Proceedings of TMCE 2014, May 19–23, 2014, Budapest, Hungary, edited by I. Horváth and Z. Rusák
© Organizing Committee of TMCE 2014, ISBN 978-94-6186-177-1

SHAPE RECOGNITION AND RETRIEVAL METHOD BASED ON DYNAMIC CLUSTERING

Bilal Mokhtari
LE2I laboratory, University of Burgundy, BP 47870, 21078 Dijon France
Dept. of Computer Science, Faculty of science, University of Biskra, BP 145 07000 Biskra Algeria
bilal.mokhtari@u-bourgogne.fr

Kamal Eddine Melkemi
Dept. of Computer Science, Faculty of science, University of Biskra, BP 145 07000 Biskra Algeria
melkemi2002@yahoo.com

Dominique Michelucci
LE2I laboratory, University of Burgundy, BP 47870, 21078 Dijon France
dominique.michelucci@u-bourgogne.fr

Sebti Foufou
Dept. of Computer Science and Engineering, CENG, Qatar University, Doha, Qatar
sfoufou@qu.edu.qa

ABSTRACT

This paper presents a shape matching framework based on a new shape decomposition approach. A new region-based shape descriptor is proposed to compute the best match between given $2D$ or $3D$ shapes. In order to find similar shapes in a database, we first split the interior of each shape into the adequate set of parts, classes, or ellipsoids, then find the corresponding parts between different shapes, and finally compute their similarity.

Essentially, we compute the best shape decomposition into k classes using an improved version of the k-means clustering algorithm without prior fixing of the number of parts. Additionally, we propose a new tool which determines the best ellipsoids packing in order to efficiently represent a shape according to its components.

The shape recognition process compares the optimal ellipsoidal partition of the new shape with the different models of a database and extracts the closest shapes. The performances of our shape matching framework are shown through experiments on various data of MPEG-7 and benchmark databases.

KEYWORDS

Shape matching, shape recognition, shape decomposition, dynamic clustering, content-based shape retrieval, classification.

1. INTRODUCTION

Shape matching and recognition aims at measuring how much objects are similar, finding the closest shapes to a given shape, called a model, or associating an object to the corresponding class of similar objects [36, 38]. However, this process remains long-standing challenge due to the fact that shapes belonging to the same class are not necessarily close to each other, e.g. articulated bodies. Also, distortion, local deformations, and geometric transformation, may add more difficulties.

A crucial step in the shape matching and recognition process is to find a way to describe and to represent shapes by means of extracted features [15, 25]. This is why in our proposed matching framework, we are interested to find the best way to describe features extracted from either boundary information or interior content. Two type of methods are available: contour-based methods, and region-based methods.

The shape-context approach is one of the popular contour-based shape matching approaches. In [4], the matching of two shapes is to compare the corresponding context maps of shapes descriptor. The shape context is among the robust discriminative descriptor, however it is not adapted to articulated shapes. Another contour-based descriptor, called height function, defines each sample point related to the distances of the other sample points on its tangent line [37]. This descriptor is robust to deformations and shows a great invariance to the geometric transformations. Although the contour-based descriptors are widely used, almost all of these descriptors suffer from several drawbacks. One of the most challenging is that they use a small set (e.g sampled contour points) of shape information. They also require a point-to-point matching which increases computational complexity.

In region-based descriptors, all points within a shape are taken into account, rather that exploiting only boundary points. We can find methods using skeleton descriptors [2, 27], moments invariants [16], and generic Fourier descriptors [39]. Unlike the contour-based descriptors, region based methods allow to capture part structure of the shape, since they combine information across the whole object. Additionally, they are more robust against noise and distortions, and can be employed to represent disjoint shapes. In this paper, we propose a new region-based approach for 2D and 3D shape matching. Our approach operates on the whole set of shape points and combines shape decomposition, similarity measure and database clustering. The database preprocessing phase consists in: (i) ellipsoidal partitioning of shapes, (ii) classification of shapes into different groups. In the shape matching and retrieval phase, a given shape, called model, is matched with the different shapes of the closest group. We calculate the minimal cost of the perfect matching using the Hangarian method [18].

In the proposed shape retrieval approach, we use the ellipsoid skeleton representation proposed by Banégas et al. [3] to find a tree of ellipsoids from a set of 3D points for a given level of details. We also propose a novel approach based essentially on a new shape decomposition technique using an adequate set of ellipsoids. We compute the best shape partition into k regions improving the dynamic k-means clustering algorithm [10].

Dissimilar to the classic algorithms, our proposed clustering approach finds the best partition of interior pixels or voxels into k different ellipsoids without prior fixing of the number of ellipsoids. In fact, this algorithm optimizes both intra-class and inter-class differences, either by fulfilling the general requirement for ellipsoidal decomposition of a given shape, or by considering the spatial relation between different parts of the shape. So, we determine the best ellipsoidal packing of a shape, by iteratively incrementing the partition class number until convergence.

Computing the similarity between two shapes A and B both decomposed into k ellipsoid representations is equivalent to an optimal matching in a complete bipartite graph with $2k$ vertices. The k vertices of A are linked to the k vertices of B. The cost of an edge A_iB_j is obtained by using the Euclidean distance between the characteristic point of the class A_i and the characteristic point of the class B_j. Indeed, we find the corresponding parts between the two shapes A and B. The cost of a perfect matching is the sum of the costs of its matching edges. The matching evaluation is obtained by calculating the minimal cost. This cost is the distance related to the dissimilarity between the shapes A and B.

We assess the validity and the robustness of the proposed algorithm on the 2D data of MPEG-7 [20] database, and on the benchmark reported in [30] for 3D models.

The rest of the paper is organized as follows. Section 2 presents a brief review of fundamental researches in the field of shape matching and retrieval. Section 3 describes shape decomposition and the reconstruction of the ellipsoids skeleton in details. Section 4 presents shape retrieval and recognition, we present database classification, the characteristic point of a class, shape matching and optimal assignment of classes. In Section 5, we discuss the results of our proposed method used for the classification and the recognition of 2D/3D shapes. Finally, Section 6 draws conclusions.

2. SHAPE MATCHING AND RETRIEVAL APPROACHES

In this section, we provide a brief description of shape matching and retrieval methods, reference [36] surveys the state of the art. A comparative study of shape matching algorithms is undertaken in [35] since matching represents a fundamental operation in the field of content-based retrieval approaches [8, 21, 34]. Shape matching algorithms can be classified in two main categories: (i) local approaches use local characteristics of shapes, (ii) global approaches use global characteristics of shapes.

