3D Object Retrieval Based on PSO-K-Modes Method

Xiangjun Zhao
School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, China
Email: xjzhao@jsnu.edu.cn

Mei Lu
School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, China
Email: lumei@jsnu.edu.cn

***Abstract*—By use of semantic attributes of 3D object, the user can search for targeted objects, which main advantage is that it does not require the user to sketch a 3D object as the query for 3D object retrieval, and the retrieval system can obtain a better retrieval performance. There are many categorical datum among these attributes, and how to use those and find the most similar objects is a vital problem to resolve. However, several elements with different types may have a shorter Euclidean distance. It is obvious the objects belonging to the same category are closer. Therefore, we present a 3D object retrieval method with clustering principle and RBF interpolator, which need a robust clustering method. The k-modes is a classic clustering algorithm for categorical data set. Its principle is simple, but it is easy to converge to a local optimum. PSO (Particle Swarm Optimization) algorithm is an effective tool for optimization, so we attempt to overcome the local optimum problem with PSO for categorical data set. PSO usually used to solve continuous optimization problems., but the categorical data are non-continuous. This paper presents an a novel k-p-modes algorithm to overcome these problems. Results show the method is effective.**

***Index Terms*—3D object retrieval, particle swarm optimization, K-Mode, clustering algorithm**

I. INTRODUCTION

Three-dimensional (3D) models have become widespread in not only research and industry, but also entertainment, such as computer vision, CAD, and computer game. Because of the demand to organize these 3D models for effective creation and distribution, 3D models retrieval has been drawn more and more attention. A shape-based 3D model retrieval system compares shape of 3D models for their retrieval[1]. However, 3D models that have a similar meaning may own different shape, or 3D models that have different meanings may own a similar shape. Literature [2] developed an alternative solution, and users search for targeted 3D objects in mind by specifying the attribute values. This is an intuitionistic way of delivering search intents to the system for common users, because humans tend to describe an object using high-level semantic attributes. Besides, these semantic descriptions of 3D objects are an important complement to low-level shape features, and thus can use to improve existing 3D object retrieval algorithms.

A common method to find the result objects is to calculate the Euclidean distance between an element and the virtual object with given attributes, and take the ones with smaller distance as research results. But it often is not a best result. Although some objects with same category maybe are further away from the virtual object than a few other elements with different category, since they have the same pattern, and have more similarity. Therefore, it is necessary to perform clustering analysis for finding a better retrieval result[3].

Cluster analysis is an important tool for information retrieval, which can divide the known data set into meaningful or useful clusters[4]. The K-means algorithm[5-7] is well-known for its efficiency in clustering large data sets. But, it may only stop at local optima of the optimization problem and working only on numerical data limits its use. Furthermore, how to use the clustering analysis to improve 3D retrieval is a subject to be noted and studied.

Large categorical data sets are often met in such areas as data mining. The categorical data are discrete, sometimes non-numeric. For example, there are a set of data describing the properties of 3D model: shape, uses, scale. Many clustering algorithms for categorical data have been developed. Huang proposed the algorithm for clustering large data sets with categorical values in 1998[5]. On this basis, Huang and others proposed a fuzzy k-modes algorithm in 1999[6]. These algorithms are similar to the k-means algorithm except for calculating the similarity between two elements. Like the k-means algorithm, the k-modes algorithm also stops at locally optimal solutions, that depends heavily on the initial modes and the distribution of the data set. Many literature have tried to find a globally optimal solution. For example, the fuzzy k-Modes algorithm, the tabu

Manuscript received January 1, 2012; revised June 1, 2012; accepted May 1, 2012.

Copyright credit, Associate professor, Ph. D. research interests include machine learning, and information retrieval etc.

doi:10.4304/jsw.8.4.963-970

searching method[7]; the genetic fuzzy k-modes algorithm[8].

In PSO, the system is initialized with a population of random solutions and searches for optima by updating generations. The PSO potential solutions, called particles, fly through the problem space by following the current optimum particles. Comparing with GA, PSO is easy to implement and has fewer parameters to adjust. PSO simulates the behaviors of bird flocking. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods. Another reason that PSO is attractive is that there are a few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. How to apply PSO to k-modes algorithm needs to be explored.

The rest of the paper is organized as follows: Section 2 introduces some related work. Our K-P-modes algorithm is introduced in Section 3. Section 4 describes 3D object clustering retrieval approach. A variety of experimental results are presented in Section 5. Finally, we provide some concluding remarks in Section 6.

II. Related Work

A. 3D Model Retrieval

Traditional 3D model retrieval puts the emphasis on shape matching. That is, determining the extent to which two shapes resemble each other[9]. Matching by feature correspondences and matching by global descriptors are two main approaches to this problem. The strategy in the former approach is to compute multiple local features for every object and then to compute a distance measure between pairs of objects[10]. On the other hand, the global descriptor-based approach reduces intrinsic shape characteristics to vectors or graph-like data structures, called shape descriptors, and then evaluates the distance between the descriptor pairs as a measure of similarity. Finding correspondences is a well-known difficult computational problem in computer vision and shape analysis[11]. Global descriptors method attempt to solve the correspondence problem by registering the shape information on a common grid. References [3] provides more comprehensive reviews.

