Pattern Recognition 43 (2010) 2950–2961

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Parallel relevance feedback for 3D model retrieval based on fast weighted-center particle swarm optimization

Baokun Hu, Yusheng Liu *, Shuming Gao, Rui Sun, Chuhua Xian

State Key Laboratory of CAD&CG, Zhejiang University, 310027, PR China

ARTICLE INFO

Article history:
Received 6 April 2009
Received in revised form
9 February 2010
Accepted 11 February 2010

Keywords:
Relevance feedback
Parallelism
3D model retrieval
Particle swarm optimization
Information retrieval

ABSTRACT

In this study, we present a parallel approach to relevance feedback based on similarity field modification that simultaneously considers all factors affecting the similarity field for 3D model retrieval. First, we present a novel unified mathematical model which formalizes the problem as an optimization problem with multiple objectives and constraints. Secondly, our approach optimizes all the parameters synchronously by treating all the modification operations of the similarity field equally. Thirdly, we improved the standard particle swarm optimization in two different ways. Finally, we present several experiments that show the advantages of our method over existing serial ones.

1. Introduction

According to one estimate, nearly 80% design work of new product is based on reuse of existing examples and design knowledge [1]. Today, with the wide use of computer-aided design (CAD) systems and 3D digital scanning technologies, there are a tremendous number of 3D models in all kinds of libraries. Obviously, reusing rich digital design resources is extremely beneficial for designers, since modeling of 3D objects from scratch is still error-prone and time-consuming. To effectively take advantage of existing reusable design cases, the first task is to discover them quickly and exactly, which is a nontrivial problem. In recent years, various researchers have done a lot of work on this problem and proposed many different solutions [1]. Nevertheless, retrieval results often fall short of users' expectations, because of the wide semantic gap between user-interpretable high-level semantic features and machine-representable low-level values. To narrow the semantic gap, researchers have introduced relevance feedback (RF) to represent users' true intent in the retrieval system. RF has attracted much attention in content-based image retrieval (CBIR) in recent years, and some researchers have used it for 3D model retrieval as well [2–8]. However, it is still an open problem for 3D model retrieval.

In this paper, we propose a new RF approach based on parallel optimization of all the parameters affecting the similarity field of 3D model retrieval. Here, we formulate RF as a multiple-objective optimization problem in which all the parameters are optimized equally. Furthermore, to satisfy RF's requirements of fast response and robustness to limited feedback examples, we present an algorithm called "fast weighted-center particle swarm optimization" (FWCPSO). This algorithm solves RF effectively with self-adaptive particle velocity.

The rest of this paper is organized as follows. Section 2 provides an overview of related work. Section 3 analyzes the deficiencies of existing methods and gives motivation of our work. Section 4 describes our new mathematical model for similarity field modification (SFM)-based RF, which is solved with parallel optimization based on standard particle swarm optimization (PSO). Section 5 discusses how to reduce the dimension of the descriptor and the number of the parameters to be optimized. Section 6 provides two strategies to improve the performance of the typical PSO algorithm, and thus we propose FWCPSO for the specific requirements of RF. Section 7 gives the implementation of our methods and some experimental results with discussion. The last section concludes and discusses future work.

2. Related work

Some researchers have introduced RF technologies for 3D model retrieval to improve unsatisfied query results. These methods can be classified into the following categories:

Query optimization-based methods: Bang et al. presented a feature space wrapping approach [2]. In this approach, all the models in the database move to the relevant or irrelevant models

* Corresponding author.
E-mail addresses: hubaokun@cad.zju.edu.cn (B. Hu), ysliu@cad.zju.edu.cn (Y. Liu), smgao@cad.zju.edu.cn (S. Gao), rsun@cad.zju.edu.cn (R. Sun), xianchuhua@cad.zju.edu.cn (C. Xian).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.02.010

in the feature space through interaction. The movement of each model is the compound effect of the models submitted by the user, and closer relevant models contribute more to movement than do further ones. Atmosukarto et al. adjusted the weights of features to improve the distance measurement of similarity [3]. Their method records models' ranks with multiple integer queues. With these queues, the similarity between the relevant models and the query models is always set to 1, ensuring that the number of relevant objects in the retrieval result increases monotonically with the number of feedback iterations. Papadakis et al. introduced pseudo-RF into 3D model retrieval [4]. This approach is based on the assumption that the m-most similar models are considered relevant and thus used as training data. Then, in the feature space, the feature vector of a model is moved to its cluster centroid. In the method of Biao et al. [5], both the relevant set and the weights of the descriptors are updated during interaction for RF, and the algorithm then optimizes the similarity to refine the retrieval result. This algorithm uses the strengths of different descriptors to enhance feedback result. A new method based on parallel optimization has been introduced into CAD model retrieval by Hu et al. and brought good feedback result in [6].

Classifier-based methods: Elad et al. [7] did the initial work on the application of RF technology to 3D model retrieval for VRML models with the kernel-based RF technique. Their approach trains a support vector machine (SVM)-based classifier on moment descriptors of relevant models and irrelevant models, which are marked by the user in the feedback interaction. This training maximizes the margin between descriptors of the relevant and irrelevant models. Then, the trained SVM classifier divides the models in the library into two subgroups: relevant and irrelevant. Leifman et al. did RF by combining linear discriminant analysis (LDA) with biased discriminant analysis (BDA) [8]. Their method builds a compound classifier with the goal of maximizing the distance between the different classes while simultaneously minimizing the distance within the same class. The compound classifier adaptively takes the form of LDA or BDA according to Fisher's linear discriminant (FLD) criterion.

Recently, some articles have surveyed RF for 3D model retrieval. Novotni [9] compared existing algorithms on the Princeton Shape Benchmark (PSB) [10]. His experiments contain some useful and interesting conclusions: for example, the kernel-based RF approach in [7] outperformed the others.

3. Problems and motivation

For clarity of description, this section begins by defining some basic concepts. We discuss some open problems that often cause RF to fail and, accordingly motivate our RF approach to 3D model retrieval.

3.1. Basic concepts

Definition 1. *A feature vector* is a set of parameters that represents a model. Often, it is denoted by a 1D array $[x_1, x_2,\ldots, x_n]$. Furthermore, each feature vector can be regarded as a point in N dimensional space (U), which is called as *a feature point*.

Definition 2. *Feature similarity*: If the distance D between two feature points is satisfied with the condition $D \le \varepsilon$, where ε is a given threshold, their corresponding models are feature similar. The distance is used to represent the degree of feature similarity. The set of all feature-similar models is called a feature-similar set and written as S_f, as shown in Fig. 1.

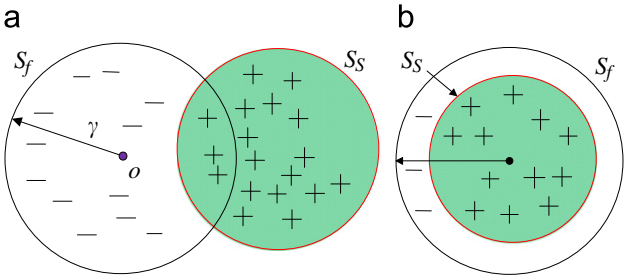

Fig. 1. Illustration of feature-similar set S_s and semantically similar set S_f: (a) normal state of S_f and S_s and (b) ideal state of S_s and S_f.

Definition 3. *Semantic similarity*: If two models are similar according to the user's perception, then they are semantically similar. The set of all semantically similar models is called a semantically similar set and written as S_s, as shown in Fig. 1.

Definition 4. *The similarity field* is the neighborhood of query points in the feature space. Given that q_0 is the query point and p_i is the point of the ith relevant model in space U, then $q_0=q_0(q_{01}, q_{02},\ldots, q_{0N})$, and $p_i=p_i(p_{i1}, p_{i2},\ldots, p_{iN})$. Let vector $v(p_i, q_0)=[(p_{i1}-q_{01}),\ldots, (p_{iM}-q_{0M})]$. Then the feature similarity for the ith model to the query model is $d_i=\|v(p_i, q_0)\|$, where $\|\cdot\|$ is the 2-norm. Suppose that γ ($\gamma>0$) is given a threshold. If $d_i<\gamma$, then p_i is in the γ-similarity field of q_0.

Essentially, computers determine whether models are similar by feature similarity. However, users make their determination by semantic similarity. Generally, the results are different. As shown in Fig. 1, users expect retrieval results to be S_s containing only relevant models. However, what they really get is S_f containing both positive examples (PEs) and negative examples (NEs). For different users, the degrees of difference of feature similarity and semantic similarity are also different. The reason is that different users have different ideas of likeness. It is this difference that leads to unsatisfied retrieval results for 3D models.

Intuitively, if point O is moved to a new position as shown in Fig. 1b, S_f changes consequently to a new state containing more PEs and fewer NEs. The task of RF is to adjust the position, shape and orientation of S_f to maximize the intersection of S_f and S_s as in Fig. 1b. We call this adjustment "similarity field modification" (SFM).

