from which the object surface is partitioned at the highest level of resolution. Afterwards, using a hierarchical procedure, a Reeb graph of a lower level of resolution is obtained by merging the intervals by pairs [1]. A parent node is associated to each connected set of each new resulting interval, and is linked to the n children nodes ($n \geq 1$) belonging to the previous higher level of resolution and from which it is the fusion (cf. Figure 5).

4. Augmented multiresolution Reeb graph matching

The strategy to compare two MRGs consists on matching between the two graphs all the nodes having a similar topological layout with respect to the connections to the neighbour nodes at the same resolution and to the parents/children nodes between successive levels of resolution. First, pairs of nodes verifying a predefined set of topological consistency criteria are detected. Then an analytical discrimination based on a *loss* function selects pairs of nodes among topologically consistent nodes. Finally the similarity of the two MRGs is calculated by matching the selected pairs of nodes. The approach is multiresolutional. The matching starts at the level of resolution $r = 0$. The procedure is recursively applied to the children node of the matched nodes until the highest level of resolution R is reached, or until no matchings are more possible. This top-down approach is a good alternative to avoid the NP-complete problem of subgraph isomorphisms [23]. In order to improve the matching, we have enriched the topological criteria and the geometrical attributes of the MRG that hence we named augmented multiresolution Reeb graph (aMRG). The following subsections present the additive topological criterion we have added to the original method [1] to overcome subsisting problems, and the new attributes we use to enhance the matching of the nodes.

4.1. *Additional topological consistency criteria*

Assuming two objects M and N we wish to compare their multiresolutional Reeb graphs. The goal of the topological consistency criteria is to select the pairs of nodes

(m, n), with $m \in \mathrm{MRG}(M)$ and $n \in \mathrm{MRG}(N)$, which could eventually be matched afterwards in the procedure. Two nodes m and n are topologically consistent if:

1. the parents m' and n' of m and n have been matched together at the previous level of resolution,
2. m and n correspond to the same interval of the μ_N function (it implies that only nodes at the same resolution are compared)
3. when two nodes are matched, a same label is assigned to them and is propagated in both graphs to their connected neighbours following the two monotonic directions of increasing and decreasing values of μ_N respectively [1]. Hence if two nodes m and n belong to a branch [a] of a graph on which some nodes have been already matched, they must have received the same labels to be also matched.

However the simple statement of these criteria is not sufficient to ensure that two branches will still be matched together when the resolution increases. In fact, branch matching information is not transmitted to the children nodes. Therefore it could happen that two branches which have been matched together lose their matching at the next higher level of resolution (see example on Figure 6). As a consequence, an additive criterion for the heritage is required to complete the three previous topological consistency criteria [1].

n°indent 4. m and n are topologically consistent if the parents of their neighbours (if they have ones) have been matched together at the previous level of resolution.

Therefore two nodes m and n at the level of resolution $r + 1$ will be matched together if they have two neighbours a and b with the parents a' and b', neighbours of the parents m' and n' of m and n respectively, have already be matched together at the level of resolution r.

Besides the node matching procedure also allows the matching between a node m and a set of nodes $\{n\}$ when m is topologically consistent to all nodes of $\{n\}$. The aim here is to alleviate the possible boundary (especially located at branch junctions) due to the segmentation of the object surface according to the discretization of function μ_N [1]. This property helps us to balance the rigidity of the 2nd topological consistency criterion.

4.2. *Analytical discrimination*

Topological criteria are not sufficient to discriminate the best matches between all the nodes or set of nodes which have been selected as candidates to the matching. Hence we define a loss function which calculate the difference of similarity between two nodes m and n. In Ref. 1, $loss(m, n) = \alpha|a(m) - a(n)| + (1 - \alpha)|l(m) - l(n)|$, a

[a] A branch of graph is a set of successive nodes ordered in μ_N and linked two by two by a single edge. Two branches match together when the nodes belonging to them match together.

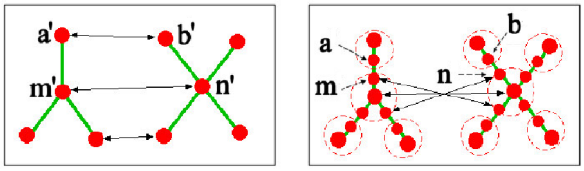

Fig. 6. Left: initially matched branches (a'-m' with b'-n'). Right: if, at the next higher resolution, two children nodes on different branches are wrongly matched (i.e. m is not matched with n), a label being propagated along the branches they belong to, those branches will not be correctly matched because their nodes will not be topologically consistent (a-m will not be matched with b-n). Therefore it is necessary to verify the matching of the parents of the neighbours a and b of two nodes m and n candidates for a matching in order to preserve the correct matching of the branches initially matched at the previous coarser level.

being the relative area of the set of triangles associated to a node with respect to the total area of the object surface, l being the relative size of the interval μ_N in which lies the node, with respect to the span $(\mu_{max} - \mu_{min})$ covered by the intervals of μ_N of all the nodes, and $\alpha \in [0, 1]$ a weighting term. $loss$ is minimal for the most similar nodes. To improve this step, we propose to introduce additive discriminative attributes and so the loss function becomes:

$$loss(m, n) = \sum_{i=0}^{N_f} \alpha_i d_i(f_i(m), f_i(n)), \tag{3}$$

where N_f is the number of attributes f_i, d_i is a distance associated to f_i, and α_i are weighting terms of the attributes f_i. The choice of the functions f_i is developed in the next sections. The choice of the constants α_i and distances d_i can be done empirically as described in Section 6. This formulation provides lots of freedom in the matching of nodes, in particular the constants α_i allow to weight the influence of each attribute f_i with regards to the type of the desired query.

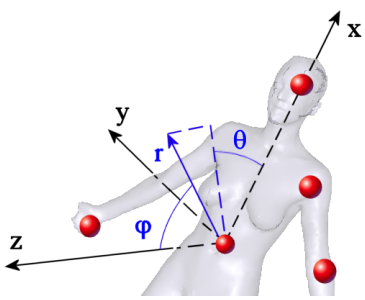

Fig. 7. Nodes localization in spherical coordinates.

4.3. *Geometrical attributes*

Matchings of graphs depending only on topology as in Ref. 1 are not always appropriate. For example, let consider a human model. An arm is topologically equivalent to a leg (they are limbs linked to the body), whereas we would like to get more natural and correct matching in our framework. Moreover, the topological consistency criteria are such defined that if two nodes are not matched together at a certain level of resolution, whereas they should be, then there is no possibility that their children nodes match together at a finer resolution. Therefore it is crucial that nodes are matched at best they can from the beginning of the consistency tests. For this reason, we propose to add information of localization in the node. A node m being associated to a set of connected triangles, so it can be located at the barycenter $m(x, y, z)$ of the triangles (cf. Figures 4 and 5). We choose a spherical coordinate representation $(\mathtt{r}, \theta, \phi)$:

- $\mathtt{r}(m) = \sqrt{x^2 + y^2 + z^2} \geq 0$, the distance between the node and the center of the frame,
- $\theta(m) = \arctan(y/x) \in [-\pi, \pi]$, the azimuth,
- $\phi(m) = \arccos(z/\mathtt{r}) \in [0, \pi]$, the elevation.

