Geodesic Distance Based SOM for Image Clustering

Chunqi Shi1,2, Sulan Zhang1,2, Zheng Zheng1,2 and Zhongzhi Shi1

Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences
Graduate School of the Chinese Academy of Sciences
100080 Beijing, China
shicq,zhangsl,zhengz,shizz@ics.ict.ac.cn

Abstract - We propose an unsupervised approach, Geodesic Distance Based SOM (GDBSOM), for clustering object images acquired under varying viewpoints, into disjoint subsets corresponding to individual objects. Given a high dimensional image space in which each image is a point represented by a vector, the high dimensional image space was projected into cluster, each of which contains image vectors from a single low-dimensional manifold (object) embedded in the high dimensional image space. The proposed GDBSOM algorithm uses the geodesic distance instead of Euclidean distance, which can reflects the intrinsic geometric structure of manifold embedded in image space more efficiently. Based on the Self-Organizing Maps (SOM) architecture, the GDBSOM algorithm well performs the unsupervised clustering and reduces the high and nonlinear dimension of image space into cluster maps of object images. The experimental results show that the GDBSOM algorithm performs better than traditional image clustering algorithms for object images under different viewing viewpoints.

I. INTRODUCTION

Content-Based Image Retrieval (CBIR) [1] is one of the long-standing research problems in computer vision and information retrieval. Particularly there is a growing interest in CBIR of large image databases in recent years. Image clustering is very important for efficient search and retrieval in large image databases [2]. In order to perform image clustering, we have to choose a representation space and then use an appropriate distance measure (or similarity measure), to match images and cluster centers in the selected representation space. The image clustering is then performed in a supervised process by using human intervention [3] or in an unsupervised process by relying on the similarity between the images and the various cluster centers [4].

In this paper the problem of clustering images of objects from different viewpoints was discussed. Given an unlabelled set of images including kinds of objects, an unsupervised method is proposed to divide the images into several disjoint subsets, such that each of the subsets only contains images of a single object. Namely the unsupervised method automatically learns the cognitive concepts of objects from large proper image database. Images represented as complex vectors form a high dimensional image space, and then the high dimensional image space is learned and reduced in the proposed clustering process which generates an organized structure in the low dimension space. The cluster result is just the low dimensional concepts space according to different objects.

With regard to such problem of clustering object images from different viewpoints, there are different approaches have been developed. The approach based on features that are insensitive to changes in viewpoints [5]. However, features such as edges cannot always be acquired stably. Moreover, information essential for clustering may be ignored by focusing on parts of images. The generative methods approach [6] has been proposed in which images of an object under varying viewpoints is generated from a small number of training images of objects, assuming the Lambertian model. The limitation is that it cannot generate specular reflection components. The approach of pattern recognition uses all pixel values as inputs. For instance, eigenfaces proposed by Turk and Pentland [17] compresses images by applying Principal Component Analysis (PCA).

The clustering framework presented in this paper is based on Self-Organizing Maps (SOM) [7]. SOM is an unsupervised algorithm which learns and visualizes characteristic features or other abstractions from the input data. It produces nonlinear dimensionality reduction by fitting a low dimensional topologically ordered grid of nodes in the high dimensional observation coordinates. This paper presents an image clustering algorithm based on the architecture of the SOM algorithm, named GDBSOM.

In the image clustering process, the SOM algorithm could be used to project the image space which is high dimensional onto a hyper plane, typically two-dimensional map of category prototype vectors represented by neurons. The projection provides an ordered map of the high dimensional space with similar image vectors clustered together into neighborhoods. Most of traditional image cluster techniques build on the assumption that the image space is Euclidean. However, in many cases, the image space might be a set of non-linear sub-manifolds which are embedded in the image space and the SOM algorithm fails to model a visually obvious structure in the data, as addressed by Tenenbaum [8]. The iterative optimization may get stuck at locally optimal solutions, when nonlinear structure in the data cannot simply be regarded as a perturbation from a linear approximation, while the proposed the GDBSOM algorithm could get an extremely well performance. Here, GDB is short for Geodesic Distance Based, which means we propose utilizing the geodesic distance instead of Euclidian distance when coming to proper neighbor of the winning neuron. The geodesic distance could better reflect the intrinsic geometric structure of manifold embedded in image space. As accepted, a good choice of a specific distance measure can bring better clustering results. In that reason, our approach has improved greatly the accuracy of clustering.

For high dimensional image space, some researches on clustering have ever considered geometric structure of image samples. There are image clustering algorithms that explicitly use the concept of the manifold [9, 10]. Generally speaking, the former approaches only consider the local structure of manifold. In [9], the concept of local linearity is embodied in
the idea of tangent space of the appearance manifold; PCA is used to estimate local linear subspaces. In [10], $L'$ distance metric of the image space is used to determine a neighborhood structure in the image space for each input image, and its concept of local linearity is focused on how best the neighbors can linearly approximate a given sample point. Different from them, the GDBSOM algorithm considers the global structure of intrinsic manifold and tends to give a more faithful reflection of the object images global structure, and its metric-preserving properties are better understood theoretically.

Our experimental results indicated that using geodesic distance measure can greatly improve the clustering result. Images of the 3D objects in the COIL database and human faces from the UMIST Face Database are used in our experiment. The result shows that our algorithm achieves lower error rates in the object image clustering which indicates that clustering using geodesic distance measure can greatly improve the clustering result.