3.2. Problems

3.2.1. Minimization of the distance sum

Rui and Huang set up a model that minimized the sum of the distances between each feature point and query point and drew the conclusion that the optimal center was exactly the arithmetical average of PEs [11]. For convenience, the average distance from PEs to their arithmetic mean is called as average center distance (DA) and denoted by D_c. However, the average arithmetic center is not always the optimal point that maximizes the intersection of S_f and S_s. To demonstrate the conclusion simply, we give a 2D example as in Fig. 2. We give three PEs, represented by A, B and C. Here, the goal is to find a point that minimizes the average distance to points A, B and C. From Napoleon's theorem, we know that the point is exactly *Fermat* point—the red point F in Fig. 2, which is generated by constructing equilateral triangles externally on each side of ΔABC [12]. Here the average distance from PEs to the *Fermat* point is called the "*Fermat* average distance" (FAD) and written as D_f. Usually, the arithmetic mean center point O is different from the *Fermat* point F as shown in Fig. 2.

3.2.2. Multiple objectives

Generally, the user's retrieval intent is complicated and uncertain. It is an inconvenient and difficult task to express much semantic meaning in one objective function, and thus multiple objectives are necessary. As shown in Fig. 3, RF algorithms have two goals. One is to separate PEs and NEs as much as possible, and the other is to make PEs cluster as tightly as possible. When we move the center point to make the average distance of PEs to the new center decrease (where f_c is the average distance of PEs (or NEs) to the new center and p is the feedback round), there are three possible trends for the average distance from the new center point to NEs when feedback round increases from p_1 to p_2: (1) increasing, as shown in Fig. 3a; (2) decreasing more slowly than PEs, as shown in Fig. 3b; (3) decreasing faster than PEs, as shown in Fig. 3c. In the first two cases, NEs move away from the center point. However, NEs move towards the center point much faster than the PEs do in the last case. Obviously, it is difficult to separate PEs and NEs in this case and thus make RF fail. Therefore, it is not easy to directly find out their relationship with feedback examples and thus cause RF to fail if both the goals are represented in one equation. Unfortunately, few RF methods for 3D model retrieval can handle multiple objectives conveniently and robustly [2,3,5].

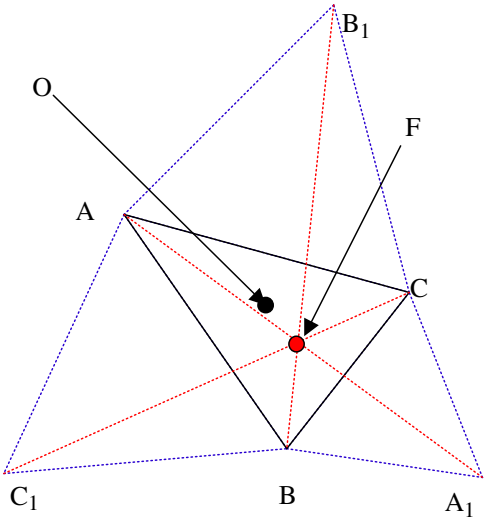

Fig. 2. The relation between *Fermat* and the average arithmetic center.

3.2.3. Serial solution

In traditional query vector modification-based RF methods, three operations can change the similarity field: scaling, translation and rotation. They are done sequentially in random order [13]. In Fig. 4a, the green circle represents the initial similarity field. The scaling, translation and rotation operations are done sequentially to refine the similarity field. The refinement results are represented by the yellow, purple, blue ellipse in Fig. 4a, b, c.

A disadvantage of sequential methods is that the operations are done only once in a specific order. The parameters of the similarity fields cannot be further optimized by the operation after another operation is finished. According to the structure risk minimization rule, minimizing the risk of each step cannot guarantee that the structure risk is minimal [14,15]. As shown in Fig. 4d, doing another scaling operation can yield a better result. In addition, different sequences of the operations may produce different results, since matrix transformation does not follow the commutative law.

In this study, all the parameters of the three operations are considered equally and simultaneously. In fact, these parameters together determine the state of the similarity field, including its position, orientation and shape, which should be adjusted iteratively and isochronously to send the similarity field to its best state so that more PEs and fewer NEs appear in the result, as shown in Fig. 5.

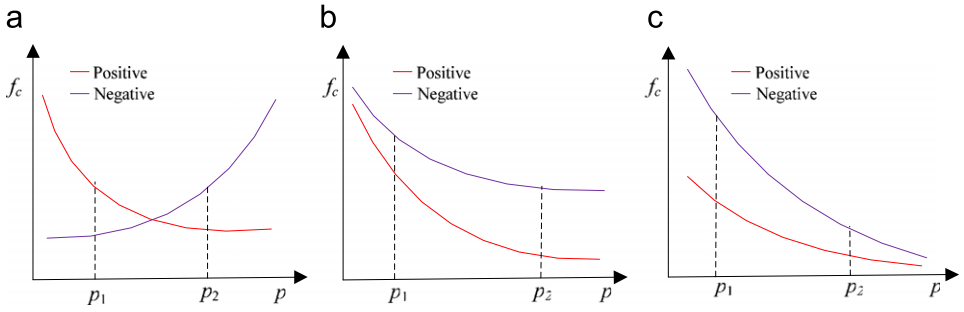

Fig. 3. Possible changing trends of NEs while the RF round number rises: (a) away from the center; (b) towards the center but slower than PEs and (c) towards the center and faster than PEs.

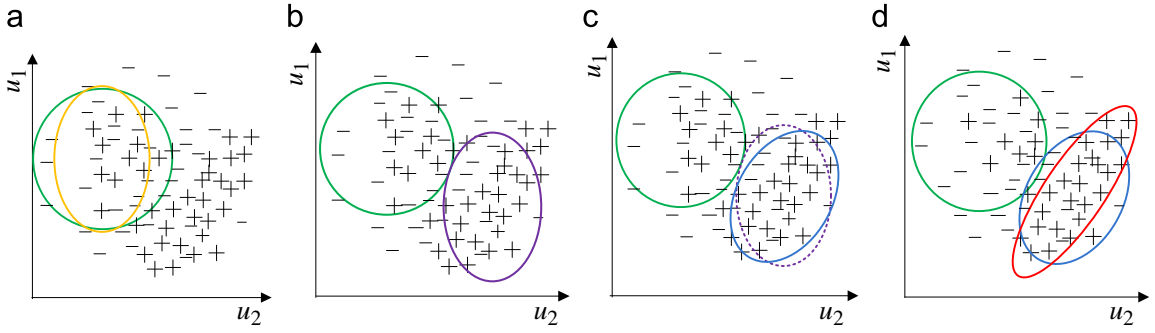

Fig. 4. Operations performed to change the similarity field sequentially and the possible better result: (a) scaling; (b) translation; (c) rotation and (d) possible better result.

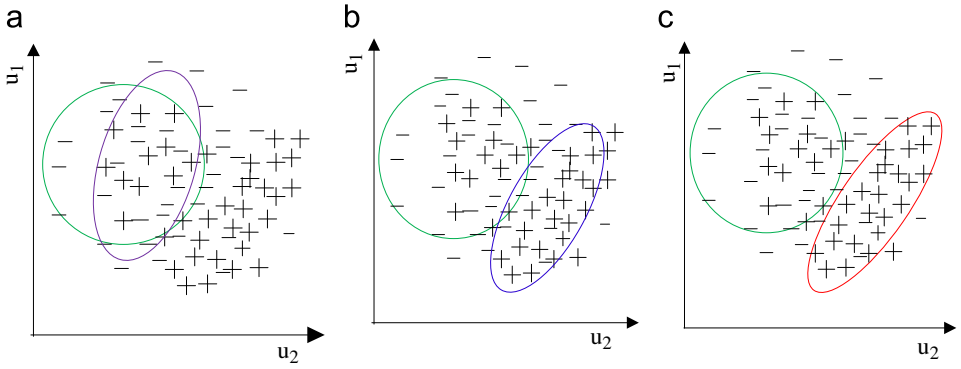

Fig. 5. Adjustment on similarity field through parallel approach: (a) initial state; (b) a better state and (c) a much better state.

4. Mathematical model for parallel RF

Based on the above analysis, we propose a novel SFM-based mathematical model of parallel RF, formalizing it as an optimization problem with multiple objectives and constraints.

4.1. Optimization parameters for parallel RF based on SFM

Generally, the initial similarity field is regarded as a hypersphere that is centered as the query point whose axes' directions are coincident with each sub-vector of the feature vector. Mathematically, the operations that are used to adjust the position, shape and orientation of SFM are represented respectively by the following three matrices, T, S and R:

$$T=\begin{pmatrix}1 & \cdots & 0 & x_1\\ \vdots & \ddots & \vdots & \vdots\\ 0 & \cdots & 1 & x_M\\ 0 & \cdots & 0 & 1\end{pmatrix};\quad S=\begin{pmatrix}w_1 & \cdots & 0 & 0\\ \vdots & \ddots & \vdots & \vdots\\ 0 & \cdots & w_M & 0\\ 0 & \cdots & 0 & 1\end{pmatrix};\quad R=\begin{pmatrix}r_{11} & \cdots & r_{1M} & 0\\ \vdots & \ddots & \vdots & \vdots\\ r_{M1} & \cdots & r_{MM} & 0\\ 0 & \cdots & 0 & 1\end{pmatrix}$$

where T, S, R are translation, scaling and rotation matrices, respectively. Suppose that the center point of the similarity field before modification is q_0 and the center point after modification operations is q_1, $q_1=T\cdot q_0$. For every other point p_i corresponding to the ith model, the distance between p_i and q_1 is given by

$$d_i=\|R\cdot S\cdot v(p_i,T\cdot q_0)\| \tag{1}$$

Therefore, the task of parallel RF is to adjust all the parameters in R, S, and T to find the set of p_i covered by the following similarity field that contains as many PEs as possible:

$$F_s(q_0,R,S,T,\gamma)=\{pi|\ \|R\cdot S\cdot v(p_i,T\cdot q_0)\|<\gamma\} \tag{2}$$

All the parameters in R, S, and T are optimization parameters. Noticeably, they are treated equally and optimized simultaneously no matter which matrix they come from.

