Optimizing Query Perturbations to Enhance Shape Retrieval

Bilal Mokhtari[1], Kamal Eddine Melkemi[2], Dominique Michelucci[3], and Sebti Foufou[3,4]

[1] Laboratory of Applied Mathematics LMA, University of Biskra, BP 145 RP, 07000, Algeria
bilal.mokhtari@univ-biskra.dz
[2] Department of Computer Science, University of Batna 2, 05000, Algeria
[3] Laboratoire d'Informatique de Bourgogne, EA 7534, Université de Bourgogne, BP 47870, 21078, DIJON CEDEX, France
[4] New York University of Abu Dhabi, P.O. Box 129188, Abu Dhabi, UAE

Abstract. 3D Shape retrieval algorithms use shape descriptors to identify shapes in a database that are the most similar to a given key shape, called the query. Many shape descriptors are known but none is perfect. Therefore, the common approach in building 3D Shape retrieval tools is to combine several descriptors with some fusion rule. This article proposes an orthogonal approach. The query is improved with a Genetic Algorithm. The latter makes evolve a population of perturbed copies of the query, called clones. The best clone is the closest to its closest shapes in the database, for a given shape descriptor. Experimental results show that improving the query also improves the precision and completeness of shape retrieval output. This article shows evidence for several shape descriptors. Moreover, the method is simple and massively parallel.

Keywords: Computer vision · 3D Shape matching and recognition · Shape Retrieval · Shape Descriptors · Cloning · Genetic Algorithms.

1 Introduction

Shape Retrieval computes which shapes in a database resemble the most to a given key shape Q, called the query [41]. Shapes are polyhedra with triangular faces. Output should be accurate (no false positive) and complete (no omitted solution). Basically, the shape retrieval algorithm computes off-line a shape descriptor, intuitively a signature or a feature vector, for each shape in the database. They do not depend on queries. It also computes on-line the shape descriptor of the query Q. Each shape descriptor induces a dissimilarity measure, or distance for short. For example, if the shape descriptor is an histogram, the dissimilarity measure can be the Chi-squared distance, the Kullback-Leibler divergence, the Hellinger distance, etc. Then, the algorithm computes this induced distance between Q and each shape in the database. Finally, the algorithm outputs the m (we use $m = 11$) shapes with the smallest dissimilarity to the query Q.

Several shape descriptors have already been proposed in the literature, but none achieves satisfying retrieval results with all kinds of shapes [8, 10, 12, 14, 19, 20]. The classical approach to solve this issue is to combine several shape descriptors using some fusion rules [1, 4, 6, 25, 27].

This article proposes to solve the problem by improving the query shape itself. Our approach is therefore orthogonal to the classical approaches, which use only one query at a time, and (a fusion of) many shape descriptors.

To improve the query, we propose a genetic algorithm (GA) [21, 23, 33, 36] called GA-SR: Genetic Algorithm for 3D Shape Retrieval. GA-SR makes evolve a population of perturbed copies of the query shape. Perturbed copies are called clones. The fittest clone Q^* is the clone the closest to its m closest shapes $M(Q^*, D)$ in the database, for a given shape descriptor and its induced distance D. The m closest shapes to Q are the m closest shapes to the fittest clone.

All shapes in the database, query Q and its clones are (generically non convex) polyhedra with triangular faces. Q and all its clones share the same topology, *i.e.,* the same incidence relations between vertices, edges and faces. The sole difference between Q and any one of its clones is that the 3D coordinates of some vertices of Q are weakly perturbed. The perturbation is small enough, in order for the query and its clones to have similar appearance for the human eye.

Improving the query also improves the precision (no false positive) and completeness (no forgotten solution) of shape retrieval, regardless of the used shape descriptor and its induced dissimilarity measure. This article shows evidences for several shape descriptors: VND (Vertex Normal Descriptor), DMC (Discrete Mean Curvature), LSD (Local Shape Descriptor), and TD (Temperature Distribution).

Shapes in the database are usually classified into several classes or clusters to facilitate the work of classical shape retrieval methods [1, 5, 6]. In opposite, GA-SR does not need to know the class of shapes in the database. This information is only needed for measuring and comparing performances of GA-SR [19, 23, 26, 27, 32, 38, 42].

The rest of this paper is organized as follows. Section 2 presents the background. Section 3 details GA-SR. Section 4 presents experimental results. Section 5 concludes.

2 Background and principles

2.1 Improvement of shape retrieval

Several efforts have already been conducted to improve shape retrieval [37, 43, 50]. Most of improvement methods are based on fusion of shape descriptors and their related dissimilarity measures. Chahooki et al. [6] proposed a method to fuse contour and region-based features for improving the retrieval precision. Akgül et al. proposed a fusion-based learning algorithm [1], which combines dissimilarity measures operating on different shape features. It computes their optimal combination by minimizing the empirical ranking risk criterion. Other fusion methods exist [7, 27].

Improving pre-existing shape descriptors also improves shape retrieval output. For example, Bronstein and Kokkinos [3] present a scale-invariant version of the heat kernel descriptor previously proposed by Sun et al. [42]. Ling and Jacobs [28] aim to make the shape context descriptor by Sun et al. invariant to articulation: they replace the Euclidean distance by the inner (also called geodesic) distance to build a shape context descriptor. Other methods [51] for improving shape retrieval associate the database with a graph whose nodes are the database shapes. Therefore, the distance between shapes is defined as the length of the geodesic path in the graph associated to the database. A learning method permits to improve dissimilarity measure using graph transduction.

The concept of perturbation has been used as a successful strategy to improve many algorithms [15, 16, 22, 29, 46, 49].

For example, Thompson and Flynn [46] extract the iris from an image by finding circular boundaries that approximate the circle surrounding the iris. A perturbation is performed by changing the values of one or more parameters of the method.