Using the example of a human model (cf. Figure 7) for which the x axis is oriented from the bottom to the top of the body, the y axis from left to right and the z axis from behind to front, we can observe that θ allows us to identify the left from the right, ϕ to distinguish arms from legs and $\mathtt{r}$ to be decisive in an ambiguous configuration such as an arm up raised to the top of the head. It becomes necessary that objects are well positioned model with respect to the referential. For this purpose, we decide either to use the natural axis of the object centered to the object center of mass with a scale normalization of the coordinates, or to rotate

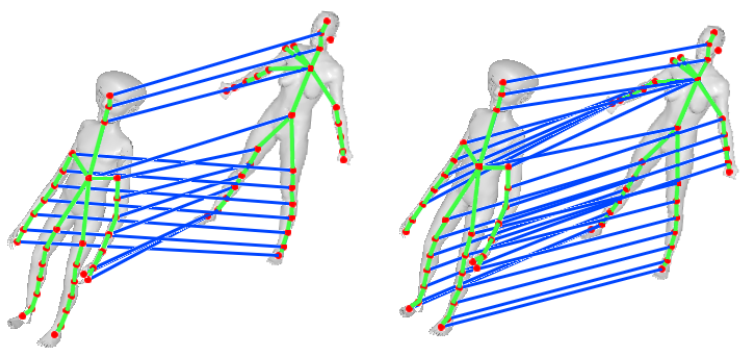

Fig. 8. Left: without geometrical information, legs can be matched with arms as there are topologically equivalent (links denote matching nodes). Right: by adding geometrical information, arms and legs are well matched.

the object by applying a classical axis alignment technique such as the Principal Component Analysis [14].

Figure 8 shows the alignment of two 3D models and the gain obtained by adding geometric attributes for the node localization in the expression of the *loss* function in Eq. 3.

5. Similarity calculation

The matching approach presented in the previous section returns the list of pairs of matched nodes between two aMRG $\{(m,n)/m \in \text{aMRG}(M)$ and $n \in \text{aMRG}(N)$, m matching with $n\}$. The similarity measure of two aMRG is calculated by summing the similarity of every pairs of matching nodes. First, a function calculates the similarity between each matched nodes. This similarity function, derived from [1], has been enriched with various features extracted from the mesh models. The following sections present the new similarity function between nodes, the proposed features, and the resulting similarity function between graphs.

5.1. *Similarity of nodes*

The similarity function between two nodes m and n has the same formulation as the *loss* function presented in Eq. 3. It relies on the same attributes f_i:

$$sim(m,n) = \sum_{i=0}^{N_f} \lambda_i d'_i(f_i(m), f_i(n)), \tag{4}$$

where d'_i is a distance and λ_i a weighting term associated to the attribute f_i, with $\sum_i \lambda_i = 1$.

To improve the object retrieval by shape similarity and allow more flexible queries with a larger set of characteristics based on geometry, colorimetry, etc. we have merged global and local geometric properties and visual properties extracted from the object surface region $S(m)$ associated to each node m:

- $v(m)$, the relative volume of $S(m)$,
- $cord(m)$, a statistic measure of the extent of $S(m)$,
- $curv(m)$, a statistic of the Koenderink shape index [8] locally estimated on $S(m)$,
- $hough(m)$, a statistic of the orientation of the triangle normals associated to $S(m)$,
- $color(m)$, a statistic of the texture/color mapped on $S(m)$ when available.

The choice of these attributes has been guided by the literature on 3D content-based retrieval. The selection of the distances d'_i and the weights λ_i will be discussed in Section 6.

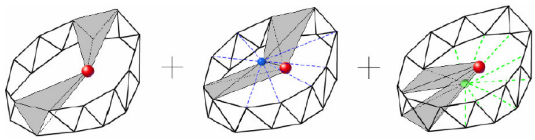

Fig. 9. Volume associated to a node obtained by summing the volume of the three types of tetrahedra included in the section.

5.2. *Volume calculation*

Let consider the interval $[\mu_1, \mu_2]$ of function μ associated to the node m. The relative volume $v(m)$ is defined as:

$$v(m) = \frac{1}{R+1} \frac{vol(m)}{vol(S)}, \tag{5}$$

where $vol(S)$ represents the total volume of the model, and $vol(m)$ the volume corresponding to the region $S(m)$ associated to m. $vol(S)$ is the sum of the tetrahedra formed by the oriented triangles of the mesh and the barycenter of the object [25]. $S(m)$ can be seen as a section of the object: $vol(m)$ is obtained by summing 1) the signed volume of the tetrahedra formed by the triangles associated to m and the barycenter of these triangles, 2) the signed volume of the tetrahedra formed by this barycenter, an edge of the mesh corresponding to the section border $\mu = \mu_1$ associated to m, and the barycenter of the points associated to this border, and 3) the symmetric volume for the surface section border $\mu = \mu_2$ (cf. Figure 9). The factor $\frac{1}{R+1}$ corresponds to the levels of resolution $r = 0, \ldots, R$ of the graph aMRG(M), leading to the relation $\sum_{m \in \text{aMRG}(M)} v(m) = 1$. Obviously, this volume calculation method is valid only if the 3D model is complete, its surface being hole free, and if the triangles have been oriented outwards.

This feature can be combined with the relative area attribute $a(m)$. We obtain an additive characterization of the section of the object associated to the node m: by introducing the ratio $Ratio(m) = 36.\pi.v(m)^2/a(m)^3$ which has no dimension. This ratio is particularly discriminant at the level of resolution $r = 0$ with primitive shapes (cf. Section 6) as the sphere (ratio=1; surface is minimal), cone (~ 0.47) or cube (~ 0.51).

5.3. *Cord statistics*

The cords measure introduced in Ref. 11 is represented by a normalized histogram of the distance between the object barycenter and the barycenter of each triangle. Applied to a node m, the histogram $cord_L(m)$ is the histogram of the distances between the barycenter of the object section $S(m)$ associated to node m and the barycenter of every triangles belonging to $S(m)$. This measure provides us with intuitive information about the extent of the set of triangles associated to a node (cf. Figure 10). Each distance to a triangle is weighted by the relative triangle area

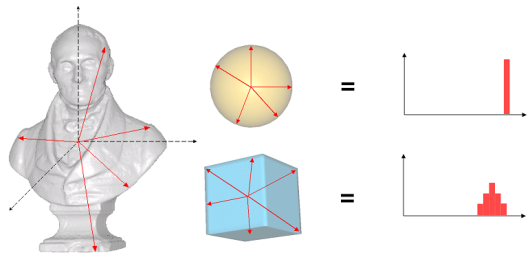

Fig. 10. Cord measurement characterizes the object geometry

normalized by the total surface area of the object. The cord lengths of m are also normalized with respect to the maximal length of the cords of m in order to adapt the dynamic of the histogram and to use the maximal range of its bins at each node.

As in Ref. 11 we add also to the cord length histogram the two histograms $cord_1(m)$ and $cord_2(m)$ of the angles of the cords with respect to the first two axis of the principal component analysis. At the coarsest level of resolution $r = 0$, the barycenter of the node is the object center of gravity. Therefore the histograms are equivalent to those used in Ref. 11 which are applied on the entire mesh.

Statistical measures based on histograms as the cords of Ref. 11, or the shape distribution of Ref. 12, describe the full shape of an object and seem to be not discriminative enough to compare complex shapes. In our context and from a certain level of resolution, nodes are much more easy to compare and are mainly associated to simple shapes similar e.g. to cylinders or paraboloids which justify the relevant use of these local geometrical measures.

5.4. *Local shape index*

Geometrical properties of the surface can be locally characterized by a local curvature measurement. For this purpose we use the shape index introduced by Koenderink [24,8] (cf. Figure 11).