The local matching methods [33, 34] allow accurate measurement of the similarity using local features extracted from small area of a shape, like sub-curves or sub-segment of boundary shape. Chen et al [7] present a local shape matching method based on the Smith-Waterman algorithm [32] to find similar parts between two shapes using a point-to-point matching. The similarity of shapes is achieved by using only some parts of the shape. Another local method for matching triangular meshes 3D surfaces is presented by Ran Gal and Daniel Cohen-Or [12] where a local surface descriptor is presented as salient features extracted from meshes of the different 3D shapes. This method allows finding similar parts in the same shape, and decreases the complexity since it calculates features of surfaces instead of vertices. Even if local methods are considered as a powerful shape representation and matching, one of their biggest challenges is to define local features in advance, which is not an easy task.

The global matching methods operate on the entire shape. In this case, the entire shape of an object represents significant and essential information to measure the global similarity. These methods have always attracted a lot of researchers [6, 9, 24]. Bernier and Landry [9] found that representing the contour points of the whole shape by means of their distance and angle from the centroid polar coordinates shows a great invariance to translations, rotations, and scaling for matching 2D shapes.

Indeed, most of global methods show more efficiency and robustness when there is no big differences between shapes and no large articulations. This is why in recent years, a lot of matching methods used partitioning techniques to decompose shapes into parts and regions in order to improve classical methods. Shape decomposition is a powerful tool to extract relevant and additional information, which provides more robustness and matching accuracy.

In image processing and shape representation, several segmentation and decomposition approaches for both 2D and 3D objects have been investigated [5, 17, 26]. Various mesh decomposition techniques are used for matching objects in 3D shape retrieval [11, 23, 31].

Our method allows to combine the global and the local aspects, by considering all points of a shape to extract local features corresponding to each part, in order to compute the global similarity of two shapes.

3. SHAPE DECOMPOSITION AND CONSTRUCTION OF ELLIPSOID SKELETON

Shape decomposition consists in representing a shape by its components (regions). This strategy is widely used in the field of data analysis. As a strategy of divide and conquer, decomposition reduces information and simplifies a complex problem to make it more understandable and easier to handle.

In nature, animals and humans are able to identify objects based on their components. We often need to see just one or more significant parts of an object to recognize it, for example non-rigid shapes. In the literature, various decomposition methods have been discussed in [14] where a decomposition algorithm is provided for non-rigid shapes. A 3D mesh decomposition method is proposed in [5]. Another decomposition approach, based on random walks, to partition 3D meshes and volume models is presented in [19]. Several convex decomposition methods have also been proposed [14, 22, 28].

In the next Section, we introduce our novel decomposition approach based on ellipsoidal packing of 2D and 3D shapes. We use a variant of the k-means algorithm to decompose a shape into k best parts. Parts are also called classes, ellipsoids, or clusters. The different ellipsoids of a decomposed shape are obtained by extracting its features. This decomposition into ellipsoids is a crucial step in the proposed matching process.

3.1. Dynamic Clustering Algorithm 1

The dynamic clustering algorithm or k-means [10] partitions a data set into k disjoint classes. The aim is to classify data into different groups of similar elements.

The k-means algorithm is widely used in various fields that require data classification. In our method, we use a dynamic clustering in order to partition a 2D or 3D shape into k different parts (classes).

For a given number of classes k, the dynamic clustering algorithm (see algorithm 1) can be written as follows: Start from some partition into k classes (see below for the choice of a good initial partition). Then, assign each point to the class with the closest gravity center. The gravity center $\bar{c_l} = (\bar{x_l}, \bar{y_l}, \bar{z_l})$ of the class c_l is:

$$\bar{c_l} = \frac{1}{n_l} \sum_{P_j \in c_l} P_j = \frac{1}{n_l} (\sum_{P_j \in c_l} x_j, \sum_{P_j \in c_l} y_j, \sum_{P_j \in c_l} z_j)$$

where n_l is the number of points in the class c_l. The distance between a point and the class c_l is the Euclidean distance between the point and the gravity center $\bar{c_l}$ of the class c_l. Once each point is assigned to its closest class, the gravity centers of the k classes are re-computed. This process is iterated until convergence, *i.e.,* when gravity centers are stable up to some ϵ threshold.

```
Algorithm 1 k-means($P, k, C$)
  Input: N points $P_i(x_i, y_i, z_i)$, $k$ centers, $C = \{c_1, c_2, \dots c_k\}$.
  Compute every $\bar{c_l}$ for $l = 1, \dots k$
  Repeat until convergence:
    for $i = 1, \dots N$:
      Let $c_n$ = argmin$_l^k(distance(P_i, \bar{c_l}))$ in
      Assign($P_i, c_n$)
    Update cluster centers $\bar{c_l}$ for $l = 1, \dots k$.
  Output: A set of $k$-classes $c_1, \dots c_k$.
```

Figure 1 shows an example of shape decomposition into different classes using Algorithm 1.

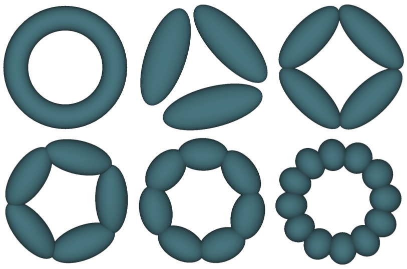

Figure 1 A shape decomposed into 3, 4, 5, 7, 12, classes.

In practice, however it is not possible to know the number of classes k of the optimal partition of points. A random choice of the number k can give poor results. So, to find the best number of classes, we increment k until some criterion is satisfied. Before describing the algorithm we are using, we need initially to introduce certain statistical notions.