Stochastic arithmetic is another field which uses random perturbations to improve the robustness of numerical computations [49].

In Computational Geometry, small random perturbations of geometric data remove all degeneracies such as, in 2D, three collinear points or four co-cyclic points [15, 16, 22]. Perturbation greatly simplifies geometric algorithms, because only a small number of generic cases needs to be considered, while the number of degenerate cases increases exponentially with the geometric dimension of the problem.

In Stochastic Resonance, perturbation enhances the transmission of information and the detection of low signals [11, 40].

In Machine Learning, several works [13, 24] recently showed that noisy computations improve associative memories.

More recently, a face recognition system [30] is enhanced by using landmark perturbation technique that sweeps more landmarks, which improves faces comparison.

Yin et al. [52] establish connections between evolutionary algorithms and stochastic approximations.

In this wake, Vaira and Kurasova [48] use a genetic algorithm based on random insertion heuristics for the vehicle routing problem with constraints.

Ernest et al. [17] use GA and Genetic Fuzzy trees to compute deterministic fuzzy controllers, for autonomous training and control of squadron of unmanned combat aerial vehicles.

GAs have been used for solving complex optimization problems [34, 35]. GAs have been also used as a powerful strategy to improve the precision in Information Retrieval Systems [18] and in Web Retrieved Documents [45].

GAs have been also used in Computer Vision and Graphics for measuring similarity of visual data, and in CBIR (Content-Based Image Retrieval). Syam and Rao [44] propose a GA-based similarity measure for CBIR: the GA integrates distinct image features in order to find images that are most similar to a given query image. Aparna [2] proposes a GA-based CBIR method to merge similarity scores: it computes the adequate weight associated with each similarity measure. Chan and King [7] combine different shape features: a GA computes suitable weights for considered features.

Several fitness functions have been used for information retrieval involving GAs. Thada and Jaglan [45] give a comparative study of similarity coefficients used to find the best fitness function, in order to find the most relevant text documents for a set of given keywords. Fan et al [18] computes the best fitness function with a GA for information retrieval.

2.2 Shape descriptors

Shape descriptor represents an essential ingredient for measuring the similarity of shapes. For a polyhedric shape with vertices V, it consists in calculating a signature for some of its vertices. It can be for all vertices in V, or for a strict subset of V refereed to by feature vertices.

Several researches have been conducted to propose discriminant shape descriptors [8, 10, 12, 14, 20].

We have considered several shape descriptors selected from different categories, such as Vertex Normal-based Descriptor VND [47], Local Shape Descriptor LSD [26], Temperature Distribution TD [19], and a Discrete Mean Curvature DMC [32]. The GA-SR methods based on these descriptors are referred as GA-VND, GA-LSD, GA-TD, and GA-DMC, respectively. We have selected these descriptors for their simplicity and efficiency. GA-SR improves all these shape descriptors, in terms of recall-precision curves. Other descriptors can be used.

The Vertex Normal-based Descriptor (VND) The VND [47] descriptor is simple and fast. It considers the normal vector at vertices. The normal vector $\overrightarrow{N}$ at a vertex v is the average of normal vectors in the 1-star of the vertex:

$$\overrightarrow{N}(v) = \frac{1}{l} \sum \alpha_f \overrightarrow{N_f} \tag{1}$$

where l is the number of faces surrounding the vertex v, and α_f is the ratio area of the face f to the total area of the 1-star. The normal vector $\overrightarrow{N_f}$ of a face f with three points p_1, p_2 and p_3, is given by:

$$\overrightarrow{N_f} = (p_2 - p_1) \times (p_3 - p_1) \tag{2}$$

$p_i = (x_i, y_i, z_i)$, $i = 1, 2, 3$, and $\times$ stands for the cross product. The orientation of $\overrightarrow{N_f}$ does not matter. Let F be the subset of feature vertices $n(F) = 3000$. Then the descriptor VND of a vertex v in F is given by:

$$VND(v) = \frac{\|\overrightarrow{N}(v)\|_2}{\sum_{v' \in F} \|\overrightarrow{N}(v')\|_2} \tag{3}$$

Discrete Mean Curvature (DMC) The Discrete Mean Curvature [32] of a vertex v is given by:

$$DMC(v) = \frac{1}{4} \sum_{i=1}^{d} l_i(\pi - \beta_i) \tag{4}$$

where d is the degree of vertex v, β_i the internal dihedral angle (in radians) between two consecutive faces around the vertex v, and l_i the length of the edge common to those faces.

Local Shape Distribution (LSD) The LSD descriptor [26] extracts n random vertices ($n = 3000$), and characterizes each sample vertex v in terms of Euclidean distances to all other points belonging to its neighborhood. The neighborhood is a spherical region centered at point v. The LSD descriptor associates to each region a histogram of Euclidean distances between the point v and points in its neighborhood.

To compute the similarity between two shapes A and B, a complete bipartite graph g is built as follows: the first set of vertices of g is given by the regions of A, the second set is given by the regions of B. The cost of an edge (a, b) between two regions in g is the Chi-squared distance between the a histogram and the b histogram. By definition, the distance between A and B is the smallest cost of perfect matchings in g. This method does not only compute a distance between two shapes A and B, but it also matches regions in A with regions in B.

The Temperature Distribution (TD) The temperature distribution [42] simulates the heat diffusion process on the surface of a model, which starts at a vertex, and goes through other vertices over time.