Transform-based methods[12-20] implicitly register the surface points onto a 3D voxel grid by means of a scalar-valued function, which is then processed by transform tools such as 3DFourier[13], angular radial transform[14], 3D Radon[15], spherical trace transform[16], spherical harmonics[12, 17-19], or wavelets [20]. A significant advantage of adopting transform method is descriptor compaction achieved by keeping first few transform coefficients in the descriptor vector. Furthermore, pose invariance can be obtained by discarding the phase of the transform coefficients at the expense of some shape information[17].

Two-dimensional view-based methods[12, 21] consider the 3D shape as a collection of 2D projections taken from different viewpoints. Each projection is then described by 2D image descriptors like Fourier descriptors[12] or Zernike moments[21]. These methods work surprisingly well despite their logical disadvantage as they discard valuable 3D information. A possible explanation for their good performance is that, as the 3D models are completely given, projections can be produced in a controlled manner so that troublesome effects of occlusion and affine deformations are avoided. These methods can also be beneficial for 2D sketch-based queries.

In reference [12], a hybrid descriptor, which is a combination of two 2D view-based methods, DBI and SIL, and a transform based method REXT, is proposed. This descriptor, i.e. DSR, is proven to be very effective on PSB[22] and on the Konstanz database[23].

Graph-based descriptors[24-26] are fundamentally different from other vector-based descriptors. They are more elaborate, but they have the potential of encoding geometrical and topological shape properties in a more intuitive manner. However, It is not easy to generalize to all 3D shape representation formats and they require dedicated matching schemes. In fact, graph-based methods do not completely avoid the correspondence issue. They just lighten it by reducing the problem of matching two feature sets to that of matching graph nodes, which still remains a complicated task for general-purpose retrieval applications. It should be noted that some part of the information contained in a graph can be encoded in the form of vector-based numerical descriptions with tools from spectral graph theory.

One key issue for a user-friendly search engine is how to deliver user search intention to the system. Content-based retrieval causes great trouble for the user who does not have a 3D query that is similar to his targeted objects in the database. To avoid this problem, some systems allow the user to form a query by sketching the object's silhouette[1, 27]. The main drawback of these methods is that a 2D sketch is difficult to draw, which leads to less retrieval accuracy. Text-based query can only get a small part of 3D objects because currently most 3D objects are pure shapes without well classification and attribute information.

B. Clustering Analysis

Clustering is a fundamental exploratory analysis problem akin to discovering natural groupings in the observed data. It is known to be a hard optimization problem mainly in its unsupervised version. The lack of any knowledge, except the data themselves, poses important challenges at different stages in clustering analysis: choosing the right distance function and the relevant features, defining the clustering criteria, validating the solution.

Clustering has a huge search space and a vaguely defined optimum; these characteristics of the problem are addressed in the existing literature by means of various heuristics and experimental studies.

Different paradigms optimize different criteria and thus deliver different data partitions. Algorithms like K-Means or hierarchical forms account for the global data distribution. There are also algorithms which exploit only local properties and specify that neighboring data items should share the same cluster. This approach to clustering is called the connectivity principle and is implemented in the single-link hierarchical algorithm and in density-based method.

Each of the algorithms mentioned above has advantages and disadvantages with regard to the computational time and parameters tuning. More importantly, they deliver different solutions. Each algorithm is appropriate to a specific data distribution. The k-Means algorithm is very effective with regard to the computational time or parameter tuning but is applicable to Gaussian clusters of equal volumes. The connectivity principle yields clusters of various shapes but the methods implementing it may suffer from the chaining effect that causes undesirable long and narrow clusters, or are very sensitive to parameters.

Soft computing techniques were proposed to ease the drawbacks of the traditional clustering algorithms. Most of them minimize the within-cluster variance, being inspired by the k-Means algorithm.

There are a few strategies designed for clustering that optimize simultaneously several criteria. The use of multi-objective evolutionary algorithms is one of the most important contributions in this regard: the solution is evolved considering several criteria[28].

The current paper proposes a hybridization of k-Means with a swarm intelligence technique, aiming at enhancing the performance of the traditional clustering algorithm. Our method is consistently different from existing approaches for clustering based on swarm intelligence. Particle Swarm Optimization is used to introduce the connectivity principle into the K-Modes clustering algorithm; the new method thus takes into account both the local and global distribution of data.

C. Particle Swarm Optimization

Particle Swarm Optimization(PSO)[29] is a meta-heuristic mainly used for numeric optimization. Its use in combinatorial optimization necessitates rather complex adaptations, such as the redefinition of its operators. Initially, PSO was intended to simulate the social behavior of flocks but its authors observed the optimization capability of the agents involved in the simulation.

PSO maintains a population of particles, each one characterized by a position vector in the search space and a velocity vector which determines its motion. The velocity vector is computed following the rules:

a. Each particle tends to keep its current direction;

b. Each particle is attracted to the best position it has achieved so far;

c. Each particle is attracted to the best particle in population.

The velocity vector is computed as a weighted sum of three terms corresponding to the rules above. Two random multipliers are used to gain stochastic exploration capability while several coefficients are weights usually empirically determined.