In statistics, variance measures how observations are distributed around their mean average. It provides an indication of the dispersion of points (observations). For a set of n points P, where $P_i(x_i, y_i, z_i)$, the variance measures the dispersion of these points regarding their gravity center $\bar{P}$. The Euclidean variance of P, or variance of P for short, is defined as follows:

$$var(P) = \frac{1}{n}\sum_{i=1}^{n}(P_i - \bar{P}) \cdot (P_i - \bar{P}) \tag{1}$$

where n is the number of points in P, $\cdot$ denotes the scalar product, P_i is a point of coordinates (x_i, y_i, z_i), and $\bar{P} = (\bar{x}, \bar{y}, \bar{z})$, the mean of observations (or gravity center) of P, equals:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i, \quad \bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i \tag{2}$$

Thus:

$$var(P) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2 \tag{3}$$

We can rewrite Equation 3 as follows:

$$var(P) = var(X) + var(Y) + var(Z) \tag{4}$$

$$var(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2, \quad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{5}$$

$var(Y, \bar{Y})$ and $var(Z, \bar{Z})$ are defined similarly. Variance measures the homogeneity of the class. The smaller the variance, the greater the homogeneity of the class. The intra-class variance is the average of the variances of the classes of a partition. It is defined by:

$$var\_intra = \frac{1}{k}\sum_{j=1}^{k} var(c_j) \tag{6}$$

In the same way, the average dispersion of the centers of classes regarding the gravity center of the shape is called the inter-class variance and is defined by:

$$var\_inter = \frac{1}{k}\sum_{j=1}^{k}(\bar{c_j} - \bar{s}) \cdot (\bar{c_j} - \bar{s}) \tag{7}$$

where $\bar{c_j}$ is the gravity center of the class c_j, and $\bar{s}$ is the gravity center of the whole shape.

For a given point set P and a given number of classes k, the sum of the inter-class variance and of the inter-class variance is independent of the partition in k classes.

Table 1 and Figure 4 show the typical variation of the intra- and the inter-class variances as a function of k the number of classes.

3.2. A 1D example

Figure 2 shows an example of the classification of a set of integers into various similar groups.

The proposed approach determines the best number of groups according to the best classification measure. Indeed, in this example, we can easily distinguish three groups (right); one group includes the negative numbers, the second contains positive numbers, and the third group contains positive number with greater values. The differences between the numbers in the same group are small; the differences between the numbers in distinct classes are bigger. Based on this example, the best partition should have minimal intra-class variance and, as much as possible, a maximal inter-class variance.

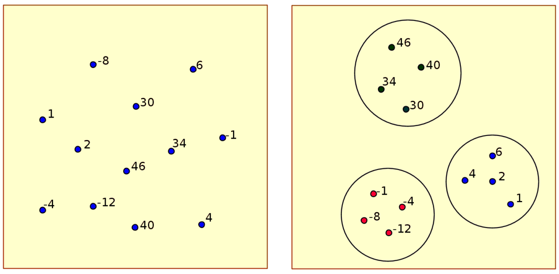

Figure 2 One D classification example. Left: a set of integers. Right: partition into three classes.

3.3. The best number of classes

The previous example illustrates the concavity of the function $var\_inter(k)$, *i.e.,* the concavity of the inter-class variance seen as a function of k, the number of classes. The best number of classes K *i.e.,* the number of classes of the optimal partition [1], is the smallest k such that $var\_inter(k-1) < var\_inter(k)$ and $var\_inter(k) > var\_inter(k+1)$.

Figure 3 shows the different classes found using Algorithm 1 according to k classes. Table 1 contains the different values corresponding to the intra/inter-variances. Again we can check that the intra-class variance decreases when k increases. In this example, the inter-class is maximal for $k = 7$; thus the best number of classes is $k = 7$. Figure 4 shows the variation of values. Figure 5 shows some decomposition results where the best number of class is calculated as stated above. However, this decomposition is not necessarily satisfying. In more complex shapes, certain parts are sometimes not well covered by the union of the ellipsoids of the classes. Before describing our tool, used to decompose parts which are not sufficiently consistent, we need to introduce the notion of ellipsoids.

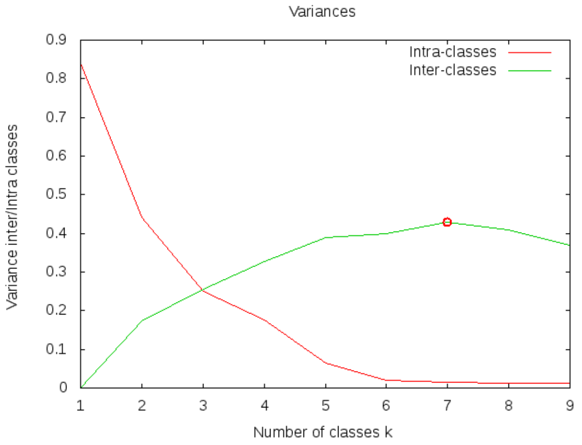

Figure 4 Variation of Intra/inter class variances according to the number k of classes. $var - intra(k)$ is a decreasing function, $var - inter(k)$ is concave.

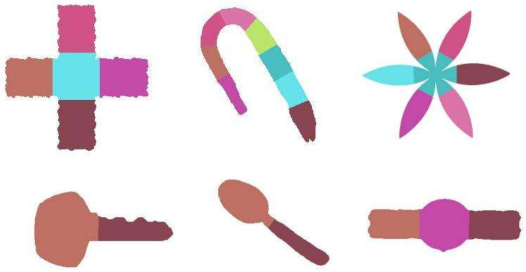

Figure 5 2D shapes decomposed into the adequate k number of classes.

3.4. The covariance matrix and the ellipsoid corresponding to a class

Each class is geometrically represented by an ellipsoid. The center of the ellipsoid is the gravity center of the class, its main axes are the eigenvectors of the covariance matrix of the class, its diameters are the square roots of eigenvalues of the corresponding eigenvectors. Physically speaking, the eigenvectors are the main axis of inertia. This descriptor is invariant with rigid body transformations (rotation, translation), and robust against noise and sampling.

To calculate the ellipsoid corresponding to a class, we need to calculate the covariance matrix M_c:

$$M_c = \begin{pmatrix} var(X) & cov(XY) & cov(XZ) \\ cov(YX) & var(Y) & cov(YZ) \\ cov(ZX) & cov(ZY) & var(Z) \end{pmatrix} \quad (8)$$

$$cov(XY) = cov(YX) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}) \times (y_i - \bar{y}) \quad (9)$$

cov(XZ) and cov(YZ) are defined similarly. The covariance matrix M_c is symmetric and positive definite, so it is diagonalizable. The orthogonalization of the matrix allows us to determine the eigenvalues, and the corresponding eigenvectors. We can summarize this as follows:

$$M_c\vec{v}_\sigma = diag(\vec{\sigma})\vec{v}_\sigma = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} \vec{v}_\sigma \quad (10)$$

where σ_1, σ_2, and σ_3 are the eigenvalues of the covariance matrix, $\vec{v}_{\sigma_i}$ is the eigenvector corresponding to the eigenvalue σ_i. The eigenvalues are real and positive. Figure 6 shows an example of the decomposition of a shape into 5-classes represented by their ellipsoids. The covariance matrix of the whole dataset defines the coordinates system of the shape. The origin is the gravity center of the cloud, and the axes are the eigenvectors of the covariance matrix. All points are described according to this coordinates system.