The temperature distribution descriptor [19] of a vertex is represented as the average of temperatures measured on all vertices in the surface of the model, after applying a unit heat at that vertex. The average temperature for a vertex v, at heat dissipation time t, is given by:

$$TD(v) = \frac{1}{n-1} \sum_{w, w \neq v} \sum_{i} e^{-\lambda_i t} \, \phi_i(v) \, . \, \phi_i(w) \tag{5}$$

where n is the number of vertices (usually $n \approx 3400$), $t = 50$ is a constant, and λ_i is the i^{th} eigenvalue (sorted in decreasing order) of the Laplacian of the underlying graph of the mesh, and ϕ_i its i^{th} eigenvector. In practice, only few eigenvectors are used, four in our experiments.

The distribution of the average temperature values is then represented by means of a histogram. The distance between two shapes is the L_2-norm computed from their histograms.

TD descriptor is invariant to isometric transformations like pose changes, and robust against noise and geometric textures like bumps. However, TD is improved by GA-SR.

2.3 Shape similarity and statistical distances

There are many statistical distances to calculate dissimilarity between two shapes represented as distributions (histograms): Kullback-Leibler divergence, Hellinger distance, Bhattacharyya distance, Chi-squared distance, L_n norm, etc. In this work, we have used some of these distances to measure the dissimilarity of shapes based on each of the used descriptors. We have used Chi-squared distance [39] for VND, LSD and DMC, and L_2-norm for TD [9], in accordance to their experiments. Note that the number of drawers of histograms of compared shapes is $b \approx \sqrt{n}$ ($b \approx 50$). In the rest of this paper, $D(A, B)$ refers to the distance between two shapes A and B.

3 GA-SR: Genetic Algorithm for Shape Retrieval

3.1 Notations and definitions

All shapes *i.e.,* the query, its clones, and shapes in the database, are polyhedra with triangular faces. A polyhedron is represented with a geometric part V and a topologic part F. V is an array of the 3D coordinates of vertices of the polyhedron: $V_i = (x_i, y_i, z_i) \in \mathbb{R}^3$. Coordinates are floating point numbers. F is an array of triangular faces: $F_k = (a_k \in \mathbb{N}, b_k \in \mathbb{N}, c_k \in \mathbb{N})$, where a_k, b_k, c_k are the indices in array V of the vertices composing face F_k. a_k, b_k, c_k are typically ordered counterclockwise, seen from outside the polyhedron. For convenience, a scaling normalization is applied to all polyhedra, so that the sum of all triangles areas equals one (one square meter, say).

A query and all its clones have the same topologic part F. However, the geometric parts are different. Let $Q = \text{shape}(V, F)$ be the query shape. Let $Q' = \text{shape}(V', F)$ be a clone of Q. The geometric part V' of Q' is defined as:

$$V' := V + P, \quad ||P||_\infty \leq \epsilon, \quad ||P||_0 = \mu = \lceil \rho n \rceil \tag{6}$$

where $P_i = (x_i, y_i, z_i) \in [-\epsilon, \epsilon]^3$ is a perturbation vector, $\epsilon \in \mathbb{R}^+$ the noise threshold, n the number of vertices. P is the unknown of our problem.

$||P||_\infty \leq \epsilon$ is imposed to guarantee the perturbation is small. This constraint is compatible with GA cross-over. Typically, ϵ is between 0.002 (2 millimeters) and 0.06 (6 centimeters). The optimal values of ϵ for VND, TD, DMC, and LSD are respectively 0.0074, 0.0022, 0.0562, and 0,0005.

Moreover, we impose that P is sparse. Let ρ be the probability for a vertex to be ϵ-perturbed. In practice, $\rho = 1/4$. The number of perturbed vertices is $\mu = \lceil \rho n \rceil$, with n the number of vertices. The number of perturbed vertices is the same for all clones of a query. This constraint is sometimes written $||v||_0 = \mu$, where $|.|_0$ is a pseudonorm *i.e.,* $||v||_0$ is the number of non zero coordinates of v.

Clones are not re-normalized. It is assumed that the perturbation size is less than the Least Feature Size of shapes, so perturbations do not introduce self-intersections or other geometric inconsistencies.

Let $M(Q, D)$ or $M(Q)$ be the set of the $m = 11$ shapes in the database which are the closest to Q, according to the dissimilarity measure D.

Let q be a shape, typically a clone of Q or Q itself. Its fitness $f(q)$ or $f(q, D)$ is the averaged distance between q and shapes in $M(q, D)$ defined as:

$$f(q, D) := (1/m) \sum_{b \in M(q,D)} D(q, b) \tag{7}$$

We are looking for the perturbation P such that the clone $q = \text{shape}(V + P, F)$ is the closest to its m most similar shapes in the database, *i.e.,* such that $f(q, D)$ is minimal. For convenience, pose $g(P) := f(\text{shape}(V + P, F), D)$. Then the problem becomes: find the optimal or a good enough perturbation X which minimizes $g(X)$ with $||X||_\infty \leq \epsilon$, $||X||_0 = \mu$:

$$X^* = \text{argmin } g(X), \quad ||X||_\infty \leq \epsilon, \quad ||X||_0 = \mu \tag{8}$$

3.2 Sensitivity to perturbations and discretization artifacts

Shape descriptors are very sensitive to noise, *i.e.,* small random perturbations and artifacts due to discretization. This sensitivity, which can be seen as a shortcoming of shape descriptors, is illustrated in Fig. 1: it shows for several shape descriptors the distance curves between a model David1 and clones of a model David2. David1 and David2 are two statues of David, in different poses. Let V_2, F_2 be the geometry and the faces of David2. Let P_2 be the normalized direction of some perturbation vector : $||P_2||_\infty = 1$ for simplicity. Each curve in Figure 1 shows the curve $d(t) = D(\text{David1}, \text{shape}(V_2 + tP_2, F_2))$, with t sampled in $[0., 0.1]$. t is on the horizontal axis, and $d(t)$ on the vertical axis. $d(0)$ is not zero: it is the distance between David1 and David2. It depends on the used descriptor. $d(t)$ quickly falls below $d(0)$ for tiny values of t in $[0, 0.006]$, then slowly increases until $t = 0.01$ or 0.07 depending on the used shape descriptor, and finally quickly increases. For $t \in (0., 0.01]$ or $(0., 0.07]$ depending on the used shape descriptor, all $d(t)$ are below $d(0)$ for this random perturbation direction P_2. These distance curves are rough or noisy. This is due to discretization artifacts. When it is possible, increasing n, the number of samples or feature vertices, and thus increasing the ratio n/b yield to smoother curves. Anyway, this noise does not jeopardize GA-SR, so it is useless to try to reduce it.