4.2. Optimization objectives

Different objectives stand for different aspects of meaning for RF. Here, we select two objectives to improve the quality of RF: one is to maximize the relative distance from PEs to NEs, and another is to minimize the internal dispersion of PEs by minimizing the standard deviation of distances from the center point to PEs.

4.2.1. Relative distance

The importance of relative distance is obvious from Section 3.2.2. Here, the FAD is computed as follows:

$$D=\frac{1}{N_+}\sum_{i=1}^{N_+}d_i=\frac{1}{N_+}\sum_{i=1}^{N_+}\|R\cdot S\cdot v(p_i,T\cdot q_0)\| \tag{3}$$

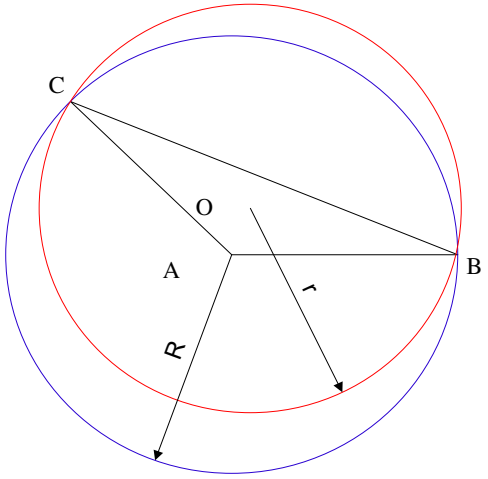

Fig. 6. Small dispersion makes PEs gather together tightly.

Furthermore, the average distance from PEs to NEs is obtained as follows:

$$J=\frac{1}{N_+N_-}\sum_{j=1}^{N_-}\sum_{i=1}^{N_+}\|v(p_i,p_j^-)\| \tag{4}$$

here p_j^- is the jth NE.

Since the ratio of FAD to the average distance from PEs to NEs reflects the distance from PEs to NEs, it is used to represent the relative distance and expressed as follows:

$$F_1=D/J=\frac{N_-\sum_{i=1}^{N_+}\|R\cdot S\cdot v(p_i,T\cdot q_0)\|}{\sum_{j=1}^{N_-}\sum_{i=1}^{N_+}\|v(p_i,p_j^-)\|} \tag{5}$$

Here, N_- and N_+ are the total number of NEs and PEs, respectively. If no NEs are submitted, let $J=1$ and then $F_1=D$; that is, F_1 degenerates from relative distance to absolute distances.

4.2.2. Dispersion of PEs

Generally, PEs in the similarity field are expected to cluster tightly around the center point. As shown in Fig. 6, if the angle BAC is less than 120°, then vertex A is the *Fermat point*. The similarity field (blue circle) centered on point A produces a larger dispersion of points A, B and C than that centered on an inner point O. According to the definition of standard deviation, the objective function describing the dispersion of PEs is defined as

$$F_2=\sqrt{\frac{1}{N_+-1}\sum_{i=1}^{N_+}\left(d_i-\frac{1}{N_+}\sum_{i=1}^{N_+}d_i\right)^2} \tag{6}$$

F_2 is the standard deviation of distances from PEs to the *Fermat point*. It reflects the density of PEs in the neighborhood of the *Fermat point*. A lower F_2 makes PEs cluster together better and leads to fewer NEs covered by the similarity field.

4.3. Parallel optimization model of RF

Based on the above analysis, the optimization model of RF can be formalized as follows:

$$Min\ (F_1(T(x_1,\ldots,x_M), S(w_1,\ldots,w_M), R(\alpha_1,\ldots,\alpha_C)))$$

$$Min\ (F_2(T(x_1,\ldots,x_M), S(w_1,\ldots,w_M), R(\alpha_1,\ldots,\alpha_C)))$$

$$s.t.\quad \begin{cases} \sum_{s=1}^{M} w_s = 1 \\ \alpha_i \in [-\pi/2, \pi/2] \end{cases} \tag{7}$$

where w_i is the weight of the ith component of the descriptor, α_i is the ith angle of the rotation, C is the total number of α_i, and α_i and C are explained in the following paragraph.

4.4. PSO-based parallel solution for RF

It is nontrivial to give an accurate solution to the optimization problem given in Eq. (7). In this study, we try to give an optimization solution based on PSO. PSO has several advantages over other artificial intelligence (AI) algorithms [18,19]. For example, it is easier to understand and implement and it converges better. Furthermore, it is better at global optimization, and it is easier to apply to multiple-objective problems.

Let P_t stand for the current position in the search space and P_{th} stand for the best position found so far for the tth particle. Let V_t be the current velocity, P_g be the best position found by the whole swarm (known as the global best (*gbest*)) up until now, c_1 and c_2 be positive numbers called "cognitive coefficients", r_1 and r_2 be random numbers between (0, 1), and $t=1, 2,\ldots, N$. The velocity is constrained by $[-V_{\max}, V_{\max}]$, P_t is constrained in the search space, and w is the inertia weight decreasing from 0.9 to 0.4. The basic PSO can be described by Eqs. (8) and (9)

$$V_t(i+1) = wV_t(i) + c_1 r_1 (P_{th}(i) - P_t(i)) + c_2 r_2 (P_g(i) - P_t(i)) \tag{8}$$

$$P_t(i+1) = P_t(i) + V_t(i+1) \tag{9}$$

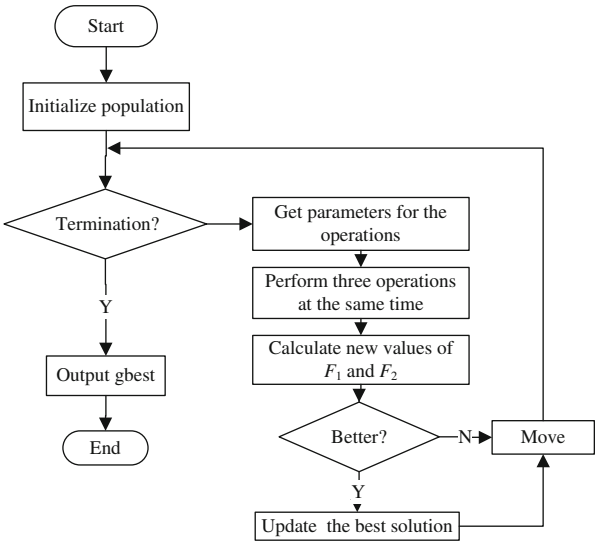

Fig. 7. Flowchart of the proposed method.

When fitness (the error of the objective function) of the objective function is less than the threshold or the maximum iteration time is reached, the algorithm terminates.

To optimize the problem in Eq. (7) and determine the corresponding optimal parameters for RF based on PSO, the first step is to describe the problem using PSO theory and determine the search space. According to Eqs. (7)–(9), the position vector of each particle contains the parameters in matrices T, S and R

$$\boldsymbol{p} = [x_1,\ldots,x_M, w_1,\ldots,w_M, \alpha_1,\ldots,\alpha_C] \tag{10}$$

The whole process of PSO based parallel solution is shown in Fig. 7. During the optimization process, each particle of the swarm marches in the search space with a given velocity and thus moves to a new position, then the operations are performed, and the values of the optimization objectives F_1 and F_2 are calculated. If the new position is better than the past ones, the best solution will be updated and the particles explore the search space furthermore. The optimization process iterates until the termination conditions are satisfied and thus one or more optimal solutions will be found.

5. Refinement of the parameters

One problem of parallel RF is that the number of parameters under consideration is much more than that in the sequential case. The higher the dimensionality is, the larger the searching space and the more time required for PSO. Generally, not all the parameters are equally important in describing the models. For different users, different parameters have different importance, since different users pay attention to different aspects of the models. These differences admit the possibility of reducing some unimportant parameters during the initial retrieval and RF to make the computations more efficient. In this study, we discuss two strategies to reduce the dimension of the parameters. First, most parameters coming from rotation are reduced by the coarse filter called "rotation simplification"; second, a manifold learning method called "local linear embedding" (LLE) is used to remove the parameters that only slightly change the similarity field.

5.1. Rotation simplification

The rotation matrix R in Eq. (7) is a full matrix that has $M*M$ parameters (M is the dimension of feature vectors). All the parameters must be computed from limited feedback examples. In fact, R can be regarded as the concatenation of a series of simple matrices, each of which represents a single rotational transformation in a certain hyperplane.