3.5. The optimal final partition

As mentioned previously, in some cases it is possible that the computation of the best number of classes

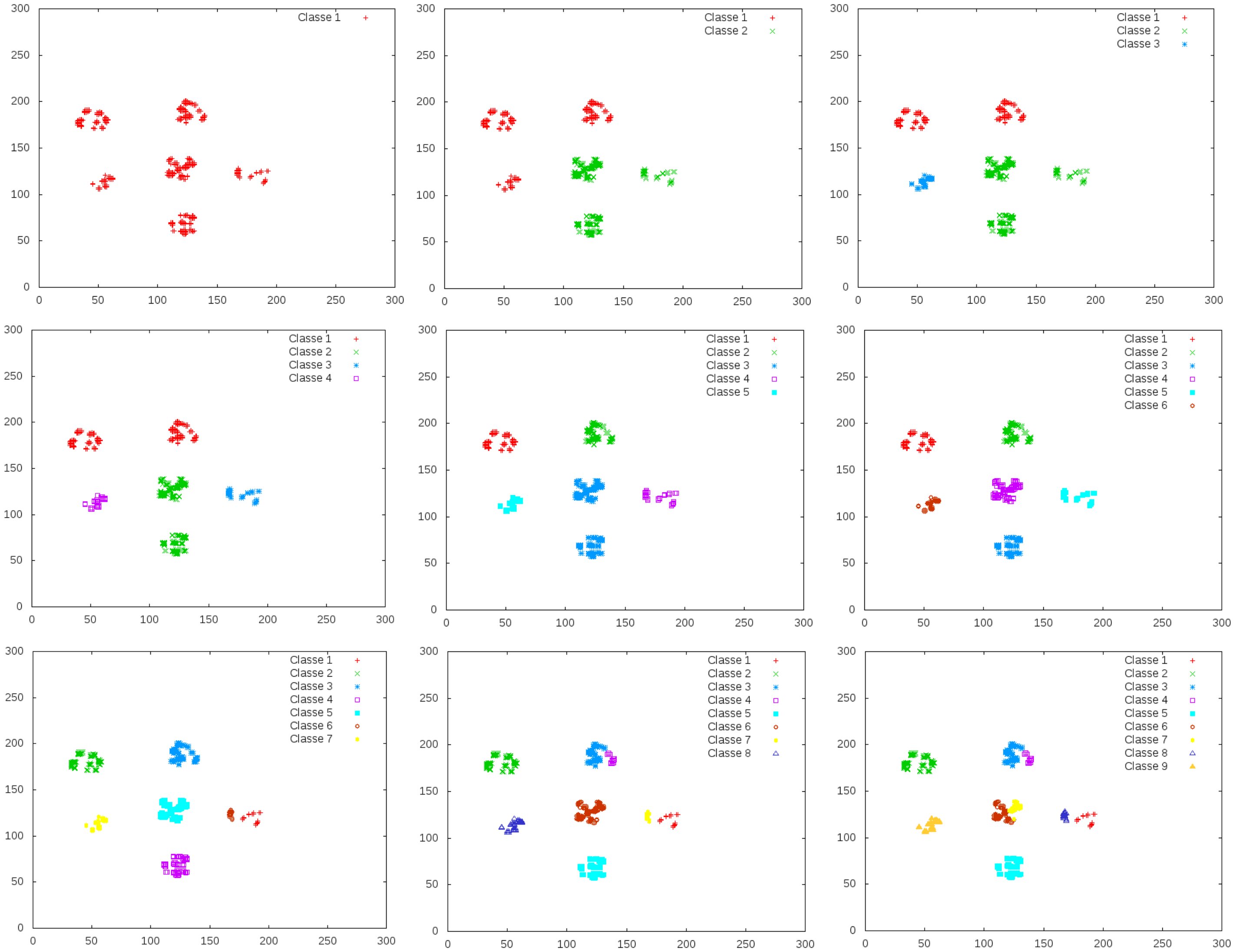

Figure 3 Clustering example using the k-means algorithm. We take the different values corresponding to the intra/inter class variance of the number k of classes.

Table 1 Values corresponding to the intra/inter class variances according to the number k of classes. K is the smallest k such that $var\_inter(k-1) < var\_inter(k)$ and $var\_inter(k) > var\_inter(k+1)$

k	1	2	3	4	5	6	7	8	9
$Intra$	0.84	0.44	0.25	0.17	0.06	0.02	0.015	0.013	0.011
$Inter$	0	0.17	0.25	0.32	0.38	0.39	0.42	0.40	0.36

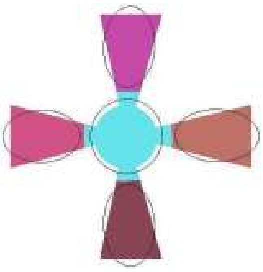

Figure 6 Ellipsoidal skeleton of a shape.

(see Section 3.3) results in classes which are not sufficiently consistent (*e.g.,* Figure 7). Thus, we suggest decomposing these classes into subclasses.

This problem results from the use of the Euclidean distance. Each point is assigned the class with the closest gravity center. No other information is considered. Indeed, using the Euclidean distance produces spherical classes (data partitioned equally

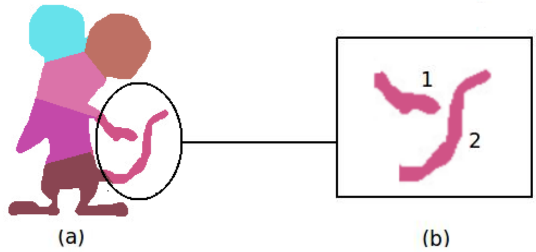

Figure 7 Example of objects with inconsistent classes.

in space). Classes are sometimes unexpected and inconsistent. Thus, we need a new technique.

To fix this issue, we detect inconsistent classes, and then we decompose them into subclasses, using the procedure explained in section 3.3. To determine if a class is sufficiently consistent or not, we calculate an error measuring how well an ellipsoid covers its corresponding class. It is the ratio of data size to ellipsoid axes:

$$Error = \frac{N_i}{M_i} \times 100\% \tag{11}$$

where N_i is the number of pixels or voxels in class c_i (a pixel/voxel is a square/cube with side length 1), and M_i is the ellipsoid area corresponding to class c_i. A class is decomposed into subclasses if the error is greater than 5%.