These features can be reproduced and more easily understood in the much simpler context of 1D shapes, see Fig. 2. A 1D shape is a continuous and derivable function from $[0, 1]$ to $[0, 1]$, for convenience. For discretization, the interval $[0, 1]$ is divided into n intervals, with $n = 250$ or 5000 in Fig. 2. Each function f is discretized with a vector F such that $F[i] = f(i/n)$, $i \in [0, n]$. The distance between two 1D shapes f_1 with vector F_1 and f_2 with vector F_2 is the Chi-squared distance between their histograms $H(F_1)$ and $H(F_2)$ with $b = 50$ buckets per histogram (this is the value used in references). The example in Fig. 2 uses 1D shapes $f_1(x) = L(a_1, x)$ and $f_2(x) = L(a_2, x)$, where $L(a, x) = ax(1 - x)$ is the Logistic map, and $a_1 = 0.7$ and $a_2 = 0.75$. Visually, f_1 and f_2, or their respective vectors F_1 and F_2, are very close, but the Chi-squared distance between their histograms is 0.16 or 0.17 (the possible maximal value in 1). Fig. 2 shows that

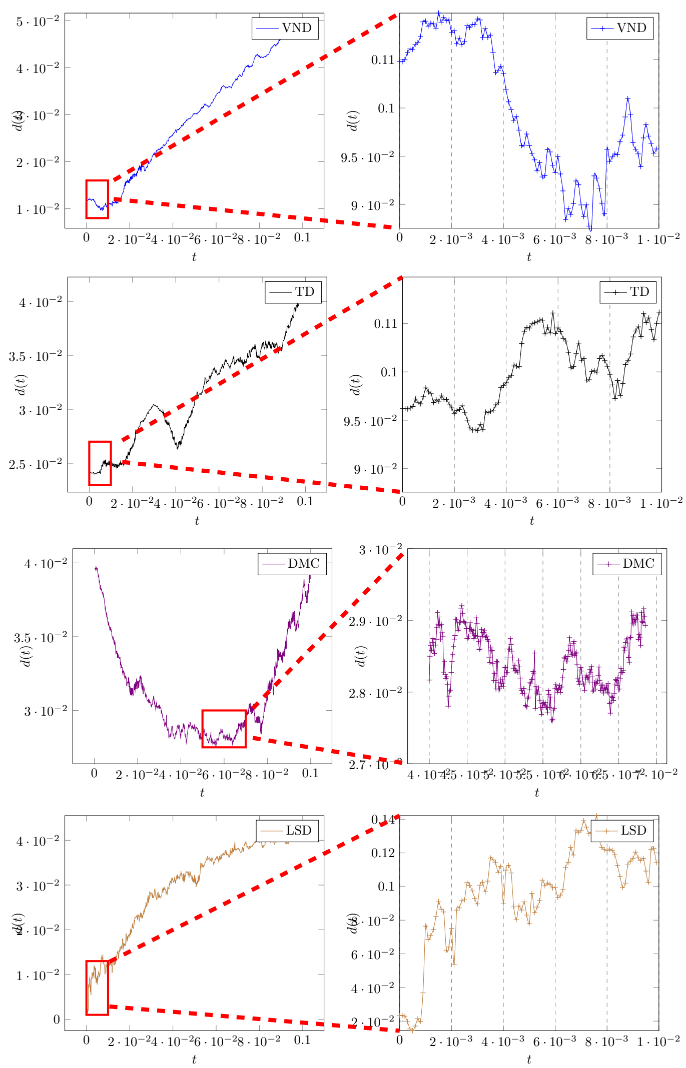

Fig. 1. The impact of perturbation parameter t (horizontal axis) applied on the clones of David2 model, using VND, TD and DMC descriptors in term of distance (vertical axis) to David1 model. Distances are computed between a model David1 and clones of David2. Values of graphs of the left column are picked with a step of 0.0001 in the interval [0,0.1], and those of the right column are picked with the step in the intervals [0,0.01], [0,0.01],[0.04,0.07], and [0,0.01].

clones of F_2 are closer to F_1. Each point (x, y) of a curve in Fig. 2 is $(x = t, y = \chi^2(H(F_1), H(F_2(t))))$ and $F_2(t) = F_2 + tP_2$, where P_2 is a random perturbation vector. Three random perturbation vectors P_2 were tried. For all of them, some clone of F_2 is better than the query F_2, *i.e.,* closer to F_1. With $n = 5000$, curves are smoother than with $n = 250$.