At a point $\mathbf{P}$, the shape index $I(\mathbf{P})$ is defined as follows:

$$I(\mathbf{P}) = \frac{1}{2} - \frac{1}{\pi} atan(\frac{k_2(\mathbf{P})}{\sqrt{|k_2^2(\mathbf{P}) - k_1(\mathbf{P})|}}), \tag{6}$$

where $k_1(\mathbf{P})$ is the Gaussian curvature and $k_2(\mathbf{P})$ the mean curvature. The Gaussian curvature at a vertex $\mathbf{P}$ is calculated using the angular defect formula:

$$k_1(\mathbf{P}) = \frac{2\pi - \sum_j \theta_j}{\frac{1}{3}\sum_j A_j}, \tag{7}$$

where θ_j is the angle at vertex $\mathbf{P}$ in the jth triangle of the 1-ring of $\mathbf{P}$, and A_j is the area of this j^{th} triangle. The mean curvature $k_2(\mathbf{P})$ at vertex $\mathbf{P}$ is calculated

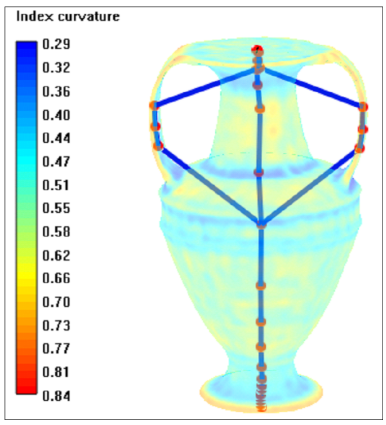

Fig. 11. Koenderink shape index calculated on each vertex of the mesh and used as node attribute.

according to the usual formula, by summing half of the angles $\phi_i, i = 1, ..., N_e$, between the normals of the pairs of consecutive triangles in the 1-ring of $\mathbf{P}$, N_e being the number of triangles adjacent to $\mathbf{P}$. Angles are weighted with the third of the mean area $\overline{A}_i$ of the corresponding pair of triangles:

$$k_2(\mathbf{P}) = \frac{1}{N_e} \sum_i \frac{\frac{1}{2}\phi_i}{\frac{1}{3}\overline{A}_i}. \tag{8}$$

Index $I(\mathbf{P})$ lies between 0 and 1. Therefore it is very suitable for normalized statistics. Shape index histogram $curv(m)$ is estimated for each node m at the highest level of resolution R. The index is calculated for each vertex of the mesh associated to m, each contribution of a point $\mathbf{P}$ being weighted by the third of the relative areas of the triangles adjacent to $\mathbf{P}$ and belonging to the section associated to m, a relative area being normalized by the total area of the object.
Histograms of parent nodes are obtained by summing the histograms of the children nodes. At level of resolution $r = 0$, the histogram is equivalent to the 3D shape spectrum descriptor calculated on the entire mesh [8].
The main weakness of this characterization is that it is unpracticable directly on models with a too noisy mesh. As a matter of fact, the local curvature estimation requires good continuity property of the surface (smoothness), meaning a good quality mesh. Hence it can be necessary to filter the data before applying the curvature calculation in case of irregular mesh.

5.5. *3D Hough descriptor*

The 3D Hough descriptor was proposed in Ref. 8 to succeed to the 3D shape spectrum descriptor. The main strength of this approach is its robustness with regards to local deformations of the mesh (mesh connectivity, vertex or edge decimation).

The 3D Hough descriptor $hough(m)$ of a node m characterizes the spatial distribution of planes associated to the triangles of the mesh associated to m. Each triangle is assigned to a plane Π parameterized in spherical coordinates $(\mathtt{r}, \theta, \phi)$, $\mathtt{r}$ being the distance of the center of the node m to Π, and (θ, ϕ) the azimuth and elevation angles representing the normal direction of Π in the object frame: $\mathtt{r} \geq 0, \theta \in [-\pi, \pi]$, and $\phi \in [0, \pi]$.

In order to obtain a homogeneous partition of the normal orientations, the faces of a subdivided octahedron are mapped on the unit sphere. Thus inhomogeneity problems at the poles due to the uniform sampling of the θ and ϕ axis in spherical coordinate parametrization are avoided. The attribute $hough(m)$ of the node m is a 3D histogram accumulating the $(\mathtt{r}, \theta, \phi)$ values of the triangles of m. Each contribution is weighted by the relative area of the corresponding triangle. At the level of resolution $r = 0$, the histogram is equivalent to the 3D Hough descriptor calculated over the entire mesh [8].

5.6. *Color texture attributes*

When browsing in a database of 3D textured objects, it can be also interesting to make queries upon the texture or the color of the models. The joint use of shape and texture characteristics is a highly discriminative criterion, but is still not much exploited in 3D. For this purpose, we use the information included in the texture maps associated to the 3D objects. These 2D features can be merged in a natural way to the 3D features previously presented (cf. Figure 12). We consider here only the simple color attributes formed by the three normalised 1D histograms on R, G and B. These histograms are first determined for each node of the aMRG at the highest level of resolution $r = R$. For each triangle of the section $S(m)$ associated to m, corresponding texels [b] from the texture map are collected. Each texel contribution to histograms is weighted by the relative size of the texel with respect to the relative size of the projected triangle. The sum of the texel contributions is

[b] A texel (texture element) is an elementary pixel of the texture map.

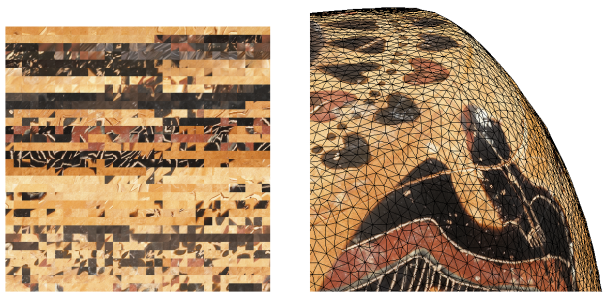

Fig. 12. Example of mesh with associated texture map.

equal to the relative area of the triangle. Afterwards, iteratively and hierarchically the nodes of the lower levels of resolution are obtained as follows:

$$color(m) = \frac{1}{R+1} \sum_{n \in \{\text{children of } m\}} color(n), \tag{9}$$

with at each level of resolution $r \in [0, R]$, $\sum_{m \in res(r)} color(m) = 1$.

5.7. *Object similarity measure*

Assuming two objects M and N, the function used to calculate their similarity is obtained by summing the similarity function of the matched pairs of nodes (m, n):

$$SIM(M, N) = \sum_{\{m,n\}} sim(m, n). \tag{10}$$

$SIM(M, N)$ takes its values in the range $[0, 1]$ due to normalized histograms, relative values (aera, volume, ratio) and barycentric weighting terms. The function SIM has the following properties:

- Self-identity: $SIM(M, M) = 0$,
- Positivity: $SIM(M, N) \geq 0$,
- Symmetry: $SIM(M, N) = SIM(N, M)$.

The function SIM is then a semi-metric [26]. The triangle inequality is not verified as similarity calculation between two graphs takes into account the number of matched pairs, which vary when comparing different objects to a same one.

Finally, we have obtained a flexible multiresolutional 3D indexing scheme, including both global and local information, and based on different features belonging to different domains (topology, geometry, and colorimetry). New features could be easily embedded to the proposed indexing approach.