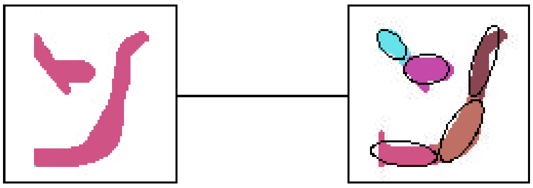

Figure 8 Decomposition of a class into subclasses.

This process assures the treatment of classes which are insufficiently consistent and keeps those which are already consistent. Assuming that we continue to increase the number of classes k, we will decompose all parts, even those which are sufficiently consistent. This method also provides a faster convergence of the algorithm.

Another solution to this issue, which we are currently testing, is to use the geodesic distance instead of the Euclidean distance. The geodesic distance of a point p to a class is the length of the shortest path *inside the shape* from p to the gravity center of the class.

3.6. Initialisation of a k-partition

In addition to the number of classes and the used distance, the initialization of the dynamic clustering is a crucial step. Initial centers influence the convergence of the algorithm. A bad choice of the initial centers will produce undesirable results. It is not easy to find the perfect initial centers. Various methods and algorithms have been presented to solve this issue (for instance starting from many random k centers). We have opted to use the initialization tool described in [13].

Algorithm 2 modified k-means algorithm

```
Input: $P_i = (x_i, y_i, z_i)$, $C = \{c_1, c_2\}$
k-means $(P, 2, C)$
while ($VarInter_{new} > VarInter_{old}$):
    $c_k \leftarrow distant(\bigcup_{i=1}^{k} farthest(C_i))$
    $C \leftarrow C \cup \{c_k\}$
    $k \leftarrow k + 1$
    k-means(k, $P$, $C$)
return k-means(k, $P$, $C$)
```

It is a modified version of the dynamic clustering algorithm.

Definition 1. *Let C be a class, c is the gravity center of this class. A point f in C is called its farthest point (we note $f = farthest(C)$) if this point is the most distant from the center c (using the Euclidean distance). Since there are k classes, $F = \{f_1, f_2, \dots f_k\}$ is the set of all farthest points. We note $distant(F)$ the point in F with the maximal distance from the gravity center of its class.*

Initially, the algorithm starts with two centers to calculate the initial two clusters. These are the two most distant points of the shape (Figure 9.a).

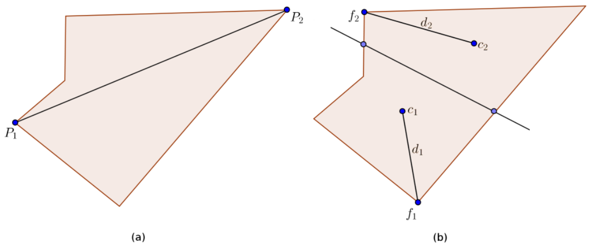

Figure 9 The initial centers for clustering. (a) the two farthest points P_1 and P_2 (the two initial centers). (b) the two farthest points P_1 and P_2 according respectively to centers c_1 and c_2. If $d_1 < d_2$, P_1 is the new center, otherwise P_2 is.

When incrementing the number of classes, and to find a new cluster, we calculate for each existing class C_i, its farthest point $f_i = farthest(C_i)$ (Figure 9.b). Thus, $distant(F)$ is the new center.

This is based on the fact that if a point is far from the center of the cluster, it is dissimilar to the other points of the same class. So, it is possible that it is badly

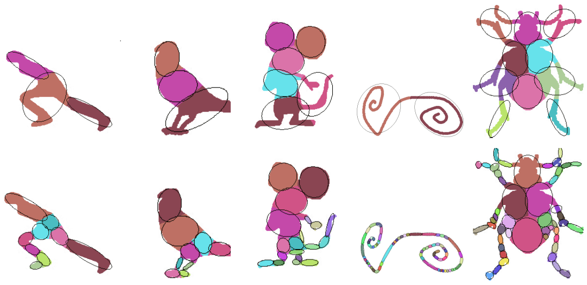

Figure 10 Objects decomposed into adequate number of ellipsoids.

assigned. Figure 10 shows some decomposition results using Algorithm 2. This method gives the appropriate number of k (classes) of a shape. The more detailed a shape, the more ellipsoids it is filled with. In Figure 11 and Figure 12, a best ellipsoid packing is calculated.

4. SHAPE RETRIEVAL AND RECOGNITION

4.1. Dataset classification

The recognition of a given shape S, called a model, implies looking into the dataset to find shapes $S_1, S_2, \ldots S_n$ which are the most similar to it, and compute the differences between the model and these shapes. To manage a large dataset (a dataset containing a big number of shapes), and to accelerate the recognition process, we have developed a tool for indexing the dataset. The idea is to classify elements of the dataset into different groups, according to their number k of classes (ellipsoids).

In a preliminary step, all shapes in the dataset are transformed into ellipsoidal skeletons. All shapes described by their ellipsoids are spread over different groups. We use the algorithm described previously (Section 3.3) to obtain the groups. Represented by discrete data, each shape is decomposed into its number of classes, in order to classify it in the right cluster. So, we get n groups of shapes, where each group contains shapes with similar or close number of classes k. Finally, the model is compared with all the shapes of the nearest group.

4.2. Characteristic vector of a class

To compute the best matching between two shapes, we need to characterize the classes (ellipsoids) describing them. To do so, we associate to each class C_i of a shape a characteristic point (signature). The elements of the class signature are the two or three eigenvalues of the corresponding ellipsoid, and the distance d between the gravity center of the ellipsoid and the gravity center of the shape. Banégas [3] tested some signature variants and concluded that the simplest signature is the best. Thus the characteristic point (signature) of a class is:

$$(vp_1, vp_2, vp_3, d) = (\sqrt{\sigma_1}, \sqrt{\sigma_2}, \sqrt{\sigma_3}, d) \tag{12}$$

where the vp_i are the lengths of the ellipsoids axes E_i (for instance in decreasing order), the σ_i are the eigenvalues of the covariance matrix of the class, and d is the Euclidean distance between the gravity center of the ellipsoid E_i and the gravity center of the shape.

Note that we can use other signatures, such as the orientation of the ellipsoid, or consider only the natural coordinates of the center of the ellipsoid. However, when ellipsoids are close to disks (eigenvalues are almost equal), eigenvectors become unstable. It is why the signature using eigenvalues is more stable and robust than the one using eigenvectors. Bangas already observed that.