Particle swarm optimization has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement. PSO simulates the behaviors of bird flocking. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. Each particle is updated according to the two "best" values. The first one is the position vector with its best fitness finish ever, and is denoted by p_{id}. The second one is the best fitness particle of this iteration, and is denoted by p_{gd}.

After finding the two best values, the particle updates its velocity and positions with the following two equations[29]:

$$v_{id}^{k+1} = w \times v_{id}^{k} + c_1 \times rand(.) \times (p_{id} - x_{id}^{k}), \\ +c_2 \times rand() \times (p_{gd} - x_{id}^{k}) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}, \quad (2)$$

where x_{id}^{k} is the value on the d th dimension of the i th particle. v_{id}^{k} is the velocity on the d th dimension of the i th particle. w is the inertia weight which is a parameter to control the impact of the previous velocities on the current velocity. c_1, c_2 are positive constants. rand(.) is a random number between 0 and 1.

Many literature attempted to combine particle swarm optimization with the clustering algorithm. For example, the fuzzy clustering algorithm based on PSO[30-32]. They all use PSO to solve the continuous problems. Yin BO, etc. propose fuzzy K-Prototypes clustering based on particle swarm optimization. Likely k-means, the method essentially uses a continuous data processing method[33].

Supposed that $\mathrm{D} = \{X_1, X_2, \ldots, X_n\}$ is a categorical data set with n objects, each of which is described by m categorical attributes $\{A_1, A_2, \ldots, A_m\}$. Attribute A_j (1<=j<=m) has n_j categories, i.e., $\mathrm{DOM}(A_j) = \{a_j^{(1)}, a_j^{(2)}, \ldots, a_j^{(nj)}\}$. n_j is the number of the value of attribute A_j. $X_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$ is the data object, where $x_{ij} \in \mathrm{DOM}(A_j)$, $1 \le i \le n$.

Def. 1 The dissimilarity measure between $X_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$ and $X_j = [x_{j1}, x_{j2}, \ldots, x_{jm}]$ can be defined by the total mismatches of the corresponding attribute categories of the two objects. The smaller the number of mismatches is, the more similar the two objects. Formally,

$$d(X_i, X_j) \equiv \sum_{l=1}^{m} \delta(x_{il}, x_{jl}), \ 1 \le i, j \le n, \quad (3)$$

where

$$\delta(x_j, y_j) = \begin{cases} 0, & x_j = y_j; \\ 1, & x_j \ne y_j. \end{cases} \quad (4)$$

Def. 2 Let the k cluster centers be represented by $Z=\{Z_1,Z_2,\ldots,Z_k\}$, and $Z_l=[z_{l,1},z_{l,2},\ldots,z_{l,m}]$, $(1\le l\le k)$. The objective of the k-Modes is to minimize

$$F(W,Z)=\sum_{l=1}^{k}\sum_{i=1}^{n}w_{li}d(Z_l,X_i)\quad 1\le l\le k,1\le i\le n, \quad (5)$$

subject to

$$w_{li}\in\{0,1\},1\le i\le n,1\le l\le k, \quad (6)$$

$$\sum_{i=1}^{k}w_{li}=1,1\le i\le n, \quad (7)$$

where $d(.,.)$ is defined in Eq. (5). W is the $k\times n$ partition matrix, and $w_{li}=1$ means the object X_i is a member of the cluster which center is Z_l. $F(W,Z)$ is also namely the objective function.

Theorem 1. The quantity $F(W,Z)$ defined in Eq. (5) is minimized if and only if

$$z_{l,j}=a_j^{(r)}\in DOM(A_j), \quad (8)$$

where

$$\left|\{w_{li}\mid x_{i,j}=a_j^{(r)},w_{li}=1\}\right|\ge\left|\{w_{li}\mid x_{i,j}=a_j^{(t)},w_{li}=1\}\right|,$$
$$1\le t\le n_j,\ 1\le j\le k. \quad (9)$$

Theorem 2. Let $Z=\{Z_1,Z_2,\ldots,Z_k\}$ be fixed, then the partition matrix W which minimizes the quantity $F(W,Z)$ defined in Eq. (5) subject to Eq. (8) is given by

$$w_{li}=\begin{cases}1, & if\ d(Z_l,X_i)\le d(Z_h,X_i),1\le l,h\le k;\\ 0, & \sum_{i=1}^{n}(X_i-\bar{X})^2,others.\end{cases} \quad (10)$$

Based on the two theorems described above, the k-Modes algorithm can be implemented recursively.

Algorithm 1

Step 1 Select k initial modes, one for each cluster.

Step 2 Allocate an object to the cluster whose mode is the nearest to it according to Eq. (3). Update the mode of the cluster after each allocation according to Theorem 1.

Step 3 After all objects have been allocated to clusters, retest the dissimilarity of objects against the current modes. If an object is found such that its nearest mode belongs to another cluster rather than its current one, reallocate the object to that cluster and update the modes of both clusters. (according to Theorem 2)

Step 4 Repeat step 3 until no object has changed clusters after a full cycle test of the whole data set.

III. THE K-P-MODES ALGORITHM

The complete K-P-modes algorithm includes digitizing the categorical data, discretizing the categorical data, selecting a dissimilarity measure, constructing the encoding and fitting function, and PSO optimization of K-Modes. The following paper will describe them step by step.