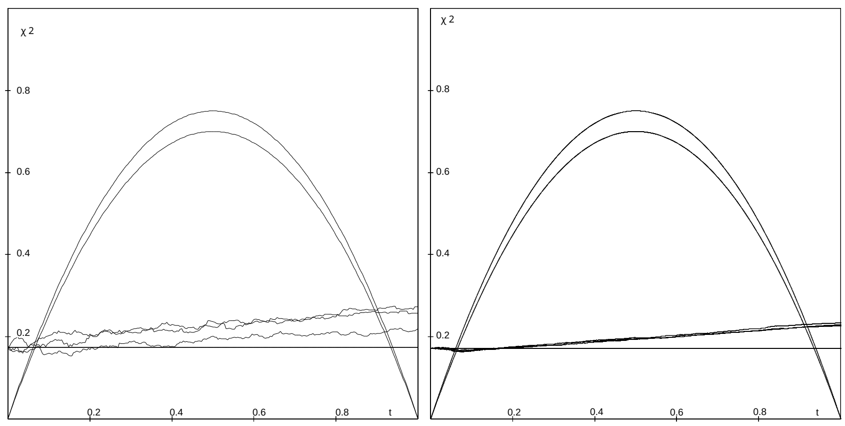

Fig. 2. Distance curves between F_1 and clones of F_2. Left: $n = 250$ samples. Right: $n = 5000$ samples. The height of the horizontal line is 0.17, the Chi-squared distance between F_1 and F_2 histograms. Many clones of F_2 are closer to F_1 than F_2 itself.

3.3 The genetic algorithm

GA-SR is a genetic algorithm. Let Q be the query, and D be the shape descriptor and its induced distance. GA-SR makes evolve a population P of $K = 15$ clones during $G = 20$ iterations, from generation P_0 to P_G. The first population P_0 contains Q and $K - 1$ mutants. Next populations are generated with GA operators: crossover and mutation. Equation (7) defines the fitness of a clone q: the closest to the set $M(q, D)$ of its most similar shapes in the database, the fittest. Figure 3 shows the evolution of the fitness value of the best clone at each generation of the GA. The best solution of the GA corresponds to the minimal fitness value, in this case at the fourteenth generation.

The genotype of a clone is an unsorted array of its μ perturbed vertices: (i, x_i, y_i, z_i), where i is the index of the perturbed vertex, and (x_i, y_i, z_i) the 3D coordinates of the vertex after perturbation. The tuple (i, x_i, y_i, z_i) is called a gene in the GA parlance. It is easy to obtain vertex coordinates of the clone from the vertex coordinates of the query and from the genotype.

Each clone in the first population P_0 is generated with mutations of the query. Let V, F be the geometric and topologic parts of Q. The μ genes of each

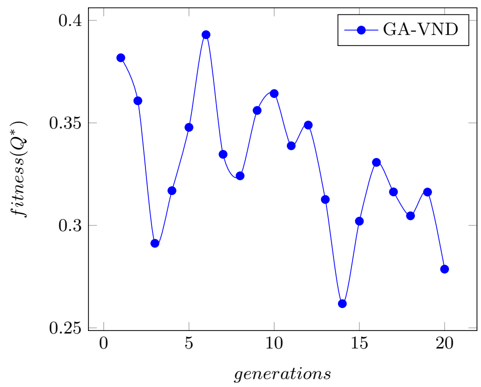

Fig. 3. Evolution of the fitness of the best clone among generations.

clone are generated as follows. μ distinct vertex indices in $1, \ldots n$ are picked at random. Let i be one of these integers. Let $V_i = (x_i, y_i, z_i)$ be the 3D coordinates of vertex V_i of Q. The gene is $(i, x_i + \epsilon\, R(), y_i + \epsilon\, R(), z_i + \epsilon\, R())$ where function $R()$ returns a pseudorandom floating point value uniformly distributed in the interval $[-1, 1]$.

For each population P_g, $g = 0, \ldots G$, the fitness function (see eq. 7) of every clone is computed. To renew the population, standard genetic operators are applied to selected parents to generate new clones: there is no elitism, so the curve in Fig. 3 is not monotonous. More precisely, $K/2$ pairs of clones are selected using the fitness-proportionate selection rule (also called the roulette-wheel selection). Each selected pair generates two new clones with a standard crossover operation between the two genotypes. These two new clones replace their parents in the next generation.

The standard crossover between a first genotype $G_1 = L_1 R_1$ (L for left, R for right) and a second genotype $G_2 = L_2 R_2$ gives two genotypes $L_1 R_2$ and $L_2 R_1$, where lengths of L_1, L_2 are equal. Any classical crossover operator can be used. Some vertices may be perturbed several times, without hindering GA-SR.

Mutation is an important operator in evolutionary algorithms. Each generated clone is subject to a post-mutation: with probability 0.01, each gene (i, x_i, y_i, z_i) is changed to $(j, x_j + \epsilon_x, y_j + \epsilon_y, z_j + \epsilon_z)$, where j is selected randomly and ϵ_x, ϵ_y, ϵ_z are pseudo random values uniformly distributed in $[-\epsilon, \epsilon]$.

4 Experiments

4.1 Databases used

We used the databases TOSCA [53] and SHREC'11 [26]. TOSCA contains 148 3D models (eg. Cats, Centaurs, Dogs, Wolves, Horses, Lions, Gorillas, Sharks, Female and Male figures). The models are distributed into 10 categories including a variety of poses. SHREC'11 contains about 600 non-rigid 3D objects classified into different groups of models, each of which contains approximately the same number of models. In both databases, 3D models are represented as triangular meshes stored in ASCII files in .off format (Object File Format). The name of each file implicitly gives the class (*e.g.,* Cats, Dogs, etc), which permits measurement of performances of retrieval algorithms.