4.3. Shape matching and perfect matchings in bipartite graphs

The matching between two shapes A and B, both described by their k ellipsoids $E_1, E_2, \ldots E_k$ and E'_1, E'_2, $\ldots E'_k$, is a minimal matching problem solved using the Hungarian method [18] in strongly polynomial time. It is equivalent to an optimal matching in a complete bipartite graph with $2 \times k$ vertices. The first k vertices represent the ellipsoids of shape A, and the last k vertices represent the ellipsoids of shape B. Each vertex A_i is linked to all vertices B_j, *i.e.*, it is a complete bipartite graph, with $k!$ perfect matchings (or bijections). The cost of the edge A_iB_j is the Euclidean distance between the characteristic point (signature) of the class A_i and the characteristic point (signature) of the class B_j. The cost of a perfect matching is the sum of the costs of the k edges in the matching. Among the $k!$ perfect matchings between the ellipsoids of A and those of B, the optimal perfect matching is the one with minimal cost. This optimal matching not only gives the dissimilarity between A and B, but also matches each ellipsoid of A to the corresponding ellipsoid in B (thus some morphing from A to B can be imagined).

5. EXPERIMENTAL RESULTS

In this section, we discuss the results achieved by the proposed ellipsoid-based shape matching approach. In order to evaluate our shape matching approach, we apply this algorithm to the well-known MPEG-7 [20]

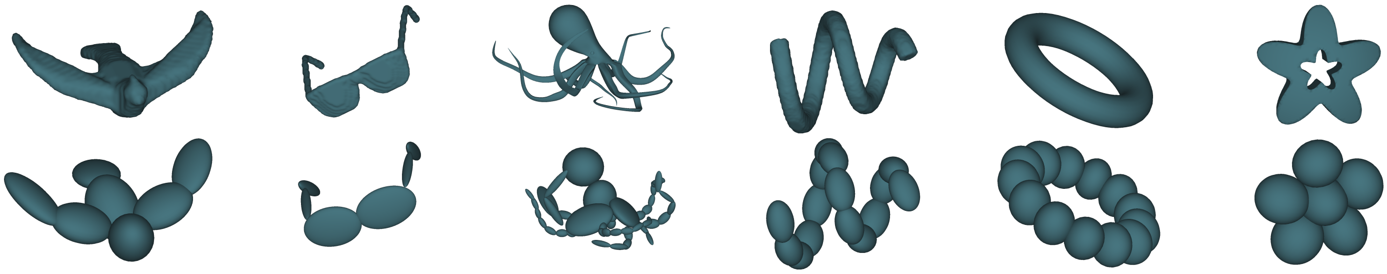

Figure 11 Decomposition examples of 3D models. Line 1: Original models. line 2: 3D Ellipsoidal skeletons.

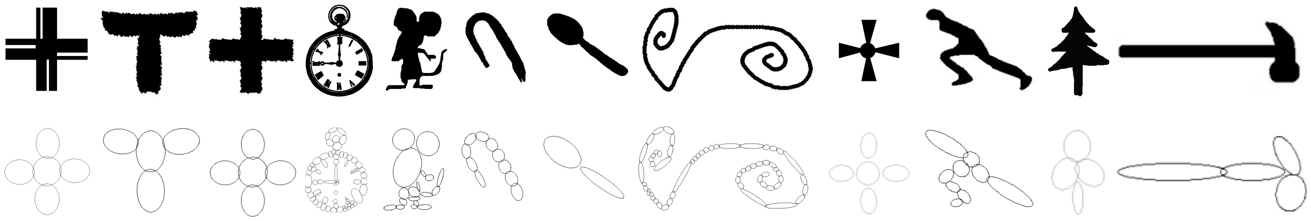

Figure 12 Decomposition examples of 2D shapes. Line 1: Original models. Line 2: Ellipsoidal skeletons.

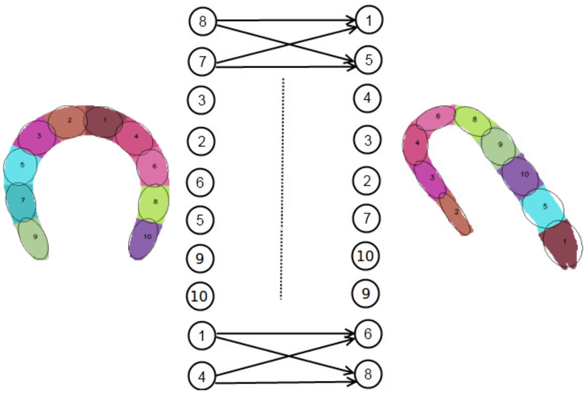

Figure 13 Optimal matching in a bipartite graph (correspondence between two ellipsoidal skeletons.)

dataset for 2D shapes, and also to the benchmark dataset for 3D models [30]. The MPEG-7 dataset contains 70 categories, where each category contains 20 shapes. Described in [30], the benchmark dataset contains 458 3D models, divided into different classes; each class contains a variety of poses.

In a preprocessed phase, all the shapes belonging to a database are transformed into ellipsoidal skeletons using our decomposition algorithm. So, the database is partitioned into a collection of classes or groups containing each one shapes with the same or nearest number of ellipsoids. The performance of the database classification for both 2D and 3D shapes is illustrated in Table 3. The results reported in the table demonstrate that the algorithm shows a great ability to place similar shapes in the same group.

To evaluate the retrieval process, in an on-line phase, a chosen element from the database is used as a query, and compared to all elements of the closest groups. Distance is measured between each shape in the collection and the query shape in order to obtain the most similar shapes. Figure 14 shows some classification results of 3D models. The first element in each line represents a model, and the remaining elements are the corresponding shapes sorted by increasing distance to the model.

The effectiveness of the retrieval engine is shown and reported in Tables 2, 4, 5, and 6 which illustrate the dissimilarity results between different shapes in the MPEG-7 database. We can see that values are significant, meaning that objects are dissimilar. Objects are similar if the distance is small (Table 2). The accuracy of our matching algorithm is measured in terms of correct matches of various 3D and 2D shapes. Tables 7 lists the number of correct matches for different shapes.

However, the performances of our method drop when applied on shapes with a large articulation and big number of parts, this is because we do not consider structural information of shapes. Thus, in some cases decomposition results can be different for the same category of shapes. A solution is to sample the postures (sitting, standing, squatting, walking, etc. for a human being). Another limitation comes from the choice of the initial centers, since it is not obvious to find the best centers which assure the best decomposition of a shape.
We have used C++ language, including OpenGL,

Table 2 Distance matrices between some similar shapes. The values are small, unlike the values corresponding to distances between dissimilar shapes as in Tables 4, 5, 6.