A. Digitizing the Categorical Data

All the categorical data are mapped to natural numbers. For example, there are some 3D Model data which has 3 attributes, functionalities, shape, natural property, shown in TABLE I. The datum are processed, and are shown in TABLE II .

B. Discretizing the Categorical Data

For each data object $X_i=[x_{i1},x_{i2},\ldots,x_{im}]$ in data set, its attribute value x_{ij} $(1\le i\le n)$ is now not only a single number, but an expanded set on the possible values, that is to say, for the *jth* value x_{ij} has a confidence coefficient on [0,1]. Besides, the default value in the matrix should be added by 0.

For example, the above data object X_3 =(2, 1, 1) would be now expanded as follows,

((0, 1, 2), (0, 1), (0, 1, 2))

$$X_3=\begin{bmatrix}0&0&1\\0&1&\\0&1&0\end{bmatrix},$$

After being added by 0,

$$X_3=\begin{bmatrix}0&0&1\\0&1&0\\0&1&0\end{bmatrix}.$$

Generally, let $n_{\max}=\max(n_j)(1\le j\le m)$, after processing, the data $X_i=[x_{i1},x_{i2},\ldots,x_{im}]$ is transformed into a $m\times n_{\max}$ matrix, which is as follows:

$$X_i=\begin{bmatrix}x_{i11}&x_{i12}&\cdots&x_{i1n_{\max}}\\x_{i21}&x_{i22}&\cdots&x_{i2n_{\max}}\\\vdots&&&\\x_{im1}&x_{im2}&\cdots&x_{imn_{\max}}\end{bmatrix}. \quad (11)$$

The excellence of is that we can now make the data more "continuously" for intermediate data object are defined, as soon as some coefficients are between 0 and 1. This idea is similar to expanding a natural number to an integer, or expanding an integer number to a real, etc..

C. Selecting Dissimilarity Measure Between Two Data Object

Since each data object change, the dissimilarity measure between X_i and Y_i should be changed accordingly.

$$X_i=\begin{bmatrix}x_{i11}&x_{i12}&\cdots&x_{i1n_{\max}}\\x_{i21}&x_{i22}&\cdots&x_{i2n_{\max}}\\\vdots&&&\\x_{im1}&x_{im2}&\cdots&x_{imn_{\max}}\end{bmatrix},$$

$$X_j=\begin{bmatrix}x_{j11}&x_{j12}&\cdots&x_{j1n_{\max}}\\x_{j21}&x_{j22}&\cdots&x_{j2n_{\max}}\\\vdots&&&\\x_{jm1}&x_{jm2}&\cdots&x_{jmn_{\max}}\end{bmatrix}.$$

The new form is as follows,

$$d\ (X_i,X_j)\equiv\sum_{l=1}^{m}\delta(x_{il},x_{jl}),\ 1\le i,j\le n, \quad (12)$$

where

$$\delta(x_{it}, x_{jt}) = \frac{1}{\sqrt{2}} \sqrt{\sum_{k=1}^{n_{\max}} (x_{it} - x_{jt})^2} \quad (13)$$

In Eq. (12), to divide $\sqrt{2}$ is to make our algorithm be compatible with the traditional k-modes algorithm. The objective function $F(W, Z)$ is the same as that of the traditional k-modes (as Eq. (5)) except for the dissimilarity measure, which is defined as Eq. (12).

TABLE I.
THE QUANTITATIVE CHARACTERISTICS OF 3D MODEL

Attr. / Object	Function	Shape	Natural property
X1	food	smooth	natural
X2	clothing	smooth	artificial
X3	housing	angular	natural
X4	transport	acuminate	natural

TABLE II.
THE INITIAL DATA OF 3D MODEL

Attr. / Object	Function	Shape	Natural property
X1	0	0	0
X2	1	0	1
X3	2	1	0
X4	3	2	0

D. The Encoding and Fitting Function

A single particle in the swarm represents one possible solution for clustering the document collection. Therefore, a swarm represents a number of candidates clustering solutions for the document collection. Each particle maintains a matrix $Z = (Z_1, Z_2, \ldots, Z_k,)$, where Z_i represents the Z_i cluster centroid vector and k is the cluster number. At each iteration, the particle adjusts the centroid vector' position in the vector space according to its own experience and those of its neighbors. We use $f(Z)$ as the fitness value to evaluate the solution represented by each particle. The fitness value is measured by the equation below:

$$f(Z) = \frac{1}{F(W, Z) + 1}, \quad (14)$$

where, $F(W, Z)$ is defined as Eq. (5). The smaller $F(W, Z)$ is, the higher the fitness value is, and the better the cluster is.

E. PSO Optimization of K-Modes

In PSO, each particle is made up of the k cluster centroids, whose form has changed as shown in (11), so the form of velocity should be changed accordingly. It has the same representation than a position: a list of sets. The only difference is that the coefficients can have any real value (not only on [0,1]).

We apply "standardized" to make sure that all coefficients in an expanded position fall in [0,1], and that the sum of all values in each line is 1. The example below shows how the operator works (remember that the coefficients are between 0 and 1).

$$\begin{bmatrix} 0 & 2 & 4 \\ 0.1 & 0.3 & 0.2 \\ 4 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0.33 & 0.67 \\ 0.17 & 0.5 & 0.33 \\ 0.8 & 0.2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The algorithm is as follows.