4.2 Tests and results

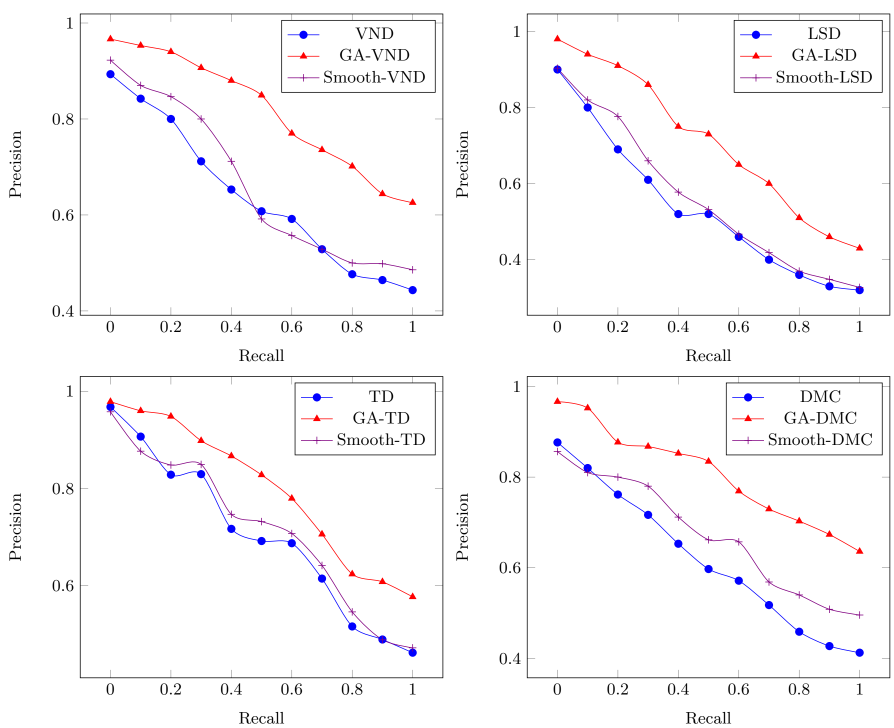

Fig. 4. Averaged 11-point precision-recall curves of random queries of the TOSCA database using original descriptors (blue), Smooth-based descriptors (violet) and the GA-based descriptor (red).

The output quality of shape retrieval algorithms is measured with precision-recall curves. They account both for precision and completeness. They are drawn with the 11-point interpolated average precision algorithm by Manning et al. [31]. It is the reason why we use $m = 11$. The higher the precision-recall curve, the better the retrieval.

Figures 4 for TOSCA and 5 for SHREC11 show the precision-recall curves of descriptors VND, LSD, DMC, and TD compared to their GA counterparts. Clearly, GA-SR significantly improves all these descriptors.

To show the effectiveness of our method, we compare it to the following existing methods: D2 [38], MDS−ZFDR [27], GPS [42], and GT [51]. Comparison

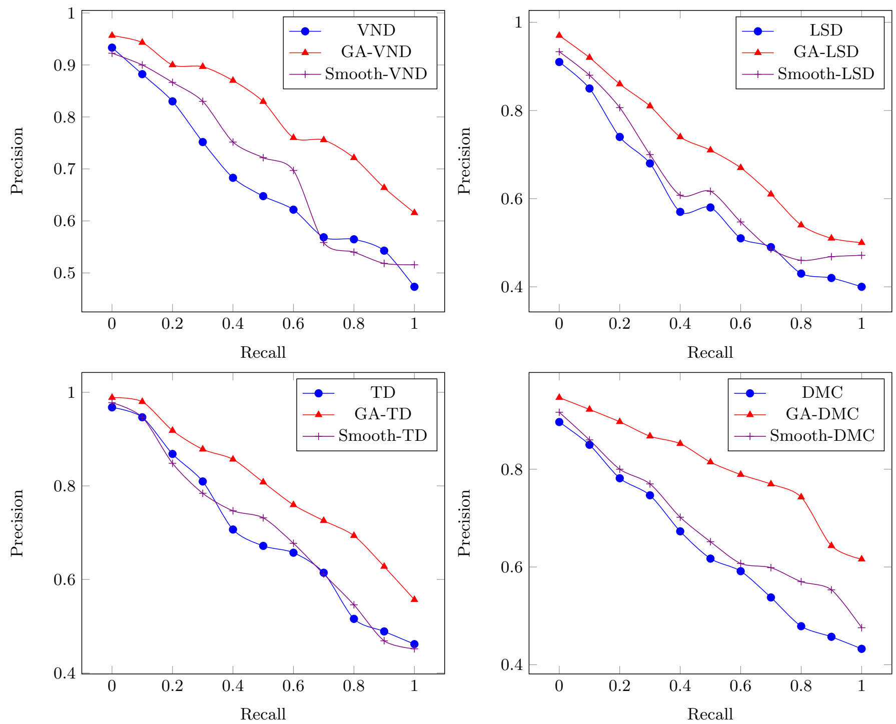

Fig. 5. Averaged 11-point precision-recall curves of random queries of the SHREC'11 database using original descriptors (blue), Smooth-based descriptors (violet) and the GA-based descriptor (red).

results are illustrated in plots of Figure 6. GA-SR shows better performance.

Smoothing of a query shape is another possible way to improve shape retrieval. To smooth a shape, all its vertices are smoothed (without any constraint regarding the order). Let v be a vertex, let g be the barycentre of its neighbors. Then v', the corresponding smoothed vertex, is defined using (9).

$$v' := \text{smooth}(v) := \alpha v + (1-\alpha) g \tag{9}$$

where α is a parameter in $[0,1]$. Then D', the smoothed distance for D in VND, LSD, DMC, TD, is (10):

$$D' := D(\text{smooth}(A), \text{smooth}(B)) \tag{10}$$

where smooth(.) is the smoothing operator. Smoothing reduces noise and irregularities, so intuitively, we expect smoothing to reduce distances: $D'(A,B) \leq$

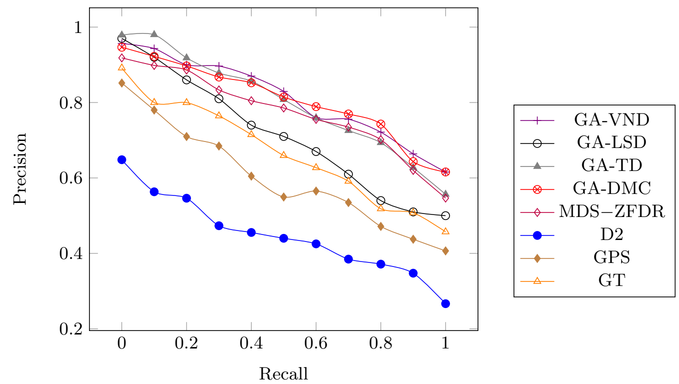

Fig. 6. Comparing results of GA-SR with other methods proposed in the literature. Curves are plotted according to results reported on the SHREC'11 database.