Let $\boldsymbol{e}=[e_1, e_2 \ldots e_M]$ stand for an orthogonal base of the feature space, and $\boldsymbol{e}_{ij}=[e_i, e_j]$ stand for a hyperplane. The rotational transformation in this plane can be expressed by the upper matrix R' in Fig. 8, where α is the rotational angle. According to the decomposition, the compound rotational transformation can be represented as the multiplication of C ($C=M(M-1)/2$) simple rotational matrices. Each rotates the similarity field by α_i ($i=0,\ldots,C$) in the counter-clockwise direction. Therefore, the

$$R' = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & \cos\alpha & & \sin\alpha & \\ & & & \ddots & & \\ & & -\sin\alpha & & \cos\alpha & \\ & & & & & \ddots \\ & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ i^{th} \\ \\ j^{th} \\ \\ \end{matrix}$$

Fig. 8. The rotation matrix in the plane spanned by axis-i and axis-j.

matrix R can be expressed by Eq. (11), and totally $M(M-1)/2$ rotational angles are needed to determine the rotational matrix in Eq. (10).

$$R = R_C R_{C-1} \cdots R_2 R_1 \tag{11}$$

In addition, considering that there are M parameters in matrices T and S, the total number of parameters in Eq. (10) to be optimized is $M(M+3)/2$.

Furthermore, not all the rotational transformations change the orientation of the similarity field greatly in a specific step. To demonstrate this, a simple situation is given in Fig. 9. Here the similarity field is an ellipsoid with three axes in which the length of axes X and Y are almost the same and both of them are much shorter than axis Z. It can be seen from Fig. 9(b) and (c) that the similarity field will change greatly if it rotates in plane XOZ or YOZ. However, we can hardly make out the change of the similarity field if it rotates around in plane XOY as shown in Fig. 9(d).

5.2. Dimension reduction with LLE

Here, we use the typical manifold learning method known as "local linear embedding" (LLE) [20] to discover and reduce the less important parameters of the position vector $\boldsymbol{p}$ in Eq. (10). LLE is a widely used nonlinear dimensionality reduction method, which tries to preserve the local geometric properties of the data by mapping a point, with its neighbor points on the manifold, into a low-dimensional space. This mapping is based on the assumption that the same linear representation will hold in the low-dimensional space. As shown in Fig. 10, the 2D manifolds are transformed into a plane. Here, the first column of figures is 2D manifolds, the middle column is the sample points and the last of figures is the embedding manifolds in low-dimensional space found by LLE. One can see that geometrical complexity is lowered remarkably.

For the results of the method mentioned in the last subsection, the problem remains of how to apply LLE directly to removing less important components. Here, the components of vector $\boldsymbol{p}$ come from three different operations and have different meanings. Therefore, it is not reasonable to compose the matrix by the vectors such as $\boldsymbol{p}$ directly and then decompose this matrix. The parameters from different operations are not equally important and thus should be considered separately during LLE-based dimensionality reduction. The different importance of different parameters will be discussed in detail in Sections 7.3 and 7.4.

6. Fast weighted-center particle swarm optimization

Although existing PSO methods converge faster than many other heuristic algorithms, it is sometimes unsatisfactory for the high-dimensionality of RF in 3D model retrieval. To further improve the computational efficiency of RF, we present a new PSO method with two important improvements based on center particle swarm optimization (CPSO) [21]: (1) the weighted center of the swarm is determined to replace the arithmetic mean, since different particles make different contributions, which decreases the risk of falling into local extremes; (2) we introduce the environmental perceptive factor to let the particle adjust its velocity adaptively according to the surrounding environment.

6.1. Weighted-center PSO

Currently, all the particles in PSO are under the guidance of *gbest* in the whole swarm or in the individual's neighborhood. However, experiments show that the center of the whole swarm is usually much closer to the real global optimum than *gbest* is [18]. Based on the observation, Yu proposed CPSO by replacing P_g with the center position of the swarm P_c [21] and thus Eq. (8) is adapted to the following Eq. (12):

$$V_t(i+1) = wV_t(i) + c_1 r_1 (P_{th}(i) - P_t(i)) + c_2 r_2 (P_c(i) - P_t(i)) \tag{12}$$

$$P_c = \left(\sum P_t(i)\right)/N \tag{13}$$

Generally, the center of the whole swarm is more stable than P_g, and thus P_c helps the swarm converge. However, a problem still arises from ignoring the difference between the superior particles and the inferior particles. Of the whole swarm, there are some

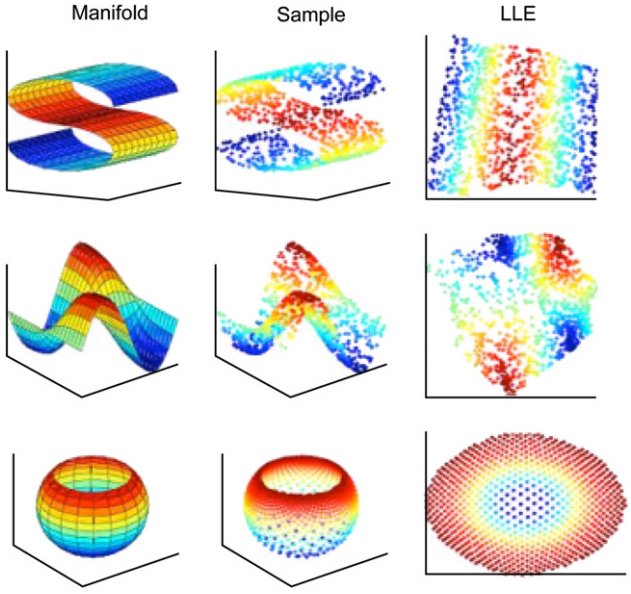

Fig. 10. Dimension reduction with LLE.

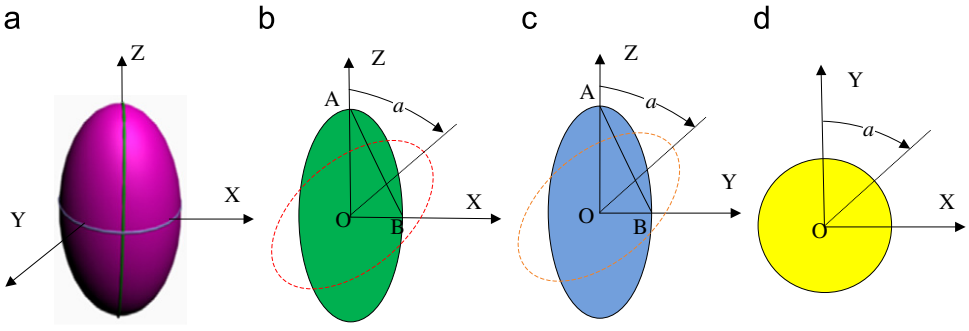

Fig. 9. The different influence of rotations in different planes on the similarity field; (a) the 3D similarity field; (b) the influence of rotation in plane XOZ; (c) the influence of rotation in plane YOZ and (d) the influence of rotation in plane XOY.

particles which have less fitness values and are much closer to the objective value than the others. We call these "good" particles as superior particles and the others as inferior particles. The reason is that P_c is in favor of the inferior particles but is against the superior particles. In this study, the best 33% particles are called as superior particles and the worse 33% particles are called as inferior particles. It is not proper to treat superior and inferior particles equally: superior particles are more important than inferior ones, since they contribute much more than the inferior ones for finding the best position. And for the inferior particles there shall be a stable destination as the superior particles to keep the convergence of the algorithm although they are less important to the optimization. Here we present FWCPSO, which takes advantage of both the stability of the center and the different influences of the superior particles and the inferior particles. FWCPSO uses a new weighted center P_w to substitute the average center P_c in Eq. (12), which is rewritten as Eqs. (14) and (15), in which the superior particles contribute more than the inferior particles.

$$V_t(i+1) = wV_t(i) + c_1 r_1 (P_{th}(i) - P_t(i)) + c_2 r_2 (P_w(i) - P_t(i)) \tag{14}$$

$$P_w = \left(\sum w_i P_t(i)\right)/N \tag{15}$$

It is know that the tth particle in the swarm has a value (or fitness) and denoted as f_t. The calculation process of f_t is given as follows: (1) update the position for the tth particle; (2) get the parameters for translation, rotation and scaling, respectively; (3) put these parameters into F_1 and F_2 to calculate the fitness.

The task at hand is how to assign the weight for each particle properly. It is not proper to take the fitness as the weight directly since the weight corresponds inversely to the fitness: high fitness values correspond to low weights, and low weights correspond to high fitness values. Here, we adopt the following strategies to assign a high weight to a particle with low fitness.