6. Experimental results

6.1. *Databases*

We have made our tests on different databases. The aim was to control the ability of the aMRG approach to retrieve similar objects and classes with different levels of difficulty and to compare aMRG with previous methods. For that we have built three databases of different sizes.

The first database is composed of 116 models classified in 16 classes (cf. Figure 13). The objects are different either in topology or in shape or in mesh resolution. Same classes are composed of primitive shapes such as cubes, spheres, cones, torus, and double torus, which were designed interactively with a 3D modeller. Some models have been reconstructed from real objects [7,5] and others have been downloaded from the Internet [6].

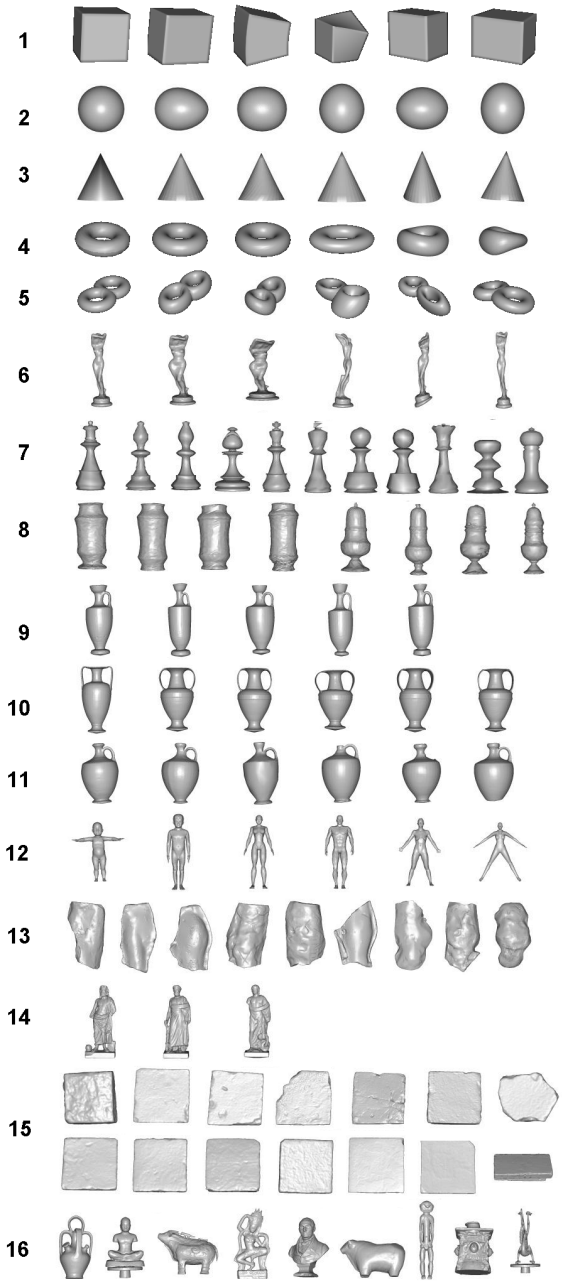

Fig. 13. Test database composed of 116 objects in 16 classes.

There are meshes of low resolution with $\sim$ 500 triangles (chess models) and meshes of high resolution with $\sim$ 150 000 triangles (statues). Some meshes are of poor quality and contain holes or have a non manifold mesh. All objects in each class have similar shapes, except for the class n° 16 which is composed of 9 different statues. Besides, some classes are quite close and could be easily confused by a bad descriptor such as the three vase classes n° 9, 10 and 11, or the human class n° 12 and the statue classes n° 6 and n° 14. This first database is used as a training database to optimize the set of weighting terms of the aMRG attributes and to select for each tested methods the distance which provides the best results for classification. The second database contains 266 objects classified in 26 classes (cf. Figure 14 a)). It includes the training database without the previous class n° 16 composed of miscellaneous statues. In addition it has been exclusively populated with 3D objects of museums. Lots of these 3D museological objects have a high resolution with more than 400 000 triangles. The interest of this database relies in its complex classification as some of the 11 additional classes are very close. The experiments on this database allow us to test the accuracy of the 3D descriptors.

The second database has been completed with 301 different objects to form the third database of 567 objects (cf. Figure 14 b)). The additional 3D objects are again exclusively models from museums. Some of them have a shape close to the classified objects but remain different enough to be intuitively not classified with the others. This database can be seen as a noisy database and should point out the ability of the various 3D descriptors to retrieve the object classes.

Fig. 14. a) Samples of object classes of the database of 266 objects. b) Samples of unclassified objects of the database of 567 objects

6.2. *Evaluation method*

For each object, the structure of the graph aMRG and its attributes are coded in a binary file. The size of the file depends on the complexity of the object and increases linearly with the number of nodes. These files are stored in the database as feature vectors. Running on a PC P4-2GHz with 1Go of RAM, one query on an aMRG with a maximum resolution level of $R = 5$ is performed in $\sim 5s$ on the database of 116 objects, in $\sim 10s$ on the database of 266 objects, and in $\sim 30s$ on the database of 567 objects. At resolution $r = 3$, this time for one query are reduced to $\sim 4s$, $\sim 8s$, and $\sim 19s$ respectively. As a comparison, the time for one query with a statistical method with a feature vector of size 128 (e.g. an histogram with 128 bins) are $\sim 3s$, $\sim 6s$, and $\sim 14s$ respectively. The level of resolution of the aMRG chosen for the query has then a strong impact on the speed of calculation. Indeed the graphs computation and comparison is more computationally costly when calculating similarity of complex shapes than of simpler primitive shapes. The aMRG feature vector construction at resolution $R = 5$ requires $\sim 30s$ for an object of $\sim 50\,000$ triangles. The computation of the function μ over the object surface is the most time consuming. For that we use the Dijkstra algorithm which has a complexity in $\mathcal{O}(Nlog(N))$, N being the number of vertices of the mesh.

In order to point out the discriminative power of the aMRG approach, we have made comparisons with the following various 3D indexing methods :

- cords histograms [11],
- D2 shape distribution [12],
- spherical harmonics [15],
- 3D shape spectrum [8],
- 3D Hough descriptor [8],
- complex EGI [10],
- area volume ratio (cf. Section 4 and [25]).

Each of these methods describes a specific geometrical characteristic of a 3D model and stored it as a feature vector. Similarity between two models can be measured using a distance between their two feature vectors [27]. To find the better distance for each method, we have studied and compared the performance of the following distances on our three classified databases :

- $L1$ Minkowski distance,
- $L2$ Minkowski distance,
- L_∞ Minkowski distance,
- Bhattacharyya distance,
- Jeffreys distance,
- χ^2 divergence,
- Histogramm intersection,
- Earth Mover distance [13,28].

The most computer time consuming is the Earth Mover distance, which is in $\mathcal{O}(N^4 log(N))$, N being the number of bins. Other distances are in $\mathcal{O}(N)$.

The dimension of the feature vectors varies depending on the chosen number N of quantization levels which results in N histogram bins. We have evaluated the influence of this choice for each method with the following values:

- 16 and 128 bins for cords histograms and the 3D shape spectrum descriptor,
- 16, 128 and 1024 bins for the D2 shape distribution,
- $128(4 * 32)$ and $512(4 * 128)$ bins for the 3D Hough transform projected on an octahedron with its 8 faces which have been subdivided one time and two times, resulting in 32 and 128 faces respectively.
- 384 bins ($128*3$) for an "extended" complex EGI. Three characteristics are stored in each cell with regards to the set of associated triangles: sum of the relative areas of the triangles, mean of the cord lengths (distance between the center of gravity of the model and the triangle barycenters), and mean of the angles between the normals and the cords.