	Rat1	Rat2	Rat3	Rat4	Rat5
Rat	0.014	0.026	0.117	0.059	0.077

	Key1	Key2	Key3	Key4	Key5
Key	0.056	0.085	0.12	0.091	0.034

	Bottle1	Bottle2	Bottle3	Bottle4	Bottle5
Bottle	0.028	0.075	0.042	0.033	0.056

Table 3 Some statistics for partitioning results of the databases. Table shows membership of some category of shapes to a group considering the number of parts (classes) k defining each group..

Database clusters	2D shapes	3D shapes
Group1 ($2 < k < 4$)	Spoons (95%)	Airplanes (97%)
	Children (93%)	Dolphins (90%)
	Bones (87%)	Pliers (95%)
Group2 ($5 < k < 10$)	Devices (96%)	Human (90%)
	Snicks (89%)	Teddy (91%)
Group3 ($10 < k < 20$)	Butterfly(60%)	Ants (75%)
	Octopus (57%)	Crabs (78%)
	Springs (83%)	Dinosaurs (80%)

Table 4 Distance matrix of two different shapes (Fish and Children).

	Fish1	Fish2	Fish3	Fish4
Childr1	0.406	0.448	0.425	0.497
Childr2	0.439	0.456	0.412	0.446
Childr3	0.482	0.437	0.492	0.465
Childr4	0.497	0.454	0.463	0.431

Table 5 Distance matrix of two different shapes (Bone and Bottle).

	Bone1	Bone1	Bone1	Bone4
Bottle1	0.341	0.368	0.335	0.319
Bottle2	0.322	0.394	0.367	0.331
Bottle3	0.383	0.328	0.372	0.324
Bottle4	0.337	0.341	0.382	0.353

Table 6 Distance matrix of two different shapes (Key and Butterfly).

	Keys1	Keys1	Keys1	Keys4
Butter1	0.637	0.591	0.603	0.654
Butter2	0.622	0.637	0.599	0.601
Butter3	0.642	0.621	0.598	0.615
Butter4	0.626	0.597	0.681	0.663

Table 7 Retrieval precision for 2D and 3D shapes.

2D Shapes	Accuracy	3D shapes	Accuracy
Whole	81%	Whole	85%
Bone	87%	Airplanes	96%
Devices	90%	Ants	78.5%
Beetle	78%	Hand	75%
Spring	82 %	Pliers	95%
Spoon	93%	Cups	80%

OpenCV, and GSL libraries for developing our application to test the presented method. Experiments were performed on a laptop (Intel®Core(TM)2 Duo CPU T5470 1.60GHz 2GB RAM) running under Debian GNU/Linux 6.0.

6. CONCLUSION

In this paper, we have described a new region-based shape matching approach for 2D and 3D shapes. Indeed, we developed a new dynamic clustering algorithm.

Our method was able to compute the best ellipsoid packing of a shape without worrying about fixing in advance the number of classes, which is determined in a non-supervised process according to shape complexity.

It seems to us that the use of the ellipsoid representation contributes to provide a robust shape descriptor, and allows to ensure the accuracy of the matching.

However, there is a limitation in our algorithm, it is specifically for non-rigid shapes, when in some cases we are not able to capture the same parts structure for the same category of shapes. We also look to use a graph-based decomposition method to replace the Euclidean distance with an approximate geodesic

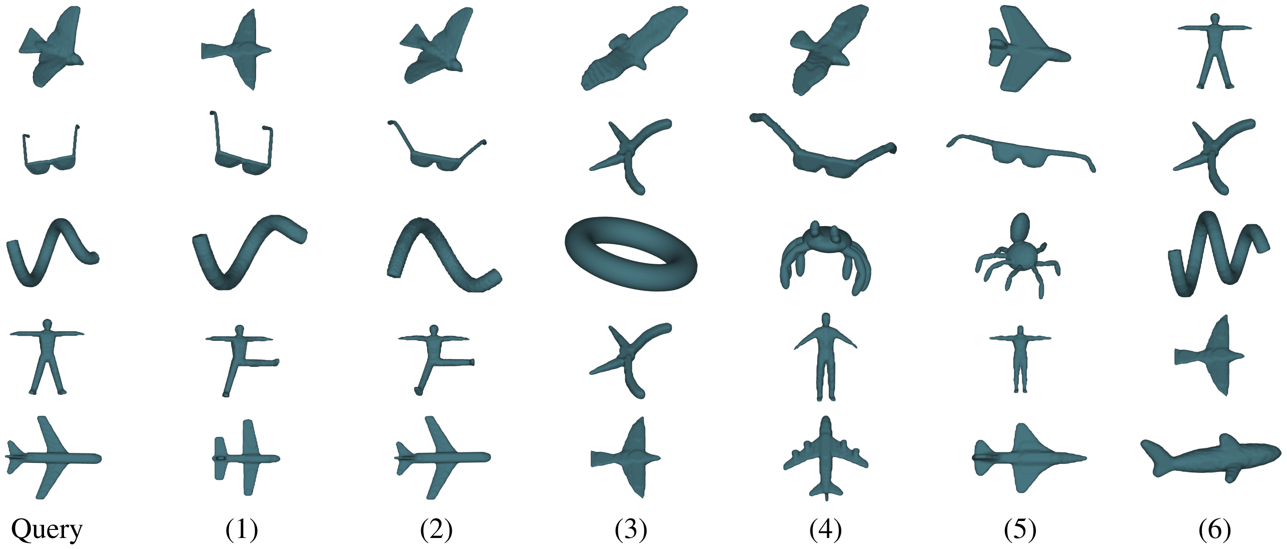

Figure 14 Classification results. Column 1: new shape (model). column i: Corresponding shapes sorted according to their distances from the model.

distance, *i.e.,* the length of the shortest path between vertices, in order to calculate clusters; graph clustering is a well studied topic [29]. Future work will also cover the application of our method for real world data and for various applications such as object detection and tracking, quality control, content-based shape retrieval, medical imaging, shape repository, model retrieval in CAD/CAM or robotics.

ACKNOWLEDGMENT

We would like to thank Mrs. Bouley Mary for helping us to improve this manuscript. This publication was made possible by NPRP grant #09-906-1-137 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

[1] K Hareesha Ahamed Shafeeq. Dynamic clustering of data with modified k-means algorithm. In *International Proceedings of Computer Science & Information Tech*, volume 27, page 221, 2012.