Step 1 Initialize

(1) Initialize the parameters. Give the cluster number k , the particle size s , study factors c_1 and c_2 , inertia weight w , particles' maximal velocity $V\max$ and the maximum iterative error.

(2) Initialize the k cluster centroids. Choose k data objects randomly as cluster centroids to generate one particle. Repeat it for s times to generate s particles. Then encode each particle and get its position matrix.

(3) Initialize pbest and gbest. Each particle's personal best position is initialized the same as its current position. Then calculate the global best position according to fitness, which is the highest fitness of all pbests.

Step 2 Generate partition matrix W .

Calculate the distance between every particle and each centroids of every particle according Eq. (12), then generate the partition matrix W according to Eq. (10), which allocates each data object to the nearest cluster. Repeat it for s times according to the centroids provided by s particles.

Step 3 Calculate each particle's fitness.

(1) Calculate the objective function $F(W, Z)$ according to the generated s partition matrix W and Eq. (5).

(2) Calculate the fitness according to Eq. (14).

Step 4 Update pbest and gbest.

(1) For each particle, set current value as the new pbest if its current fitness value is better than the best fitness value (pbest) in history.

(2) Choose the particle with the best fitness value of all the particles as the gbest.

Step 5 Update the particle, which means maybe generate new centroids. Update each particle's position and velocity according to Eq. (1), (2).

Notation:

(a) Particles' velocities on each dimension are clamped to a maximum velocity $V\max$. If the sum of accelerations would cause the velocity on that dimension to exceed $V\max$, which is a parameter specified by the user, then the velocity on that dimension is limited to $V\max$.

(b) The results do not need to be rounded.

Step 6 Termination. Judge the termination criterion, namely whether the iteration reaches the maximum iterations or the best fitness reaches the designated value. If the criterion is satisfied then stop the program, or else go to step 2.

Ⅳ. 3D OBJECT CLUSTERING RETRIEVAL METHOD

Clustering deals with finding a structure in a collection of unlabeled data. A cluster is therefore a collection of elements which are "similar" between them and are "dissimilar" to the elements belonging to other clusters. However, it is not to say that the distance between elements within a cluster must be smaller than the distance of elements between different cluster. That is to say

$$\{x \mid dist(x,y) < dist(x,z) \cap x, y \in C_1 \cap z \in C_2\} \neq \Phi.$$

One way of 3D model retrieval is to directly return the cluster hitted by retrieval condition, but which usually is a too large retrieval result. The other is to redesign a more reasonable method to solve the problem. We propose a new retrieval method based on RBF(Radial Basis Function)[34] and clustering analysis.

Supposed that X is a virtual element with the retrieval condition. If we want to get those elements closest to X, two factors should be considered, one is distance and the other is its cluster. We can get a revised distance between elements within a cluster according to Figure 1. The pentagram represents the virtual element with the retrieval condition. The outmost curve is the convex hull of the cluster hitted by retrieval condition. The inner curve is equidistance curves. How to revise the distance is the vital problem. The upper right window shows the target distance mapped from the equidistance curves and the pentagram is located in a central location. This is a typical interpolation problem.

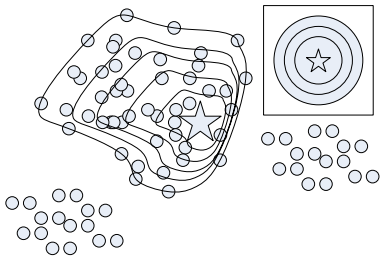

Figure 1. Revised distance within a cluster

Radial functions have proven to be an effective tool in multivariate interpolation problems of scattering data. Given the set of values $Y = (Y_1, Y_2, ..., Y_n)$ at the distinct points $X = (X_1, X_2, ..., X_n) \subset R^d$, we want to approximate the real valued function $f(X)$ by an interpolator $S(X)$; such that

$$S_i(X_i) = Y_i \quad i = 1, ..., N.$$

We choose $S(X)$ to be a RBF of the form

$$S(X) = p(X) + \sum_{i=1}^{N} \lambda_i \Phi(|X - X_i|), \quad X \in R^d, \quad (15)$$

where p is a polynomial of low degree and the basic function Φ is a real valued function on [0, ∞], usually unbounded and of non-compact support. λ_i is a real-valued weight, |.| denotes the Euclidean norm and $|X\text{-}Xi|$ is simply a distance which shows how far X is from X_i. Actually, a RBF is a weighted sum of translations of a radially symmetric basic function augmented by a polynomial term.