$D(A, B)$. Smoothing is simple and fast, in particular faster than GA-SR. Figures 4 for TOSCA and 5 for SHREC'11 show the precision-recall curves for VND, LSD, DMC, TD, their smoothed counterparts, and their GA counterparts. Clearly, cloning achieves better retrieval results than smoothing.

Finally, GA-SR is compatible with fusion: let $D_1, \ldots D_s$ be s shape descriptors and their induced distances. Then define their fusion distance D with: $D(A, B) := \min(D_1(A, B), \ldots D_s(A, B))$ (or any other fusion rule), and use this distance D with GA-SR. We compared GA-SR and SR with this min-merged shape descriptor, and here too, GA-SR improves SR. No figure is provided for conciseness.

5 Conclusion

Shape descriptors are very sensitive to small perturbations. This shortcoming is also an opportunity for improving shape retrieval. GA-SR achieves better results than previous classical retrieval methods, and better results than smoothing. Other shape descriptors are easily taken into account. GA-SR is simple and massively parallel. It needs no machine learning, no deep learning, no supervision.

References

1. Akgül, C.B., Sankur, B., Yemez, Y., Schmitt, F.: Similarity score fusion by ranking risk minimization for 3d object retrieval. In: Proceedings of the 1st Eurographics conference on 3D Object Retrieval. pp. 41–48. Eurographics Association (2008)
2. Aparna, K.: Retrieval of digital images based on multi-feature similarity using genetic algorithm. International Journal of Engineering Research and Applications (IJERA) **3**(4), 1486–1499 (2013)

3. Bronstein, M.M., Kokkinos, I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 1704–1711. IEEE (2010)
4. Bu, S., Cheng, S., Liu, Z., Han, J.: Multimodal feature fusion for 3d shape recognition and retrieval. MultiMedia, IEEE **21**(4), 38–46 (2014)
5. Carneiro, G., Chan, A.B., Moreno, P.J., Vasconcelos, N.: Supervised learning of semantic classes for image annotation and retrieval. Pattern Analysis and Machine Intelligence, IEEE Transactions on **29**(3), 394–410 (2007)
6. Chahooki, M., Charkari, N.M.: Shape retrieval based on manifold learning by fusion of dissimilarity measures. Image Processing, IET **6**(4), 327–336 (2012)
7. Chan, D.Y.M., King, I.: Genetic algorithm for weights assignment in dissimilarity function for trademark retrieval. In: Visual Information and Information Systems. pp. 557–565. Springer (1999)
8. Chang, M.C., Kimia, B.B.: Measuring 3D shape similarity by graph-based matching of the medial scaffolds. Computer Vision and Image Understanding **115**(5), 707 – 720 (2011), special issue on 3D Imaging and Modelling
9. Chang, S.K., Wong, Y.: Ln norm optimal histogram matching and application to similarity retrieval. Computer Graphics and Image Processing **13**(4), 361–371 (1980)
10. Chao, M.W., Lin, C.H., Chang, C.C., Lee, T.Y.: A graph-based shape matching scheme for 3D articulated objects. Computer Animation and Virtual Worlds **22**(2-3), 295–305 (2011)
11. Chapeau-Blondeau, F., Rousseau, D.: Raising the noise to improve performance in optimal processing. Journal of Statistical Mechanics: Theory and Experiment **2009**(01), P01003 (2009)
12. Chen, D.Y., Ouhyoung, M.: A 3D object retrieval system based on multi-resolution reeb graph. In: Computer Graphics Workshop. pp. 16–20 (2002)
13. Chen, H., Varshney, L.R., Varshney, P.K.: Noise-enhanced information systems. Proceedings of the IEEE **102**(10), 1607–1621 (2014)
14. Coifman, R., Lafon, S., Lee, A., Maggioni, M., Nadler, B., Warner, F., Zucker, S.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. Proceedings of the national academy of sciences **102**(21), 7426–7431 (2005)
15. De Berg, M., Van Kreveld, M., Overmars, M., Schwarzkopf, O.C.: Computational geometry. Springer (2000)
16. Emiris, I.Z., Canny, J.F.: A general approach to removing degeneracies. SIAM Journal on Computing **24**(3), 650–664 (1995)
17. Ernest, N., Cohen, K., Kivelevitch, E., Schumacher, C., Casbeer, D.: Genetic fuzzy trees and their application towards autonomous training and control of a squadron of unmanned combat aerial vehicles. Unmanned Systems **3**(03), 185–204 (2015)
18. Fan, W., Gordon, M.D., Pathak, P.: A generic ranking function discovery framework by genetic programming for information retrieval. Information Processing & Management **40**(4), 587–602 (2004)
19. Fang, Y., Sun, M., Ramani, K.: Temperature distribution descriptor for robust 3d shape retrieval. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on. pp. 9–16. IEEE (2011)
20. Gal, R., Cohen-Or, D.: Salient geometric features for partial shape matching and similarity. ACM Trans. Graph **25**(1), 130–150 (jan 2006)
21. Goldberg, D.E., et al.: Genetic algorithms in search optimization and machine learning, vol. 412. Addison-wesley Reading Menlo Park (1989)