To evaluate the ability of a method to retrieve similar objects in a database, we use usual tools of classification [4,29]:

- The *best matches images* show the k first objects calculated by a method as the most similar to a query.
- The *distance matrix* is the symmetric matrix, with its element (i, j) corresponding to the value of similarity between object i and j. Objects are gathered in classes along both dimensions. This representation gives a rapid view of the classification skills of the evaluated method. As a matter of fact, by assigning dark colors for similar objects having a strong similarity value, the distance matrix of a discriminative method should reveal dark squares aligned along the diagonal of the matrix. Each one of these squares corresponds to the similarity values of objects belonging to a same class. This representation allows us to characterize the homogeneity of the classes.
- The *tier image* represents for each query object of a class of size C: 1) the nearest neighbours of the query which belong to the same class among the $C - 1$ first retrieved objects (with black pixels in figures), 2) the objects among the $C - 1$ first retrieved objects which are not from the same class (red pixels), 3) the $C - 1$ following objects retrieved between the rank C and $2(C - 1)$ (blue pixels). The different colors help to quickly give an estimation of the well retrieved classes. The squares on the diagonal of the image corresponding to a same class should be completely black for a very discriminative method, and black and blue for a less discriminative method. If white points (i, j) remain inside the square, it mean that for a query corresponding to object i, the object j belonging to the same class as i has not been retrieved by the method within the $2(C - 1)$ nearest objects. Note that the matrix is not symmetric and has to be read line by line.

- The *precision-recall* curve is a statistical calculation over the database representing the ability of the evaluated method to retrieve the classes. Only classified objects have to be taken into account. Considering an object i of a given class, let assume C to be the size of its class, R the chosen number of returned objects by the request, and Rr the number of objects belonging to the same class as i among the R returned ones. The curve is composed by the *precision* in ordinate, defined as the ratio $Precision = Rr/R$, and by the *recall* in abscissa, defined as the ratio $Recall = Rr/C$. A perfect discriminative method would then provides a horizontal line at $Precision = 1.0$.

The problem with these representations, especially the tier image, is that they become difficult to read when there are many objects and classes in the database. Furthermore it becomes difficult to compare them when they look similar. Therefore it is also useful to calculate score values to summarize method performances [4]:

- The *first tier score* T_1 is obtained by calculating the macro average of the percentage of well retrieved objects (belonging to the same class of size C as the query object) among the $C-1$ first retrieved objects, meaning the average of the average of the percentage of each class:

$$T_1 = \frac{1}{Nc}\sum_{i=1}^{Nc} Tc_i \quad \text{and} \quad Tc_i = \frac{1}{C_i}\sum_{j=1}^{C_i} T_j^i, \tag{11}$$

  where Nc is the number of classes in the database, Tc_i the first tier score of the ith class, C_i the size of the ith class and T_j^i the first tier score of the jth object of the ith class.
- The *second tier score* is similarly obtained but with percentages calculated over the $2(C-1)$ first retrieved objects, and similarly also for the bulls eye percentage score (BEP) which is calculated over the $2C$ first retrieved objects. The BEP has been officially chosen in the MPEG-7 norm [8].
- The *E measure* introduced by Ref. 4 characterizes the efficiency of the classification, based on the first 32 retrieved objects. The calculation relies on the precision and recall values:

$$E = \frac{2}{\frac{1}{P} + \frac{1}{R}}. \tag{12}$$

  Higher values stand for better retrieval and maximum score is 1.0.
- The *discounted cumulative gain* (DCG) cumulates in all the database contributions with weights depending on the ranking of the retrieved objects [4]. The contribution of the rth retrieved object is 0 if it is not in the same class of the query, and is equal to $\frac{1}{log_2(r)}$ if it is in the same class. Afterwards the DCG is normalized by the maximum possible DCG score corresponding to a perfect retrieval.

Among all the methods we implemented for comparison to our proposed aMRG approach, the 3D Hough descriptor [8] has shown the best performance on the train-

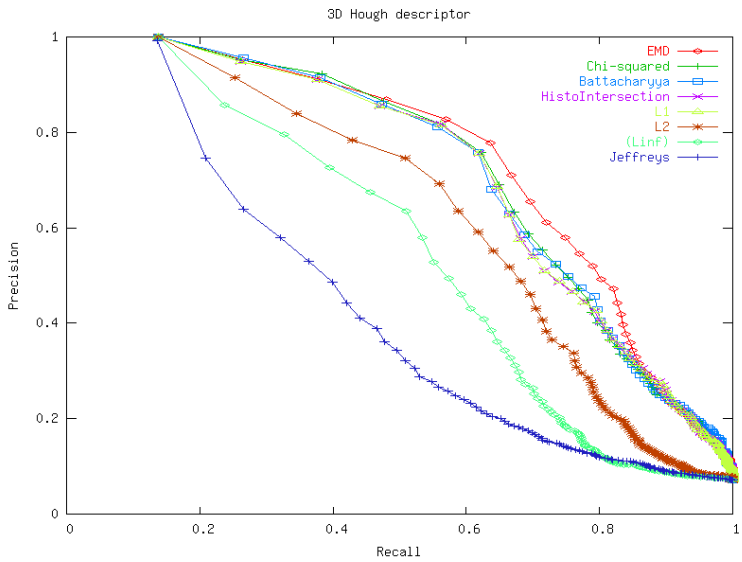

Fig. 15. Precision-recall curve computed for the 3D Hough descriptor with eight different distances. Earth Mover distance provides the best results.

ing database of 116 objects. Therefore we first present here the results obtained with the 3D Hough descriptor approach. The precision-recall curves show better results for the Earth Mover distance [28] compared to the other tested distances (cf. Figure 15). However, there are no significant differences with three of the other distances which are "cheaper" to compute (χ^2, bhattacharyya and histogram intersection). For this test we used a single subdivision of the octahedron which performs better on the training database than two subdivisions. Thus the feature vector size is $4 * 32 = 128$ floats.

Calculation of the distance matrix and tier image reveals the ability of the 3D Hough descriptor approach to retrieve 13 classes out of 16. One class is not retrieved at all (n° 16) and two others (n° 12 and 13) are only partially recovered (cf. Figure 16). The 3D Hough descriptor seems to present difficulties to retrieve the human and mould classes (n° 12 and 13), whereas it performs well on primitive classes (n° 1 to 5), vases (n° 8 to 11) and paved stones (n° 15). Classes corresponding to various distorted versions of a same model are not well retrieved. This can be observed with the torus (class n° 4) and the goddess statues (class n° 6). This behaviour of the 3D Hough descriptor relies on the calculation of normal orientations. Important geometrical variations of the surface affect a lot the feature vector representations. Moreover the original approach takes only the distance of the origin to the planes which are larger than 70% of the maximal distance. The moulds are globally flat and concave, and human models filiform; as a consequence

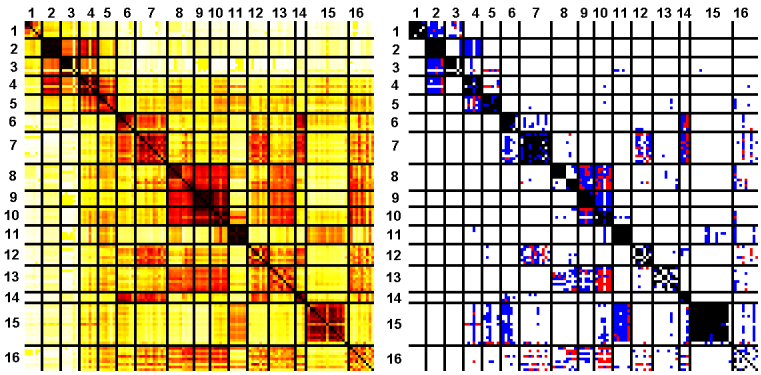

Fig. 16. Left: 3D Hough distance matrix. Right: 3D Hough tier image. These representations show the ability of the 3D Hough descriptor to retrieve classes.

they are difficult to characterize using this approach.