The mapping function S is actually an interpolator. The number of the constraints is N. Literature [34] showed that the smoothest interpolation function, which has the minimum semi-norm, has the simple form

$$S^*(X) = p(X) + \sum_{i=1}^{N} \lambda_i |X - X_i|, \quad (16)$$

where p is a linear polynomial, which has the form $p(X) = c_1 + c_2 x_1 + ... + c_{d+1} x_d$. The coefficients λ_i are real numbers and |.| is the Euclidean norm on R^d. Take into account the orthogonality condition

$$\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} \lambda_i x_{1i} = \sum_{i=1}^{n} \lambda_i x_{2i} = ... = \sum_{i=1}^{n} \lambda_i x_{di} = 0. \quad (17)$$

These side conditions along with the interpolation conditions of Eq. (16) lead to a linear system to solve the coefficients that specify the RBF. The linear system has the following form:

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} & 1 & x_{11} & \cdots & x_{d1} \\ \phi_{12} & \phi_{22} & \cdots & \phi_{2n} & 1 & x_{12} & \cdots & x_{d2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{1n} & \phi_{2n} & \cdots & \phi_{nn} & 1 & x_{1n} & \cdots & x_{dn} \\ x_{11} & x_{21} & \cdots & x_{n1} & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & 1 & 0 & \cdots & 0 \\ x_{1d} & x_{2d} & \cdots & x_{nd} & 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ c_1 \\ \vdots \\ c_{d+1} \end{bmatrix} = \begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \\ y_{n1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (18)$$

where $\phi_{ij} = |X_i - X_j|, i, j = 1, \ldots, n$.

Solving the linear system determines λ_i and c_i, and hence $S(X)$. Note the above linear system only resolves the mapping function y_{i1}, substituting y_{i1}(i=1, …, n) with y_{ij} (i=1, …, n; j=2, …, d) will give the linear system that determines S_j. In our method, target coordinate can be determined by the following method: the virtual element move to the centroid of the cluster, and elements locating at the convex hull are projected radially to the outer circle.

Based on the above interpolator, we can get a discernment formula:

$$Disc(X, X') = \begin{cases} dist(S(X), S(X')), & X, X' \in \text{cluster}_1; \\ dist(S(X), S(X'')) + dist(X', X'') & \\ \quad X, X' \notin \text{cluster}_1, \ X'' \text{ is the intersection} & \\ \quad \text{of envelope circle and the line } XS(X'). & \end{cases} \quad (19)$$

where *dist(.,.)* is the common distance function.

On the basis of cluster analysis, we can calculate the revised distance *Disc* between each element and X, and select m smallest *Disc* elements as the query results.

Ⅴ. 3D OBJECT SEMANTIC RETRIEVAL AND EXPERIMENT

We first verify the K-P-modes algorithm. We use two measures to evaluate the quality of our methods.

We use the corrected Rand index[35] to assess the recovery of the underlying cluster structure. Let $P = [C_1, C_2, \ldots, C_{k_1}]$ and $P' = [C_1', C_2', \ldots, C_{k_2}']$ be two clustering results of D. The number of points, denoted by

n_{ij}, simultaneously in C_i and C_j', i.e. $n_{ij} = |C_i \cap C_j'|$, then the corrected rand index is defined as

$$\gamma = \frac{\left(\binom{n}{2}\sum_{i=1}^{k_1}\sum_{j=1}^{k_2}\binom{n_{ij}}{2} - \sum_{i=1}^{k_1}\binom{|C_i|}{2}\sum_{i=1}^{k_2}\binom{|C_j'|}{2}\right)}{\left(\frac{1}{2}\binom{n}{2}\left(\sum_{i=1}^{k_1}\binom{|C_i|}{2} + \sum_{i=1}^{k_2}\binom{|C_j'|}{2}\right) - \sum_{i=1}^{k_1}\binom{|C_i|}{2}\sum_{i=1}^{k_2}\binom{|C_j'|}{2}\right)}. \quad (20)$$

TABLE III.
COMPARISON BETWEEN THE RESULTS OF K-P-MODES AND GF K-MODES

	Max F	Ave. F	Best γ	Ave. γ	the number of times γ=1
GF k-modes	193.833	209.083	1.000	0.771	24
K-P-modes	120.31	126.5	1.000	0.982	82

The corrected rand index γ ranges from 0 when the two clustering have no similarities, to 1 when the two clustering are identical. Because we know the true clustering of the data set, the true clustering and the resulting clustering are used to calculate γ.

Our data set is the well-known soybean data set[36]. The soybean data set has 47 records, each of which is described by 35 attributes. Each record is labeled as one of the four diseases: diaporthe stem rot, charcoal rot, rhizoctonia root rot and phytophthora rot. Except for the phytophthora rot which has 17 instances, all other diseases have 10 instances each. Since there are 14 attributes that have only one category, we only selected other 21 attributes for the purpose of clustering.

Results are shown in TABLE Ⅲ. We specify k =4, s =20, $c_1 = c_2$ =2, w =0.5, and max = 200. After running a hundred times, the misclassification rate is 1.91 using our algorithm, and 3.76 using genetic fuzzy k-Modes algorithm for clustering categorical data (GF k-modes).

Compared the value of $F(W, Z)$, Ave. γ, The number of times γ =1, we know, the one by our algorithm is better than the one by GF-k-modes. And The misclassification rate by ours is much lower.

We provide eleven attributes for the user to describe 3D objects, including functionalities(food, clothing, housing, transport, other), shape(smooth, angular, acuminate), natural property (natural and artificial). At the clustering stage, we perform clustering analysis of the 3d model. At retrieval stage, given every attribute value, we can decide the category of the object with the attribute value, and can get retrieval result. Certainly, if the result set is too large, we can use multi-tier clustering, and get smaller category and acute result. Our 3D retrieval data set is the all 1814 models of the Princeton Shape Benchmark [22]. 3D model retrieval results are shown in figure 2-4. When we specify the attribute vector (function, shape, natural property) is equal to (transport, angular, artificial), and a retrieval result can be got in figure 2. If let (function, shape, natural property) is equal to (other, angular, artificial), and a retrieval result can be got in figure 3. If (function, shape, natural property) is equal to (transport, smooth, artificial), we can get a retrieval result shown in figure 4.