First, the fitness of each particle is normalized by $f_{\max}$ and $f_{\min}$ as Eq. (16), where, f_{nt} stands for the normalized fitness, $f_{\max}$ stands for the maximum fitness and $f_{\min}$ the minimum fitness of the particles in the whole swarm.

$$f_{nt} = \frac{f_t - f_{\min}}{f_{\max} - f_{\min}} \tag{16}$$

Second, the absolute weight of each particle is calculated based on its fitness, as in Eq. (17). Here, we choose the exponential function $\exp(\cdot)$ since experiments demonstrate that it decreases at the proper speed with the normalized fitness increasing.

$$r_i = \exp(-f_t) \tag{17}$$

Third, the weight of the tth particle is calculated as follows:

$$w_t = \frac{r_i}{\sum \|r_i\|} \tag{18}$$

6.2. Environment-perceptive factor

For each particle, its surrounding environment is always changing. Therefore, in order to make the FWCPSO algorithm converge faster, the particle should explore the search space intelligently, at low speed, to start searching for the best position. At the same time, it collects bumpiness information about the surrounding environment. If the fitness seldom changes or just changes a little in the past steps, it may be inferred from experience that the environment nearby is relatively flat. Therefore it speeds up for convergence. On the contrary, if the fitness changes frequently and acutely, the environment nearby must be scraggy, and the particle will slow down to pay more attention to exploration. In other words, the individual particle shall be able to fly faster in a flat area, such as [0, 50] in Fig. 11, and slowly in a rugged area, such as [60, 80] in Fig. 11.

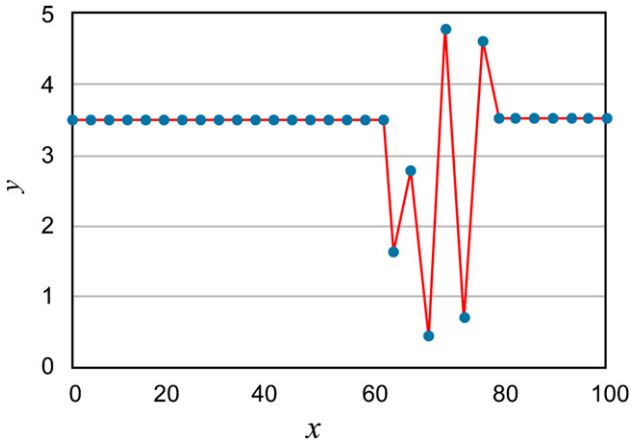

Fig. 11. Different velocities in different areas of the searching space.

Table 1
Relative improvements of FAD over DA for different dimensions and different iteration times.

D_s	*Itr*				
	50 (%)	100 (%)	200 (%)	350 (%)	500 (%)
5	1.69	3.62	4.56	4.61	4.67
10	2.32	3.57	4.74	4.85	4.92
15	2.41	3.64	4.81	4.90	5.03
20	2.74	3.91	5.17	5.18	5.21
25	3.29	4.39	6.13	6.54	6.58
30	3.63	5.13	6.75	6.99	7.08
40	3.88	5.55	6.82	7.16	7.24
50	3.95	6.17	7.12	7.15	7.49
60	4.24	6.69	7.85	8.29	8.33
70	4.77	6.22	8.06	9.16	9.69
80	5.68	7.81	9.64	10.17	10.90
90	5.69	8.59	10.72	11.09	12.18

Table 2
Relative improvement of FAD over DA for different number of samples.

n	*Itr*				
	50 (%)	100 (%)	200 (%)	350 (%)	500 (%)
10	2.30	3.25	4.16	4.65	4.86
30	3.46	4.01	4.49	5.04	5.25
50	3.95	4.97	6.72	7.15	7.49
75	4.11	5.02	6.88	7.52	7.77
90	4.29	6.08	7.01	7.57	7.89

In this study, we introduce a coefficient called the environmental perceptive factor to describe the particle's perception of its surrounding environment. To define it, we first introduce some concepts. In the ith step, the tth particle moves forward at velocity $V_t(i)$. The fitness of the ith step is denoted by $F_t(i)$, and the fitness of the $(i-1)$th step is denoted by $F_t(i-1)$. To express how fast the fitness changes with $V_t(i)$, a parameter, $\alpha_t(i+1)$, is defined as follows:

$$\alpha_t(i+1) = \frac{F_t(i) - F_t(i-1)}{\|V_t(i)\|} \tag{19}$$

A series of values of $\alpha_t(i+1)$ achieved from successive steps reflects how bumpy its nearby environment is. In order to measure the bumpiness, the standard deviation of the latest k

Table 3
Convergence speed of FWCPSO compared with PSO and CPSO.

Itr	PSO	FWCPSO	CPSO
(a) Rosenbrock			
1	1.02E+14	7.53E+13	7.53E+13
20	3.19E+12	4.76E+10	6.22E+10
40	2.19E+12	2.63E+10	3.82E+10
60	1.33E+12	1.14E+10	1.83E+10
80	6.99E+11	4.00E+09	8.00E+09
100	3.58E+11	1.15E+09	3.17E+09
120	1.87E+11	3.60E+08	1.44E+09
140	9.75E+10	9.36E+07	8.48E+08
160	5.21E+10	1.36E+07	6.66E+08
180	3.30E+10	1.44E+06	5.89E+08
200	2.20E+10	1.25E+05	5.57E+08
(b) Rastrigrin			
1	2.84E+03	1.49E+03	1.49E+03
20	2.84E+03	1.49E+03	1.49E+03
40	2.84E+03	1.49E+03	1.49E+03
60	2.82E+03	1.34E+03	1.36E+03
80	2.68E+03	1.11E+03	1.13E+03
100	2.47E+03	9.85E+02	9.93E+02
120	2.19E+03	9.11E+02	9.34E+02
140	1.94E+03	8.65E+02	8.93E+02
160	1.74E+03	8.43E+02	8.70E+02
180	1.57E+03	8.22E+02	8.51E+02
200	1.44E+03	7.98E+02	8.36E+02
(c) Griewank			
1	1.68E+03	4.79E+02	4.80E+02
20	1.67E+03	4.79E+02	4.80E+02
40	1.67E+03	4.79E+02	4.80E+02
60	1.67E+03	3.64E+02	3.67E+02
80	1.53E+03	1.55E+02	1.67E+02
100	1.28E+03	5.90E+01	7.04E+01
120	1.01E+03	2.53E+01	3.75E+01
140	7.65E+02	1.29E+01	2.53E+01
160	5.68E+02	5.48E+00	1.93E+01
180	4.39E+02	2.22E+00	1.69E+01
200	3.46E+02	1.31E+00	1.56E+01
(d) Ackley			
1	1.81E+01	1.36E+01	1.37E+01
20	1.81E+01	1.36E+01	1.37E+01
40	1.81E+01	1.36E+01	1.37E+01
60	1.81E+01	1.26E+01	1.27E+01
80	1.78E+01	9.75E+00	1.00E+01
100	1.72E+01	7.16E+00	7.68E+00
120	1.65E+01	5.52E+00	6.24E+00
140	1.55E+01	4.46E+00	5.47E+00
160	1.44E+01	3.54E+00	5.01E+00
180	1.35E+01	2.40E+00	4.69E+00
200	1.27E+01	1.20E+00	4.51E+00

Note: *Itr* is the iteration number; the other three columns are the minimum value of the corresponding methods.

values of $\alpha_t(i+1)$ is calculated at first as follows:

$$\sigma_t(i+1)=\sqrt{\frac{1}{k-1}\sum_{j=0}^{k-1}(\alpha_t(i+j)_i-\frac{1}{k}\sum_{j=0}^{k-1}\alpha_t(i-j))^2} \tag{20}$$

Then it is normalized further by the sigmoid function to produce the environmental perceptive factor as follows:

$$\beta_t(i+1)=\frac{2E}{1+\exp(\sigma_t(i+1))} \tag{21}$$

Here E is a given positive number to determine the maximum velocity. Factor $\beta_t(i+1)$ describes the circumstances of the environment and tells whether or not the particle will march faster. With this factor, Eq. (9) is updated by the following Eq. (22):

$$P_t(i+1)=P_t(i)+\beta_t(i+1)V_t(i+1) \tag{22}$$

7. Experiments

Our 3D retrieval system with RF is implemented in Microsoft Visual C# 2005 with SQL Server 2005 and the DirectX 2007 SDK. We use two libraries to test our proposed method. One is the PSB including animals, plants, vehicles, people and so on [10]. Another is the Purdue Engineering Shape Benchmark database (ESB) [23] including many kinds of CAD parts. To ensure the comparability of different RF methods, the same descriptor of 6-order moment extended based on Ref. [7] is used. Here we suppose the RF methods are robust to the descriptors though there is the probability that the comparison results will be different to some extent with different descriptors.

Specifically, we devise five groups of experiments to test our method. The first is to compare the FAD and DA. The second is to compare the convergence performance of FWCPSO with PSO and CPSO. The third is to show the different importance of the three types of parameters. The fourth is to exhibit the influence of dimensionality reduction on retrieval precision and computational efficiency. The last is to compare the performance of our parallel-optimization RF method extensively with other methods.