6.3. *aMRG results*

The experimental results have led us to test all the combinations of distances and methods and, using the T_1 and BEP scores, we have evaluated the performance of each method associated to its best distance. The aMRG has been calculated with resolutions up to $r = 5$. The aMRG returns equivalent scores for the resolution levels $r = 4$ and $r = 5$, and for the histogram intersection distance (cf. Table 1). Therefore we choose the resolution $r = 4$ for performance reason. We observe that the L1 distance, χ^2 divergence and EMD return the best results on this database for most of the methods (cf. Table 2).

Table 1. Score values (T_1, BEP, E measure and DCG) on database 1 (116 models) for the aMRG method at resolutions $r = 0$ to 5 and with the histogram intersection distance.

r	0	1	2	3	4	5
T_1	0.867	0.868	0.882	0.905	**0.912**	0.914
BEP	0.962	0.964	0.976	0.983	**0.985**	0.984
E	0.346	0.346	0.346	0.348	**0.348**	0.347
DCG	0.921	0.922	0.928	0.932	**0.935**	0.935

We can read the results as the follows: for a query on the database of 116 objects, the aMRG has 86.6% of good results among the $C - 1$ first retrieved objects (C being the size of the query class), and 94.0% of good results among the $2C$ first retrieved objects. The precision-recall curves also show the superiority of the aMRG compared to the other methods on the first database (cf. Figure 17). The distance

Table 2. Score values (T_1, BEP, E measure and DCG) on database 1 (116 models) for the best combinations of distances and implemented methods.

	Shape dist D2	Sph harmo	Vol area ratio	Shape spec	Cord length	CEGI	3D Hough desc	aMRG $r = 4$
	L1	χ^2	L1	χ^2	EMD	Jeffreys	EMD	Histo int
T_1	0.337	0.383	0.491	0.541	0.691	0.735	0.792	**0.912**
BEP	0.458	0.529	0.681	0.663	0.808	0.864	0.900	**0.985**
E	0.219	0.247	0.296	0.273	0.308	0.331	0.329	**0.348**
DCG	0.625	0.657	0.726	0.751	0.850	0.869	0.895	**0.935**

matrix and the tier image show the ability of the aMRG to retrieve all the classes (cf. Figure 18), except the heterogeneous class n° 16 composed of miscellaneous statues. The primitive shapes were well classified, as well as the different vase classes which remain close to each other. The mould class n° 13 has also been well retrieved and is close to the paved stones class n° 15. Thus, compared to other methods the aMRG has in addition the ability to give more homogeneous classification (cf. Figure 18 right).

For this experiment, attributes and weighting terms of the aMRG have been empirically chosen as follows :

- $f_0 = a$, $\lambda_0 = 0.2$,
- $f_1 = cordL$, $\lambda_1 = 0.2$,
- $f_2 = cord1$, $\lambda_2 = 0.1$,
- $f_3 = cord2$, $\lambda_3 = 0.1$,
- $f_4 = curv$, $\lambda_4 = 0.1$,
- $f_5 = hough$, $\lambda_5 = 0.2$,
- $f_6 = ratio$, $\lambda_6 = 0.1$.

To point out the advantages of the aMRG we also present in Figure 19 the results obtained with the original MRG approach [1]. The distance matrix and the tier image show results of lower quality and a worse classification. Scores are also lower compared to aMRG scores: $T_1 = 0.629$ and $BEP = 0.751$. Results are rather spread out in the matrix instead of being concentrated around the diagonal. The aMRG brings then a real improvement to the original method.

The experiments with regards to the second database of 266 classified objects show also that the aMRG approach performs the best for shape similarity retrieval and classification (cf. Figure 20). A study of the aMRG at different levels of resolution with the same settings as the previous test reveals that the aMRG returns on this database better results at the level of resolution $r = 3$ and with the histogram intersection distance (cf. Table 3). The computation of the similarity measure at the resolution $r = 3$ makes the query faster. The aMRG was also compared to the other implemented methods with their best associated distance. Scores are reported on Table 4.

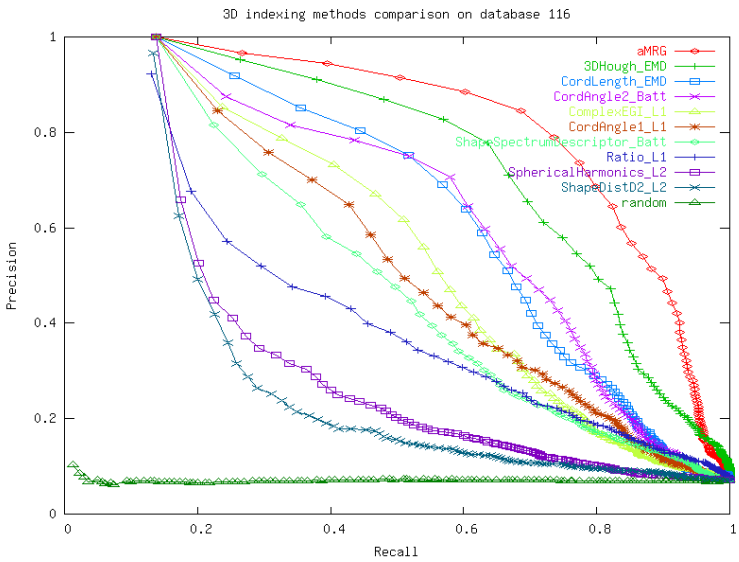

Fig. 17. Precision-recall curves of the tested 3D indexing methods on the first database of 116 objects. aMRG returns the best performance.

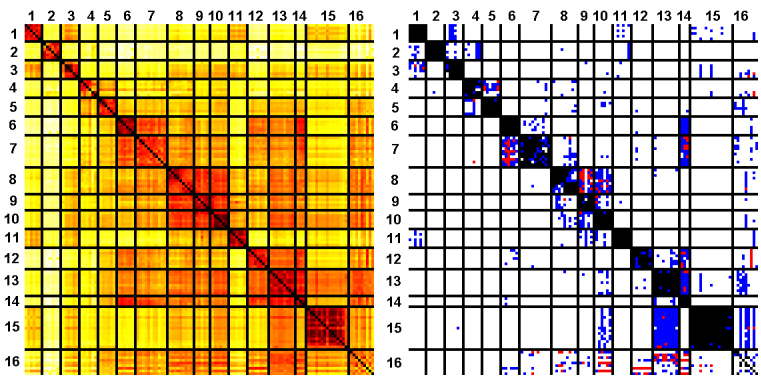

Fig. 18. Left: aMRG distance matrix. Right : aMRG tier image. aMRG method retrieved all the classes (except the 15th). Moreover the classification is homogeneous.