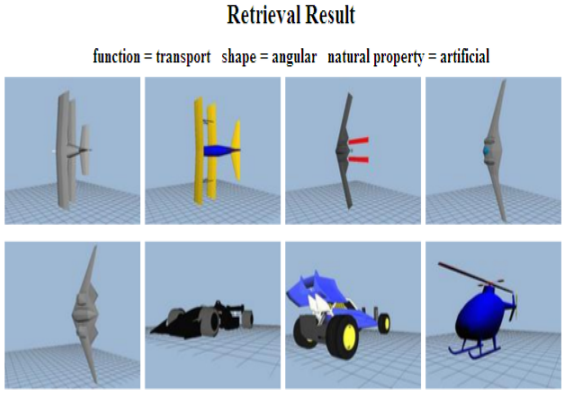

Figure 2. Retrieval result specifying angular artificial transport model

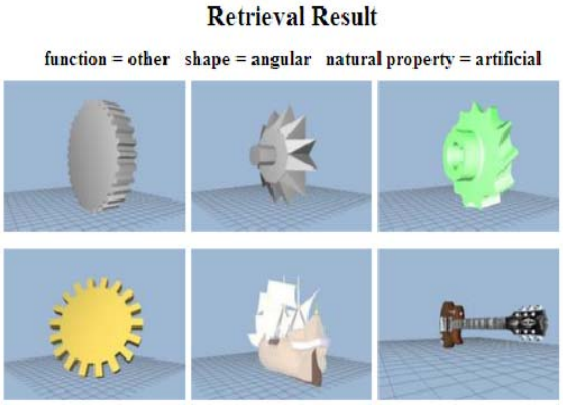

Figure 3. Retrieval result specifying acuminate artificial model

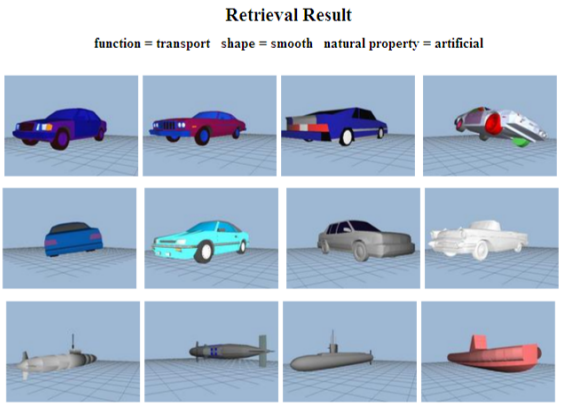

Figure 4. Retrieval result specifying smooth artificial transport model

VI. CONCLUSION

In this paper, we have present a user-friendly 3D object retrieval system. With this system, the user is able to search the database using semantic attributes to describe targeted objects, instead of finding or sketching a 3D object as the query. So far there are eleven attributes in our system. This set can be enriched in future work. To obtain a better retrieval result, and we integrate the PSO algorithm and the k-Modes algorithm to find the global optimization method. Thinking of that the PSO is often used to solve the continuous problem, while the

categorical data are discrete, we integrate the two by expanding the categorical data into numerical and continuous data. Two experiments show that the k-p-modes is promising and effective. It also points out a novel method to resolve the problem like this.

ACKNOWLEDGEMENTS

Research is partially supported by the National Natural Science Foundation of China under Grants No. 61272297.