7.1. Comparison of FAD with DA

Here the value of FAD is calculated with our FWCPSO algorithm, and the value of DA is generated with Rui's method. Table 1 gives the improvement ratios of FAD over DA for different dimensions and different iteration times of the FWCPSO algorithm. One can see from Table 1 that the ratio increases along with the problem dimension. Also, we test the relationship

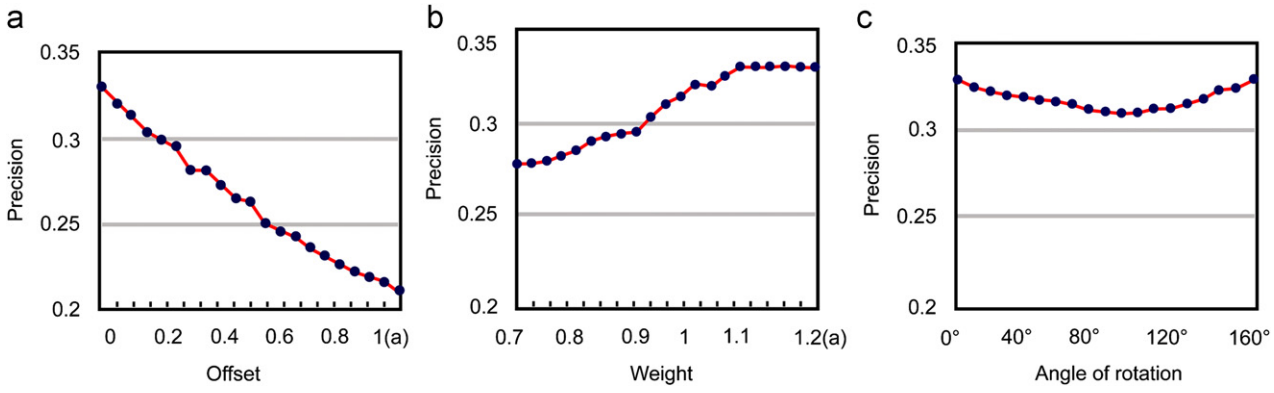

Fig. 12. Comparison of influence of translation, scaling and rotation: (a) influence of translation; (b) influence of scaling and (c) influence of rotation.

between the total number of samples (N) and the maximum number of iterations (Itr) in FWCPSO with a dimension of 50, as in Table 2, which lists the improvement ratios of FAD over DA.

7.2. Convergence comparison of FWCPSO with PSO and CPSO

Without the loss of generality, we use four typical functions (Rosenbrock, Rastrigrin, Greiwank and Ackley) to evaluate the convergence performance of FWCPSO while comparing to PSO [18] and CPSO [21]. In all the cases, the parameters are the same as in [22], that is, $c_1=c_2=2.05$, $\chi=0.729$, and $k=4$. Table 3(a)–(d) shows the convergence of WCPSO compared with PSO and CPSO. For all four functions, the dimension of the searching space (D_s) is 100, the swarm size (S) is 50, and the maximum iteration number is 200. It is obvious that for all typical functions, the proposed FWCPSO algorithm converges faster than PSO and CPSO.

Table 4
Results of the precision after the first round of RF for different proportion of parameters from T, S and R.

Proportion (T, S, R) (%)	Precision (%)					
	$D=20$	$D=30$	$D=40$	$D=50$	$D=60$	$D=80$
(80, 10, 10)	43.47	44.46	45.53	45.62	45.77	46.13
(70, 20, 10)	44.53	45.06	46.42	46.50	46.87	46.95
(60, 30, 10)	45.35	45.57	46.96	47.05	47.53	47.91
(50, 30, 20)	45.89	46.14	47.40	47.82	48.37	48.68
(50, 20, 30)	45.30	45.57	46.41	46.52	46.93	47.70
(40, 40, 20)	44.51	45.06	45.10	45.38	45.59	46.84
(40, 30, 30)	43.65	44.46	44.57	45.62	45.81	46.07

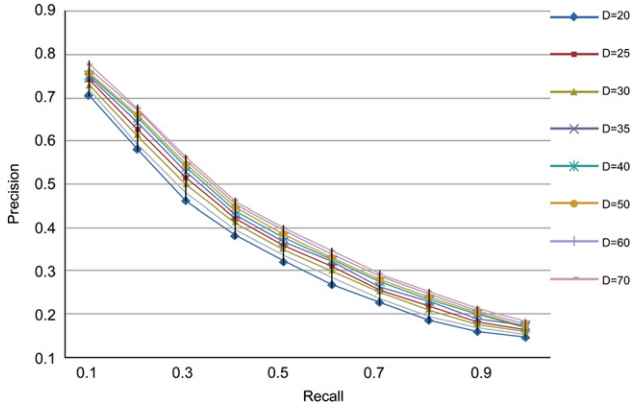

Fig. 13. The precision and recall curves for different dimension (D).

7.3. Influence of translation, rotation and scaling

In this section we explore the different influence of translation, rotation and scaling on RF with PSB first. And then we further explore the suitable number of parameters of translation, rotation and scaling for the given total number of parameters with the best RF result, which will be used to guide the dimension reduction. Obviously it's hard to show the effect of each parameter. We select two typical dimensions d_7 and d_8 to demonstrate how they affect the result. It can be seen from Fig. 12(a) that the retrieval precision decreases quickly while the similarity field is translated along the major axis. However, the retrieval precision increases from 28.2% to 33% while we scale the major axis from 70% to 120%. The reason may be that more positive examples are included in the similarity field when the major axis is lengthened. Moreover, it can be seen from Fig. 12(c) that the precision just varies slightly when we rotate the major axis. It is an important conclusion since the number of rotation parameters is much larger than the sum of other two kinds of parameters.

To further determine which part of dimensions (or corresponding parameters) should be reduced or preserved during different dimensionality-reduction strategies, we did a series of experiments as follows. For the given number of reduced dimensions, we asked five different designers to use different proportions of parameters from T, S and R in the final refined parameter vector to test the precision of RF. We recorded and analyzed the results. Table 4 gives some typical results after the first round of RF. We can see from Table 4 that most components of T and S are reserved during dimension reduction and the best proportions of parameters from T, S and R for the final refined vector are about 50%, 30% and 20%. This can be used to help set the target dimension of parameters for LLE procedure and rotation simplification procedure.

7.4. Effect of dimensionality reduction

This experiment demonstrates how dimensionality reduction affects retrieval precision and computational efficiency. The maximum number of iterations for FWCPSO is set to 1000. We

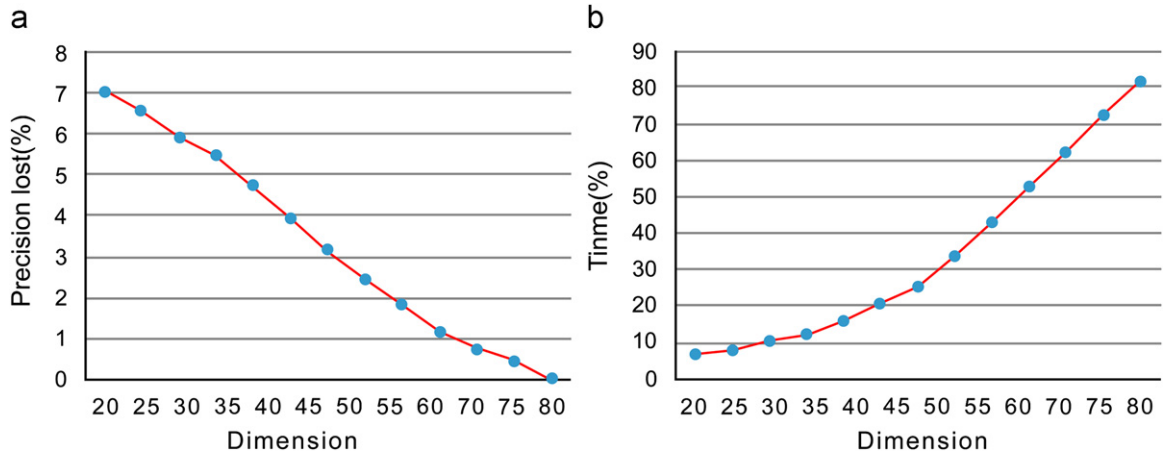

Fig. 14. The illustration of precision loss and the time cost varying with the different dimensions (D): (a) precision lost varying with dimension and (b) time cost varying with dimension.

can see from the P–R curve for different dimensions in Fig. 13 that the precision will increase as the number of dimension increases. In order to remove the less important component in the descriptor and keep the retrieval precision as high as possible, we must make some trade-offs.

Fig. 14 shows how the dimension of the descriptor (D) affects the precision and the speed of the relevance algorithm without rotation simplification procedure and dimensionality reduction procedure. From Fig. 14(a) we can see that if we want the loss of retrieval precision to be less than 2%, the dimension D shall be at least 60. While $D=80$ the total number of parameters is more than 3320, and the execution time of FWCPSO is 81.9 s, but when $D=60$, the dimension of the total number of parameters decreases to 1830 and the execution time decreases to 43.2 s. That is, the execution time has been reduced by 47.8%. Therefore it is worthwhile to lose limited precision to accelerate the RF in 3D model retrieval where real-time response is strongly required.

Furthermore, according to the proportion 50%, 30% and 20% for the number of translation, scaling and rotation parameters in the last section, and taking advantage of the dimension reduction strategies in Section 5, the number of scaling and rotation parameters are reduced to 36 and 24, respectively if the number of translation parameters is 60. Therefore the total dimension of the searching space should be reduced from 3320 to 120. The average time consumed by FWCPSO is reduced from 81.91 to 2.75 s. Obviously it is more effective to only deal with the reserved representative parameters than all the parameters produced by translation, scaling and rotation.