[2] Xiang Bai and L.J. Latecki. Path similarity skeleton graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1282–1292, 2008.

[3] Frederic Banégas, Marc Jaeger, Dominique Michelucci, and M. Roelens. The ellipsoidal skeleton in medical applications. In *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*, SMA '01, pages 30–38, New York, NY, USA, 2001. ACM.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, 2002.

[5] Halim Benhabiles, Guillaume Lavoué, Jean-Philippe Vandeborre, and Mohamed Daoudi. Learning boundary edges for 3d-mesh segmentation. *Computer Graphics Forum*, 30(8):2170–2182, 2011.

[6] A.C. Berg, T.L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, pages 26–33, 2005.

[7] Longbin Chen, R. Feris, and M. Turk. Efficient partial shape matching using smith-waterman algorithm. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–6, 2008.

[8] Xu Chunjing, Liu Jianzhuang, and Tang Xiaoou. 2d shape matching by contour flexibility. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(1):180–186, January 2009.

[9] Minh-Son Dao and Raffaele de Amicis. A new method for boundary-based shape matching and retrieval. In *Image Processing, 2006 IEEE International Conference on*, pages 1485–1488, 2006.

[10] E. Diday. Une nouvelle méthode en classification automatique et reconnaissance des formes, la méthode des nuées dynamiques. *Revue de Statistique Appliquée*, 19(2):19–33, 1971.

[11] Alfredo Ferreira, Simone Marini, Marco Attene, ManuelJ. Fonseca, Michela Spagnuolo, JoaquimA. Jorge, and Bianca Falcidieno. Thesaurus-based 3d object retrieval with part-in-whole matching. *International Journal of Computer Vision*, 89(2-3):327–347, 2010.

[12] Ran Gal and Daniel Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.*, 25(1):130–150, January 2006.

[13] Z. Guellil and Lynda Zaoui. Proposition d'une solution au problème d'initialisation cas du k-means. In *CIIA*, 2009.

[14] Liu Hairong, Liu Wenyu, and L.J. Latecki. Convex shape decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 97–104, 2010.

[15] Jiaxi Hu and Jing Hua. Salient spectral geometric features for shape matching andretrieval. *The Visual Computer*, 25(5-7):667–675, 2009.

[16] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.

[17] Qi-Xing Huang, Martin Wicke, Bart Adams, and Leonidas J. Guibas. Shape decomposition using modal analysis. *Comput. Graph. Forum*, pages 407–416, 2009.

[18] H.W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, pages 83–97, 1955.

[19] Yu kun Lai, Shi min Hu, Ralph R. Martin, and Paul L. Rosin. Rapid and effective segmentation of 3D models using random walks. *Computer Aided Geometric Design*, pages 665–679, 2009.

[20] Longin Jan Latecki, Rolf Lakmper, and Ulrich Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 424–429, 2000.

[21] Guillaume Lavoué. Combination of bag-of-words descriptors for robust partial shape retrieval. *The Visual Computer*, 28(9):931–942, 2012.

[22] Jyh-Ming Lien and Nancy M. Amato. Approximate convex decomposition of polyhedra and its applications. *Computer Aided Geometric Design*, 25(7):503 – 522, 2008.

[23] Hsueh-Yi Sean Lin, Ja-Chen Lin, and H.-Y.M. Liao. 3d shape retrieval using cognitive psychology-based principles. In *Multimedia, Seventh IEEE International Symposium on*, pages 8 pp.–, 2005.

[24] Lu Lv, Yongjin Liu, Wenqi Zhang, Changhao Jiang, Xiaoyu Chen, Terry K. K. Chang, and Matthew Ming-Fai Yuen. Shape profile matching and its applications. *JSW*, 4(5):413–421, 2009.

[25] S. Manay, D. Cremers, Byung-Woo Hong, A.J. Yezzi, and S. Soatto. Integral invariants for shape matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1602–1618, 2006.

[26] Min Meng, Jiazhi Xia, Jun Luo, and Ying He. Unsupervised co-segmentation for 3d shapes using iterative multi-label optimization. *Computer-Aided Design*, 45(2):312 – 320, 2013. ¡ce:title¿Solid and Physical Modeling 2012¡/ce:title¿.

[27] Xiaoning Qian and Byung-Jun Yoon. Shape matching based on graph alignment using hidden Markov models. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 934–937, 2010.

[28] Zhou Ren, Junsong Yuan, Chunyuan Li, and Wenyu Liu. Minimum near-convex decomposition for robust shape representation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 303–310, Nov 2011.

[29] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[30] Kaleem Siddiqi, Juan Zhang, Diego Macrini, Ali Shokoufandeh, Sylvain Bouix, and Sven Dickinson. Retrieving articulated 3-d models using medial surfaces. *Mach. Vision Appl.*, 19(4):261–275, May 2008.

[31] Ivan Sipiran, Benjamin Bustos, and Tobias Schreck. Data-aware 3d partitioning for generic shape retrieval. *Computers and Graphics*, 37(5):460 – 472, 2013.

[32] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, March 1981.

[33] Hedi Tabia, M. Daoudi, Jean-Philippe Vandeborre, and O. Colot. A new 3d-matching method of nonrigid and partially similar models using curve analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(4):852–858, 2011.

[34] Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Partial 3d shape retrieval by reeb pattern unfolding. *Computer Graphics Forum*, 28(1):41–55, 2009.

[35] R.C. Veltkamp. Shape matching: Similarity measures and algorithms. In *Proceedings of the International Conference on Shape Modeling & Applications*, SMI '01, pages 188–, Washington, DC, USA, 2001. IEEE Computer Society.

[36] Remco C. Veltkamp and Michiel Hagedoorn. State of the art in shape matching. Technical report, PRINCIPLES OF VISUAL INFORMATION RETRIEVAL, 1999.

[37] Junwei Wang, Xiang Bai, Xinge You, Wenyu Liu, and Longin Jan Latecki. Shape matching and classification using height functions. *Pattern Recognition Letters*, 33(2):134 – 143, 2012.

[38] Sen Wang, Yang Wang, Miao Jin, X.D. Gu, and D. Samaras. Conformal geometry and its applications on 3d shape matching, recognition, and stitching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(7):1209–1220, 2007.

[39] Dengsheng Zhang and Guojun Lu. Generic fourier descriptor for shape-based image retrieval. In *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*, volume 1, pages 425–428 vol.1, 2002.