22. Hoffmann, P.H.C.M., Revol, W.L.N. (eds.): Reliable Implementation of Real Number Algorithms: Theory and Practice. Springer (2008)
23. Holland, J.H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press (1975)
24. Karbasi, A., Salavati, A.H., Shokrollahi, A., Varshney, L.R.: Noise facilitation in associative memories of exponential capacity. Neural Computations **26**(11), 2493–2526 (2014)
25. Kittler, J., Hatef, M., Duin, R.P., Matas, J.: On combining classifiers. Pattern Analysis and Machine Intelligence, IEEE Transactions on **20**(3), 226–239 (1998)
26. Li, B., Godil, A., Aono, M., Bai, X., Furuya, T., Li, L., López-Sastre, R.J., Johan, H., Ohbuchi, R., Redondo-Cabrera, C., et al.: Shrec'12 track: Generic 3d shape retrieval. In: 3DOR. pp. 119–126 (2012)
27. Li, B., Godil, A., Johan, H.: Hybrid shape descriptor and meta similarity generation for non-rigid and partial 3d model retrieval. Multimedia tools and applications **72**(2), 1531–1560 (2014)
28. Ling, H., Jacobs, D.W.: Using the inner-distance for classification of articulated shapes. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 2, pp. 719–726. IEEE (2005)
29. Luo, J., Gu, F.: An adaptive niching-based evolutionary algorithm for optimizing multi-modal function. International Journal of Pattern Recognition and Artificial Intelligence **30**(03), 1659007 (2016)
30. Lv, J.J., Cheng, C., Tian, G.D., Zhou, X.D., Zhou, X.: Landmark perturbation-based data augmentation for unconstrained face recognition. Signal Processing: Image Communication **47**, 465–475 (2016)
31. Manning, C.D., Raghavan, P., Schütze, H., et al.: Introduction to information retrieval, vol. 1. Cambridge university press Cambridge (2008)
32. Meyer, M., Desbrun, M., Schröder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. In: Visualization and mathematics III, pp. 35–57. Springer (2003)
33. Miranda, V., Ranito, J., Proenca, L.M.: Genetic algorithms in optimal multistage distribution network planning. IEEE Transactions on Power Systems **9**(4), 1927–1933 (1994)
34. Misevičius, A.: Experiments with hybrid genetic algorithm for the grey pattern problem. Informatica **17**(2), 237–258 (2006)
35. Misevičius, A., Rubliauskas, D.: Testing of hybrid genetic algorithms for structured quadratic assignment problems. Informatica **20**(2), 255–272 (2009)
36. Mitchell, M.: An introduction to genetic algorithms. MIT press (1998)
37. Mohamad, M.S., Deris, S., Illias, R.M.: A hybrid of genetic algorithm and support vector machine for features selection and classification of gene expression microarray. International Journal of Computational Intelligence and Applications **5**(01), 91–107 (2005)
38. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. ACM Transactions on Graphics (TOG) **21**(4), 807–832 (2002)
39. Pele, O., Werman, M.: The quadratic-chi histogram distance family. In: Computer Vision–ECCV 2010, pp. 749–762. Springer (2010)
40. Rousseau, D., Anand, G., Chapeau-Blondeau, F.: Noise-enhanced nonlinear detector to improve signal detection in non-gaussian noise. Signal Processing **86**(11), 3456–3465 (2006)
41. Safar, M.H., Shahabi, C.: Shape analysis and retrieval of multimedia objects, vol. 23. Springer Science & Business Media (2012)

42. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. In: Computer graphics forum. pp. 1383–1392. Wiley Online Library (2009)
43. Super, B.J.: Retrieval from shape databases using chance probability functions and fixed correspondence. International Journal of Pattern Recognition and Artificial Intelligence **20**(08), 1117–1137 (2006)
44. Syam, B., Rao, Y.: An effective similarity measure via genetic algorithm for content based image retrieval with extensive features. International Arab Journal of Information Technology (IAJIT) **10**(2) (2013)
45. Thada, V., Jaglan, V.: Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm. International Journal of Innovations in Engineering and Technology **2**(4) (2013)
46. Thompson, J., Flynn, P.: A segmentation perturbation method for improved iris recognition. In: Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on. pp. 1–8 (Sept 2010)
47. Thürrner, G., Wüthrich, C.A.: Computing vertex normals from polygonal facets. Journal of Graphics Tools **3**(1), 43–46 (1998)
48. Vaira, G., Kurasova, O.: Genetic algorithm for vrp with constraints based on feasible insertion. Informatica **25**(1), 155–184 (2014)
49. Vignes, J.: A stochastic arithmetic for reliable scientific computation. Mathematics and computers in simulation **35**(3), 233–261 (1993)
50. Wong, W.T., Shih, F.Y., Su, T.F.: Shape-based image retrieval using two-level similarity measures. International Journal of Pattern Recognition and Artificial Intelligence **21**(06), 995–1015 (2007)
51. Yang, X., Bai, X., Latecki, L.J., Tu, Z.: Improving shape retrieval by learning graph transduction. In: Computer Vision–ECCV 2008, pp. 788–801. Springer (2008)
52. Yin, G., Rudolph, G., Schwefel, H.P.: Establishing connections between evolutionary algorithms and stochastic approximation. Informatica **6**(1), 93–117 (1995)
53. Young, S., Adelstein, B., Ellis, S.: Calculus of nonrigid surfaces for geometry and texture manipulation. Visualization and Computer Graphics, IEEE Transactions on **13**(5), 902–913 (2007)