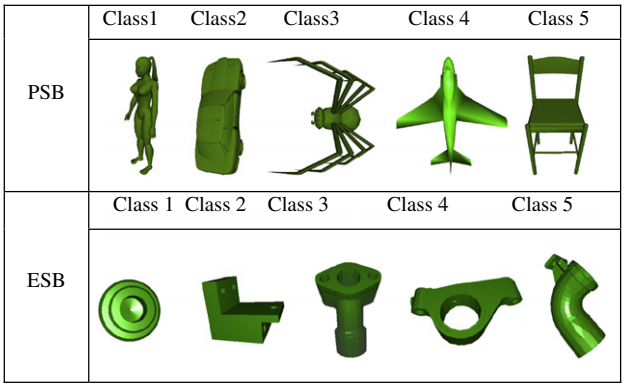

Fig. 15. Different types of models selected as query models.

7.5. Comparison of feedback results

We use this experiment to compare our parallel RF method with feature space wrapping approach [2], weights optimization approach [5], SVM approach [7] and BDA/LDA approach [8]. To ensure the comparability, the descriptor must be the same for all these methods. Therefore, the initial conditions for all RF methods are the same which are the retrieval results based on the same descriptor and the same retrieval method. For Elad's method, we take the libSVM [25] and make some minor modification to meet the requirement. For other methods, we implement them by ourselves based on the open source project Math.Net. Both of them can be obtain conveniently from the internet.

Five typical models shown in Fig. 15 are selected from the PSB library [10] and the ESB library [23] as query models, respectively. They represent five categories in the two libraries. To compare these methods, we consider the following measurements: precision–recall, discounted cumulative gain (DCG) and the second tier measurement [10]. During feedback, four PEs and four NEs are given. The max number iteration for FWCPSO is set to 500.

Fig. 16 illustrates the precision–recall curve of our feedback method in comparison with the methods in [2,5,7,8] after the first round of RF. Here, the precision is the average precision of the five query models for each library. It can be seen from Fig. 16 that all methods improve the initial retrieval result, which ranges from 6.1% to 14.7% on the PSB library and from 5.3% to 12.4% on the ESB library. Specifically, our method increases average precision by 14.7% on the PSB library and 12.4% on the ESB library.

Fig. 17 demonstrates the relation between the round number of the RF and DCG. The measurement DCG takes the sequence of the relevant models appearing in the result list into consideration together with the evaluation by the user. That is, the models in the front of the result will be assigned higher weights than those in the last. It is obvious from Fig. 17 that the first round is the most important and the measurement DCG increases with the round number of the interaction. Our method has the largest increase on DCG among all the five methods.

Fig. 18 gives the evaluation results with the second tier measurement, where the vertical axis stands for the percentage of similar models appearing in the first K models. $K=2*(|C|-1)$ and C is the number of models in the same class of the query model. Higher values of the vertical axis mean higher precision. The horizontal axes are the five query models, and the five clusters of bars stand for the five methods, including our method. One can see that as evaluated by the second tier measurement, our method performs better on both the PSB library and ESB library.

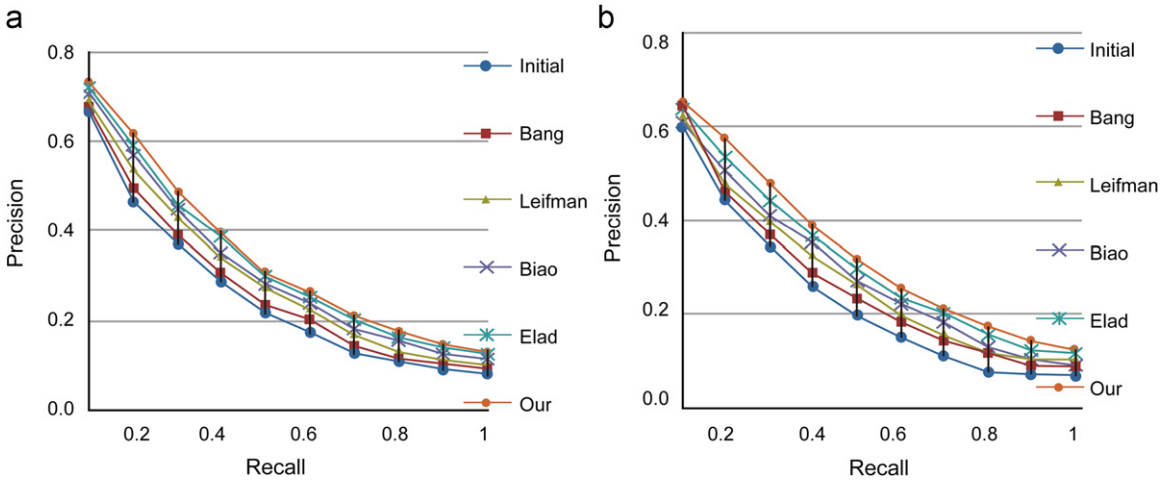

Fig. 16. The precision–recall curve of different methods after the first RF round: (a) P–R curve of PSB lib and (b) P–R curve of ESB lib.

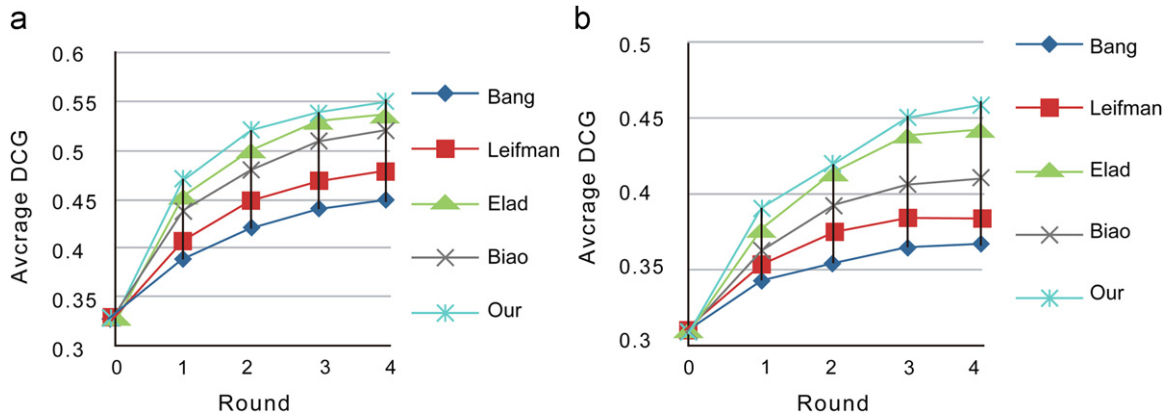

Fig. 17. Relationship between DCG and the RF round (round 0 is the initial retrieval result): (a) testing result of PSB lib and (b) testing result of ESB lib.

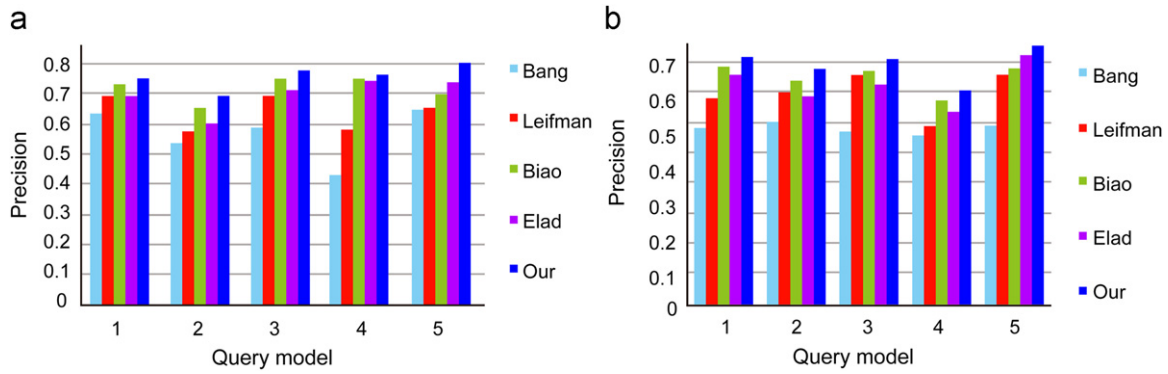

Fig. 18. Evaluation results with the second tier measurement: (a) testing result of PSB lib by second tier and (b) testing result of ESB lib by second tier.