Finally we run the algorithms on the complete database of 567 objects, and again the aMRG still performs better than the other approaches. The aMRG was computed a level of resolution $r = 3$ with the same settings as the previous test, and was compared to the other methods with the distances of the previous test. This time the aMRG associated to the histogram intersection distance has returned slightly better results than with the $L1$ distance. However performances of retrieval are lower for all the methods (cf. Figure 21 and Table 5), which is simply due

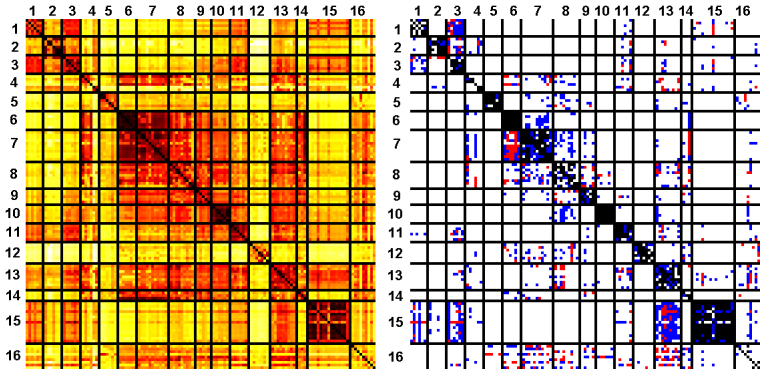

Fig. 19. Left: MRG distance image. Right : MRG tier image. The original MRG approach returns lower performance than the aMRG approach.

Table 3. Score values (T_1, BEP, E measure and DCG) on database 2 (266 models) for the aMRG method at resolutions $r = 0$ to 5 and with the histogram intersection distance.

r	0	1	2	3	4	5
T_1	0.749	0.750	0.762	**0.770**	0.747	0.733
BEP	0.858	0.859	0.871	**0.874**	0.852	0.833
E	0.391	0.391	0.394	**0.394**	0.387	0.380
DCG	0.871	0.872	0.875	**0.880**	0.867	0.859

Table 4. Score values (T_1, BEP, E measure and DCG) on database 2 (266 models) for the best combinations of distances and implemented methods.

	Shape dist D2	Sph harmo	Vol area ratio	Shape spec	Cord length	CEGI	3D Hough desc	aMRG $r = 3$
	L1	χ^2	L1	χ^2	EMD	Jeffreys	EMD	Histo int
T_1	0.250	0.294	0.304	0.404	0.551	0.614	0.667	**0.770**
BEP	0.326	0.416	0.468	0.499	0.656	0.751	0.790	**0.874**
E	0.160	0.205	0.246	0.250	0.311	0.354	0.360	**0.394**
DCG	0.555	0.595	0.637	0.672	0.767	0.804	0.835	**0.880**

to the huge amount of unclassified objects added to the database 3. E measure characterizes the ability to retrieve objects of the same class among the 32 first most similar objects to an object query and its calculation is based on precision and recall. Therefore as can be seen on Figure 21 and Table 3 the score on the "noisy" database 3 is lower as retrieval is more difficult. Compared to the other methods, the DCG score of the aMRG approach is still high, meaning that objects of a same class (i.e. similar objects) are still found close to the object query. We observe also on this database that the 3D Hough descriptor still works much better

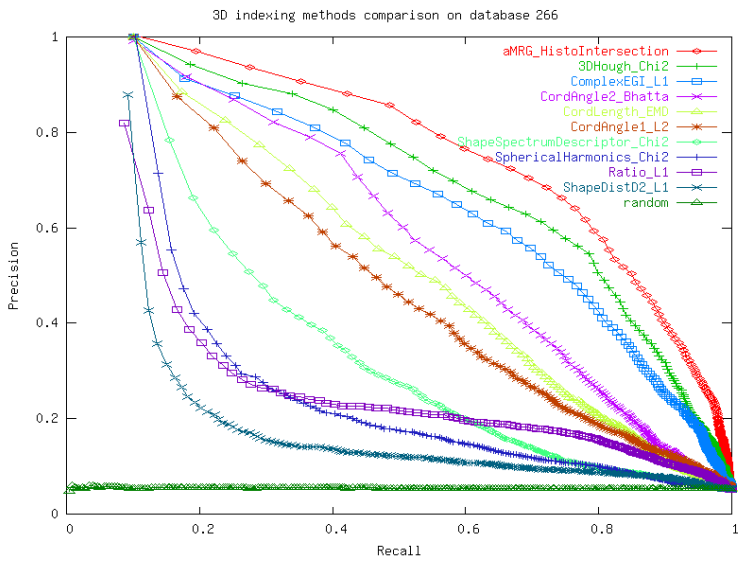

Fig. 20. Precision-recall curves of the tested 3D indexing methods on the second database of 266 objects. aMRG returns the best performance.

that all the other ones but remains below the aMRG performances.

Table 5. Score values (T_1, BEP, E measure and DCG) on database 3 (567 models) for the best combinations of distances and implemented methods.

	Shape dist D2	Sph harmo	Vol area ratio	Shape spec	Cord length	CEGI	3D Hough desc	aMRG $r = 3$
	L1	χ^2	L1	χ^2	EMD	Jeffreys	χ^2	Histo int
T_1	0.201	0.249	0.243	0.332	0.467	0.526	0.640	**0.699**
BEP	0.248	0.314	0.324	0.415	0.568	0.657	0.752	**0.820**
E	0.114	0.155	0.158	0.194	0.268	0.318	0.352	**0.367**
DCG	0.500	0.538	0.574	0.599	0.702	0.746	0.813	**0.841**

Our results are quite encouraging as the parameters were not well optimized. A better choice of parameters should return better results. In addition, shape matching using graph techniques seems to be unadapted for compact objects such as spheres or moulds, as their graph representations are not very significant. However all our experiments on the three databases reveal the ability of the aMRG to characterize this kind of shape very well.

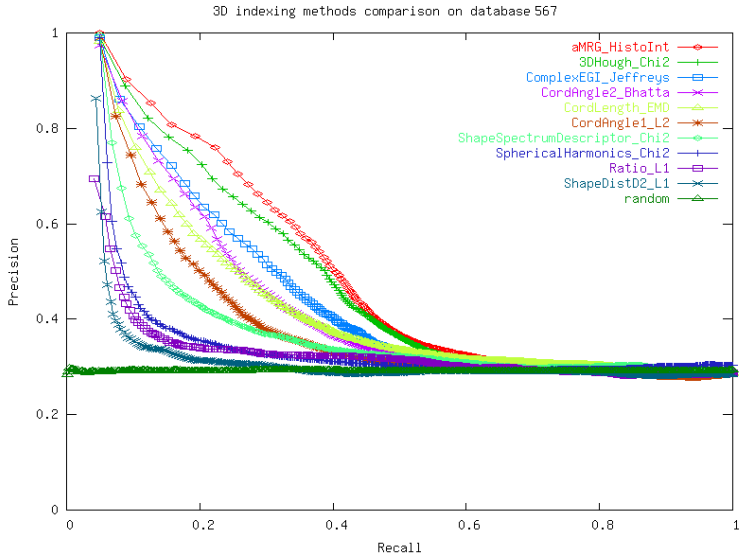

Fig. 21. Precision-recall curves of the tested 3D indexing methods on the third database of 567 objects. aMRG returns the best performance.