REFERENCES

[1] Yoon S., Scherer M., Schreck T., et al. Sketch-based 3d model retrieval using diffusion tensor fields of suggestive contours[C].In: Proc. of ACM MM. 2010.
[2] Gong Boqing, Liu Jianzhuang, Wang Xiaogang, et al. 3d object retrieval with semantic attributes[C].In: ACM Multimedia. 2011. 757-758.
[3] Liu Yu, Ling Xiaoxi, Shi Zhewen, et al. A survey on particle swarm optimization algorithms for multimodal function optimization[J]. Journal of Software, 2011, 6(12):2449-2455.
[4] Tan Pang-Ning, Steinbach Michael, Kumar Vipin. Introduction to data mining[M]: The People's Post and Telecommunication Press. 2006.
[5] Huang Z. Extensions to the k-means algorithm for clustering large data sets with categorical values[J]. Data Mining and Knowledge Discovery, 1998, 2(3):283-304.
[6] Huang Z, Ng M.A Fuzzy k-modes algorithm for clustering categorical data[J]. IEEE Trans.on Fuzzy Systems, 1999, 7(4):446-452.
[7] Ng M., Wong J. . Clustering categorical data sets using tabu search techniques[J]. Pattern Recognition Letters, 2002, 35(12):2783-2790.
[8] Gan G., Wu J., Yang Z. A genetic fuzzy k-modes algorithm for clustering categorical data[J]. Expert Systems with Applications, 2009, 36:1615-1620.
[9] Tangelder J.W.H., Veltkamp R.C. A survey of content based 3d shape retrieval methods, in *Multimedia Tools and Applications*. 2008.
[10] Funkhouser T., Shilane P. Partial matching of 3d shapes with priority-driven search, in *Proc. Symp. Geometry Processing*. 2006.
[11] Belongie S., Malik J., Puzicha J. Shape matching and object recognition using shape contexts[J]. IEEE Trans. Pattern Analysis and Machine Intelligence, 2002, 24(4): 509-522.
[12] Vranic D.V. 3d model retrieval[D]. Univ. of Leipzig, 2004
[13] Dutagaci H., B.Sankur, Y.Yemez. Transform-based methods for indexing and retrieval of 3d objects, in *Proc. Fifth Int'l Conf. 3D Digital Imaging and Modeling*. 2005.
[14] Ricard J., Coeurjolly D., Baskurt A. Generalizations of angular radial transform for 2d and 3d shape retrieval[J]. Pattern Recognition Letters, 2005, 26(14):2174-2186.
[15] Daras P., Zarpalas D., Tzovaras D., et al. Shape matching using the 3d radon transform[C].In: Proc. Second Int'l Symp. 3D Data Processing, Visualization, and Transmission. 2004. 953-960.
[16] Zarpalas D., Daras P., Axenopoulos A., et al. 3d model search and retrieval using the spherical trace transform[J]. EURASIP J. Advances in Signal Processing, 2007.
[17] Kazhdan M., Funkhouser T., Rusinkiewicz S. Rotation invariant spherical harmonic representation of 3d shape descriptors[C].In: Proc. 2003 Eurographics/ACM SIGGRAPH Symp. Geometry Processing. 2003. 156-164.
[18] Vranic D.V. An improvement of rotation invariant 3d shape descriptor based on functions on concentric spheres[C].In: Proc. IEEE Int'l Conf. Image Processing. 2003. 757-760.
[19] Papadakis S.P.P., Pratikakis I., Theoharis T. Efficient 3d shape matching and retrieval using a concrete radialized spherical projection representation[J]. Pattern Recognition, 2007, 40(9):2437-2452.
[20] Laga H., Takahashi H., Nakajima M. Spherical wavelet descriptors for content-based 3d model retrieval[C].In: Proc. IEEE Int'l Conf. Shape Modeling and Applications. 2006. 15-25.
[21] Chen D.-Y., Tian X.-P., Shen Y.-T., et al. On visual similarity based 3d model retrieval[J]. Computer Graphics Forum, 2003, 22:223-232.
[22] Shilane P., Min P., Kazhdan M., et al. The princeton shape benchmark[C].In: Proc. of Shape Modeling International. 2004.
[23] Bustos B., Keim D.A., Saupe D., et al. Feature-based similarity search in 3d object databases[J]. ACM Computing Surveys, 2005, 37(4):345-387.
[24] Hilaga M., Shinagawa Y., Kohmura T., et al. Topology matching for fully automatic similarity estimation of 3d shapes[C].In: Proc. ACM SIGGRAPH. 2001. 203-212.
[25] Tung T., Schmitt F. The augmented multiresolution reeb graph approach for content-based retrieval of 3d shapes[J]. In'l J. Shape Modeling, 2005, 11.
[26] Sundar H., Silver D., Gagvani N., et al. Skeleton based shape matching and retrieval[C].In: Proc. Shape Modeling Int'l. 2003.
[27] Princeton 3d model search engine, http://shape.cs.princeton.edu/search.html.
[28] Handl J., Knowles J. Improving the scalability of multiobjective clustering[C].In: Proceedings of the Congress on Evolutionary Computation. 2005.
[29] Zhong Shaobo, He Zhongshi. Application of particle swarm optimization algorithm based on classification strategies to grid task scheduling[J]. Journal of Software, 2012, 7(1):118-124.
[30] Xu Lei, Zhang Fengming. Fuzzy clustering algorithm based on pso[J]. Computer engineering and design, 2006, 27(21):4128-4129.
[31] Liu Jingming, Han Lichuan, Hou Liwen. Cluster analysis based on particle swarm optimization algorithm[J]. Systems engineering-theory & practice, 2005, 6:54-58.
[32] Feng Zheng, Yan Min, Zhang Zhifeng A pso-based fuzzy clustering algorithm[J]. Computer engineering and applications, 2006, 27:150-151, 165.
[33] Yin Bo, He Songhua. Fuzzy k-prototypes clustering based on particle swarm optimization[J]. Computer engineering and design, 2008, 29(11):2883-2885.
[34] Carr J, Beatson R, J J Cherri, et al. Reconstruction and representation of 3d objects with radial basis functions[C].In: Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH Los Angeles, California, USA, 2001. 67-76.
[35] Hubert L., Arabie P. Comparing partitions[J]. Journal of Classification, 1985, 2:193-218.
[36] Blake C., Merz C. . Uci repository of machine learning databases, http://www.ics.uci.edu/mlearn/MLRepository.

Xiangjun Zhao, Associate professor, Ph. D. research interests include computer graphics, machine learning, and information retrieval etc.

Mei Lu, Ph. D. student, research interests include machine learning, information retrieval etc.