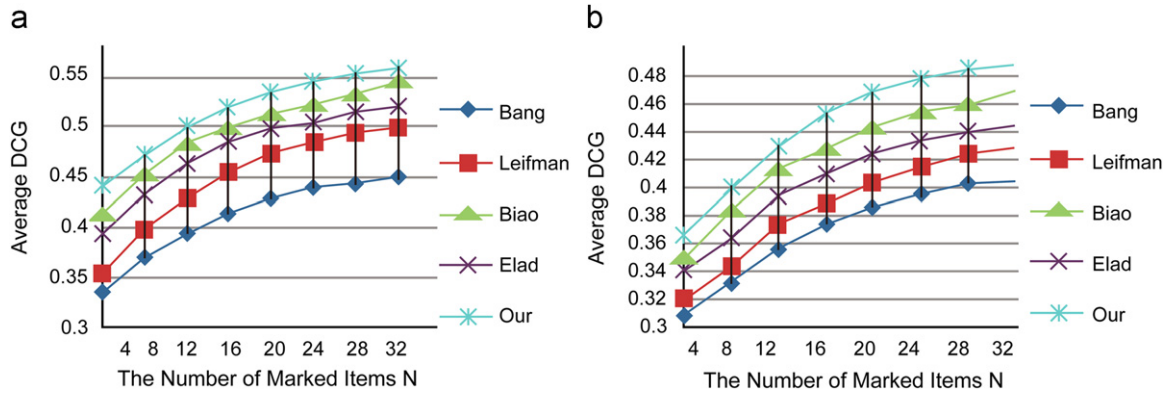

Fig. 19. DCG performance of the second RF round as a function of the number of marker items N in the first round: (a) testing result of PSB lib and (b) testing result of ESB lib.

To further measure how the users are involved in RF, the second-round protocol [24] is used. It can be seen from Fig. 19 that how the average DCG varies with the number of feedback examples N in which half of the RF examples are PEs and half are NEs. While the marked items are eight, the improvements are about 15% on PSB and 13% on ESB. However while 16 items are marked, the improvement is about 22% on PSB and 18% on ESB. It is can be seen that as the number of marked items N increases, the average DCG performance also increases but the rate of increment slows down.

In addition, we also calculate the computational performance statistically, where the dimension of the searching space for FWCPSO is 120 as in Section 7.3. The time cost of our method is largely dependent on dimension and the maximum number of iterations of the FWCPSO algorithm. We demonstrate the comparison in the following tables. In this study, the maximum iteration number is set to 500 for general purposes and may be higher for high precision requirements. The time cost of the proposed method is about 2.75 s after the parameter refinement strategies are adopted. The relative time-cost of other methods to ours are shown in Table 5. It can be seen that the proposed method is faster than Elad's and Leifman's but slower than Biao's and Bang's.

Table 5
Relative time-cost of different methods.

Methods	Bang	Leifman	Elad	Biao	Our method
Relative cost	0.68	75.81	5.66	0.86	1.00

8. Conclusions and future work

We proposed a novel parallel-optimization RF method based on FWCPSO in this study. The main characteristics of the work are:

(1) We established a unified mathematical model for RF that is formalized as a constraint optimization problem with multiple objectives based on SFM. This model regards the parameters of three operations modifying SFM as the optimization parameters. Here, to represent the user's retrieval intent more accurately and conveniently, we consider not only the relative distance between PEs and NEs but also the dispersion of PEs as the optimization objectives. We refine the mathematical model further.
(2) We proposed a PSO-based solution to handle SFM based RF in 3D model retrieval. All the parameters are treated equally and optimized simultaneously no matter which operation they belong to.
(3) We presented a novel FWCPSO algorithm to find a globally optimal solution to the RF the optimization problem efficiently. Here we use the *Fermat point* as the center point of the point set, which is more accurate than the arithmetic mean point. Furthermore, we introduce the environmental perceptive factor to adjust the marching velocity adaptively and thus accelerate the convergence.
(4) We devised a series of experiments to compare the performance of our FWCPSO algorithm with other related algorithms, which shows that FWCPSO performs better in most cases.

Three aspects of our work that require more research in the future are: refining the mathematical model so as to find better objectives to describe users' intentions in different ways; finding new functions that can express environment information better than the ones that we used in this paper; and devising a more appropriate method to determine the importance of different parameters during dimensionality reduction.

Acknowledgments

We thank all the reviewers for their comments and suggestions and appreciate previous researchers whose work helps us greatly. Also, the authors appreciate the support from 863 High-Technology Project of China (no. 2006AA01Z313, 2006AA01Z335) and Key Project of Science & Technology of Zhejiang Province (2008C01048) and Key Project of NSF of Zhejiang Project (Z107497).

References

[1] N. Iyer, S. Jayanti, K. Lou, et al., Three-dimensional shape searching: state-of-the-art review and future trends, Computer-Aided Design 5 (2005) 509–530.
[2] H.Y. Bang, T. Chen, Feature space warping: an approach to relevance feedback, in: IEEE International Conference on Image Processing, Rochester, New York, USA, 2002, pp. 22–25.
[3] I. Atmosukarto, W.K. Leow, Z. Huang, Feature combination and relevance feedback for 3D model retrieval, in: MMM, 2005, pp. 334–339.
[4] P. Papadakis, I. Pratikakis, T. Trafalis, et al., Relevance feedback in content-based 3D object retrieval: a comparative study, Computer-Aided Design and Applications Journal 5 (2008) 753–853.
[5] L. Biao, Q. Zheng, A powerful relevance feedback mechanism for content-based 3D model retrieval, Multimedia Tools and Applications Archive 1 (2008) 135–150.
[6] B. Hu, Y. Liu, S. Gao, Parallel global optimal approach of feedback for 3D CAD model retrieval, in: Proceedings of 2007 Tenth IEEE International Conference on CAD/Graphics, 2007, pp. 132–137.
[7] M. Elad, A. Tal, S. Ar, Content based retrieval of VRML objects—an iterative and interactive approach, EG Multimedia, 2001, pp. 97–108.
[8] G. Leifman, R. Meir, A. Tal, Semantic-oriented 3d shape retrieval using relevance feedback, The Visual Computer (Pacific Graphics), 2005, pp. 865–875.
[9] M. Novotni, G.J. Park, R. Wessel, R. Klein, Evaluation of Kernel based methods for relevance feedback in 3D shape retrieval, in: Proceedings of Fourth International Workshop on Content-Based Multimedia Indexing, Riga, Latvia, 2005.
[10] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The Princeton shape benchmark, in: Proceedings of Shape Modeling and Applications, IEEE Press, Italy, 2004, pp. 117–167.
[11] Y. Rui, T.S. Huang, Optimizing learning in image retrieval, in: Proceedings of IEEE Conference Computer Vision and Pattern Recognition, South Carolina, 2000, pp. 236–243.
[12] H.S.M. Coxeter, Introduction to Geometry, second ed, Wiley, New York, 1969.
[13] P.Y. Yin, B. Bhanu, Integrating relevance feedback techniques for image retrieval using reinforcement learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (2005) 1536–1551.
[14] C. Olivier, P. Haner, V.N. Vapnik, Support-vector machines for histogram-based image classification, IEEE Transactions on Neural Networks 5 (1999) 1055–1064.
[15] V.N. Vapnik, Principle of risk minimization for learning theory, in: Advances in Neural Information Processing Systems, vol. 4, 1992, pp. 831–838.
[18] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of Sixth Symposium on Micro Machine and Human Science, IEEE Service Center, Piscataway, NJ, 1995, pp. 39–43.
[19] X. Hu, R.C. Eberhart, Multiobjective optimization using dynamic neighborhood particle swarm optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002), Honolulu, Hawaii, USA, 2002, pp. 1677–1681.
[20] L.K. Saul, S.T. Roweis, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[21] L. Yu, Q. Zheng, S. Zhewen, L. Jiang, Center particle swarm optimization, Neurocomputing (2007) 672–679.
[22] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimentional complex space, IEEE Transactions on Evolutionary Computation 6 (2002) 58–73.
[23] S. Jayanti, Y. Kalyanaraman, N. Iyer, K. Ramani, Developing an engineering shape benchmark for CAD models, Computer-Aided Design 38 (9) (2006) 939–953.
[24] C.B. Akgul, B. Sankur, Y. Yeemez, et al., Similarity score fusion by ranking risk minimization for 3D object retrieval, in: Proceedings of Eurographics Workshop on 3D Object Retrieval, Crete, Greece, April 2008.
[25] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

About the Author—BAOKUN HU is currently a PhD candidate of State Key Laboratory of CAD&CG, Zhejiang University, PR China. His research interest is content-based 3D model retrieval and relevance feedback.

About the Author—LIU YUSHENG is currently a professor in State Key Laboratory of CAD&CG, Zhejiang University, PR China. He received his PhD degree in Mechanical Engineering from Zhejiang University, PR China in 2000. After that, he continued his postdoctoral research in State Key Laboratory of CAD&CG, Zhejiang University and City University of Hongkong in 2000–2003. His main research interests are computer aided tolerancing, content-based retrieval and model based systems engineering.

About the Author—SHUMING GAO is a professor of the State Key Laboratory of CAD&CG, Zhejiang University. He received his PhD degree from the Applied Mathematics Department of Zhejiang University in 1990, and was a visiting scholar and a visiting professor in the Design Automation Laboratory of Arizona State University respectively in 1996 and 2001. His research interests include product modeling, CAX integration, collaborative design, virtual reality in design and manufacturing, MEMS CAD, etc.

About the Author—RUI SUN is currently a PhD candidate of State Key Laboratory of CAD&CG, Zhejiang University, PR China. His research interest is 3D model simplification, content-based 3D model retrieval and relevance feedback.

About the Author—CHUHUA XIAN is currently a PhD candidate of State Key Laboratory of CAD&CG, Zhejiang University, PR China. His research interest is 3D model deforming, content-based 3D model retrieval and relevance feedback.