7. Conclusion

With the proposed aMRG method we have augmented the MRG approach [1] by adding new characteristics to the nodes of the graphs, and by extending the topological consistency criteria of the graph matching procedure, and by adapting the similarity calculation to the new features. This method results in a flexible multiresolution and multicriteria 3D shape descriptor where are fused topologic, geometric and colorimetric properties. Our experiments on three databases of different sizes and classes reveal the efficiency of the aMRG approach. The analysis of the experimental results by using precision-recall curves, distance matrices, tier images, and various quantitative statistics (BEP, DCG, etc.) allows us to point out the ability and the accuracy of the aMRG to retrieve similar shapes and similar classes. The aMRG approach shows very interesting capacity and seems to be a very promising tool. During the tests, we have chosen heuristics values for the set of parameters λ_i. It appears that these values can be indeed optimized to gain better results. Thus as perspective work we are interested by applying optimization techniques to adjust the parameters for various classified databases and to propose to users different default sets of values. Another perspective of research is to improve the robustness of the aMRG calculation, as at the moment it requires that meshes are manifold, composed of a single connected component, and with low noise on the vertex position for curvature estimation. In addition, characterization using color is still to be explored. Here the difficulty lies in populating a database of textured

objects.

Acknowledgements

This work has been supported by the SCULPTEUR European project IST-2001-35372. The authors would like to thank Pierre-Alexandre Pont for having coded the original algorithm [1] and for the additional topological criterion, and Simon Goodall for his help in optimizing the code.

References

1. M. Hilaga, Y. Shinagawa, T. Kohmura, T.L. Kunii. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes, *ACM SIGGRAPH*, Los Angeles, CA, USA (Aug. 2001) pp. 203-212.
2. T.Tung, F.Schmitt. Augmented Reeb Graphs for Content-based Retrieval of 3D Mesh Models, *International Conference on Shape Modeling and Applications (SMI'04)*, IEEE Computer Society Press, Genova, Italy (Jun. 2004) pp. 157-166.
3. J.W.H. Tangelder, R.C.Veltkamp. A Survey of Content Based 3D Shape Retrieval Methods, *International Conference on Shape Modeling and Applications (SMI'04)*, IEEE Computer Society Press, Genova, Italy (Jun. 2004) pp. 145-156.
4. P. Shilane, P. Min, M. Kazhdan, T. Funkhouser. The Princeton Shape Benchmark, *International Conference on Shape Modeling and Applications (SMI'04)*, IEEE Computer Society Press, Genova, Italy (Jun. 2004) pp. 167-178.
5. S. Goodall, P. Lewis, K. Martinez, P.A.S. Sinclair, F. Giorgini, M.J. Addis, M.J. Boniface, C. Lahanier, J. Stevenson. SCULPTEUR: Multimedia Retrieval for Museums, Image and Video Retrieval, *Third International Conference, CIVR 2004*, Dublin, Ireland (Jul. 2004) pp. 638-646. http://www.sculpteurweb.org
6. http://www.3dcafe.com
7. C. Hernández Esteban and F. Schmitt. Silhouette and Stereo Fusion for 3D Object Modeling, *Computer Vision and Image Understanding*, Special issue on "Model-based and image-based 3D Scene Representation for Interactive Visualization", **96(3)** (Dec. 2004) pp. 367-392. http://www.tsi.enst.fr/3dmodels
8. T. Zaharia, F. Prêteux. Indexation de maillages 3D par descripteurs de forme, *RFIA*, Angers, France (Jan. 2002) pp. 48-57.
9. B.K.P. Horn. Extended Gaussian Image, *Proc. of the IEEE*, **72(12)** (1984) pp. 1671-1686.
10. S. B. Kang, K. Ikeuchi. The complex EGI: a new representation for 3D pose determination, *IEEE Trans on PAMI*, **15(7)** (Jul. 1993) pp. 707-721.
11. E. Paquet and M. Rioux. Nefertiti: a Query by Content Software for Three-Dimensional Models Databases Management, *3-D Digital Imaginig and Modeling*, Proc. Int. Conf. on Recent Advances (1997) pp. 345-352.
12. R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape Distributions, *ACM Trans. on Graphics*, **21(4)** (Oct. 2002) pp. 807-832.
13. J.W.H. Tangelder, R.C.Veltkamp. Polyhedral Model Comparison Using Weighted Point Sets. *Shape Modeling International*, Seoul, Korea (May 2003) pp. 119-129.
14. D.V. Vranic, D. Saupe and J.Richter. Tools for 3D-object retrieval: Kahrune-Loeve Transform and spherical harmonics, *IEEE Workshop Multimedia Signal Processing*, Cannes, France (Oct. 2001) pp. 293-298.
15. K. Kazhdan and T. Funkhouser. Harmonic 3D Shape Matching, *SIGGRAPH Technical Sketches* (Jul. 2002) p.191.

16. J.A. Goldak, X. Yu, A. Knight, L. Dong. Constructing discrete medial axis of 3-d objects, *International Journal of Computational Geometry and Applications*, **1(3)** (2001) pp. 327-339.
17. F. Leymarie and B. Kimia. The Shock Scaffold for Representing 3D Shape, *Visual Form*, Springer-Verlag, Lecture Notes in Computer Science, **2059** (2001) pp. 216-229.
18. A. Verroust, F. Lazarus. Extracting Skeletal Curves from 3D Scattered Data, *The Visual Computer*, Springer, **16(1)** (2000) pp. 15-25.
19. H. Sundar, D. Silver, N. Gagvani, S. Dickinson. Skeleton Based Shape Matching and Retrieval, *Shape Modelling International*, Seoul, Korea (May 2003) pp. 130-142.
20. S. Biasotti, S. Marini, M. Mortara, G. Patanè, M. Spagnuolo, B. Falcidieno. 3D Shape Matching through Topological Structures, *11th Discrete Geometry for Computer Imagery conference*, Springer-Verlag, Lecture Notes in Computer Science, **2886** (2003).
21. Y. Shinagawa, T.L. Kunii and Y.L. Kergosien. Surface Coding Based on Morse Theory, *IEEE Computer Graphics and Applications*, **11(5)** (Sep. 1991) pp. 66-78.
22. G. Reeb. Sur les points singuliers dune forme de Pfaff compltement intgrable ou dune fonction numrique [On the Singular Points of a Completely Integrable Pfaff Form or of a Numerical Function], *Comptes Rendus Acad. Sciences Paris*, **222** (1946) pp.847-849.
23. L.R. Foulds. Graph Theory Applications, *Springer Verlag*, New York, (1992).
24. J. Koenderink. Solid Shape, *The MIT Press*, (1990).
25. C. Zhang and T. Chen. Efficient Feature Extraction for 2D/3D Objects in Mesh Representation, *ICIP*, Thessaloniki, Greece (Oct. 2001) pp. 935-938.
26. J. Barros, J. French, W. Martin, P. Kelly, M. Cannon. Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. *Proc. of SPIE*, **2670** (1996) pp. 392-403.
27. A. Del Bimbo. Visual Information Retrieval, *Morgan Kaufman Publishers, Inc*, (2001).
28. Y. Rubner, C. Tomasi, L.J. Guibas. A Metric for Distributions with Applications to Image Databases, *Proceedings of the 1998 IEEE International Conference on Computer Vision*, Bombay, India (Jan. 1998) pp. 59-66.
29. S. Goodall, P. Lewis, K. Martinez. 3-D shape descriptors and distance metrics for content-based artefact retrieval, *Proceedings of Storage and Retrieval Methods and Applications for Multimedia 2005*, San Jose, California, USA (2005) pp. 87-97.