The rest of this paper is organized as follows. The GDBSOM algorithm is proposed in Section II. The experimental results are shown in Section III. Finally, the conclusions and several issues for future work are given in Section IV.

II. THE GDBSOM APPROACH FOR IMAGE CLUSTERING

Unsupervised learning algorithm SOM is a biologically inspired method to generate useful representation of data object. We use it to cluster the similar image from the same object together and generate an organized structure in low dimension space. We proposed Geodesic Distance Based SOM (GDBSOM) algorithm, because the geometric distance of the manifold embedded in image space enables the SOM to follow the topology of the underlying data set better.

The input of the problem is the image database $S$ consisted of unlabelled images $\{I_1, \ldots, I_n\}$. We assume that all images have the same number of pixels $L$, and then we view each image $I_i$ as a sample point $x_i = (x_{i1}, \ldots, x_{iD})$ in $\mathbb{R}^D$. Our algorithm outputs a cluster assignment for these images $\Pi : \{I_1, \ldots, I_n\} \rightarrow \{1, \ldots, C\}$. Two images $I_i$ and $I_j$ belong to the same cluster if and only if $\Pi(I_i) = \Pi(I_j)$. In our definition, a cluster only consists of images of one object with different view points.

A. The Geodesic Distance

Biologically the retinal image is a collection of signals from photoreceptor cells. If these numbers are taken to be coordinates in an abstract image space, then an image is represented by a point. Only three dimensions of the image space are depicted, but actually the dimensionality is equal to the number of photoreceptor cells. Due to the unwanted variations resulted from changes in lighting, facial expression, pose and etc, the image space might not be an optimal space for visual representation. To recognize them, the brain must equate all images from the same manifold, but distinguish between images from different manifolds [11]. To the best our knowledge, our approach using geodesic distance is approximately the process of equating all images from the same manifold.

In conventional image clustering approach, there is a defect using the Euclidean distance measure between pairs of images. Its drawback for using the Euclidean distance is illustrated in Fig. 1. Three images $P_1, P_2, Q_1$ (three points in high dimensional image space) on two different manifold. $P_1$ are in the same manifold with $P_2$, but in different manifold with $Q_1$, implying that $P_1$ and $P_2$ belong to the same cluster (object), while $P_1$ and $Q_1$ are not. The point $P_2$ appears deceptively closer to the point $Q_1$ than to the point $P_1$, as measured by their straight-line Euclidean distance, but by using their geodesic distance $P_2$ appears closer to $P_1$ than $Q_1$, for $P_1$ and $Q_1$ are in different manifold and their distance is definitely larger. It can be found that in the high-dimensional input image space, the Euclidean distance may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold embedded in image space. As the geodesic distance can reflect the intrinsic distance among images more efficiently, the geodesic distance based clustering approach should be intuitionence more accurate.

By constructing a sparse graph in which each node is connected only to its closest neighbors, the geodesic distance between each pair of nodes is taken to be the length of the shortest path in the graph that connects them. We use the approximated geodesic distances according to Floyd’s algorithm [12]. Floyd’s algorithm is very intuitive, well understood and produces reasonable mapping results. Also, the algorithm has been proved that the short paths between pairs of points provide increasingly better approximations to the intrinsic geodesic distances as the number of data points increase, and become arbitrarily closely as the density of data points tends to infinity [13]. Namely, the computed shortest path can be a substitute for geodesic distance and it also reflects the intrinsic structure of submanifold embedded in high dimensional image space. Its detailed steps are presented following:
1) Construct Neighborhood Graph

Construct a weighted graph $G(S)$ by connecting each point to all points. The edge weight between any two image points $X_i, X_j \in S$ is determined as follows:

$$d_{\text{Edge}}(X_i, X_j) = \begin{cases} \|X_i - X_j\| & \text{if } \|X_i - X_j\| < \varepsilon \\ \infty & \text{else} \end{cases}$$ (1)

Here, the radius $\varepsilon$ is a suitable constant. So the points which are neighbors on the manifold $M$ are determined and the constructed graph models the local geometrical structure of the image manifold.

2) Compute Shortest Paths

Estimate the geodesic distances $d_M(X_i, X_j)$ between all pair of points on the manifold $M$ by computing their shortest path distance $d_{\text{Edge}}(X_i, X_j)$ in the graph $G(S)$. And the shortest path distance:

$$d_M(X_i, X_j) = \min \{ d_{\text{Edge}}(X_i, X_j), d_{\text{Edge}}(X_i, X_k) + d_{\text{Edge}}(X_k, X_j) \}$$ (2)

Then the geodesic distances between all pairs of points in high dimensional image space is estimated.

B. The GDBSOM Algorithm

In the large image database, considering that the number of objects could be very large, the GDBSOM algorithm is based on the origin and simple SOM algorithm architecture which runs more swiftly. In order to accelerate GDBSOM’s computation, first the batch formulation is used, which has the advantages of faster convergence and less sensitive to the order of presentation of the sample data. Besides, an appropriate initialization is also very important because of its affection on GDBSOM’s convergence. In order to initialize the weights of the GDBSOM’s neurons for a smooth and swift convergence, the instruction from Mu-Chun Su’s efficient initialization scheme [14] was followed, but with a few differences: when initialization of neurons we used the intersample geometric distance within the intrinsic manifold of input images instead of the intersample Euclidean distance; and we did not follow the second step of initialization. The detail of our efficient initialization scheme is described as follows:

1) Initialize the Neurons on the Four Corners of GDBSOM’s Map Based on Largest Intersample Geometric Distance.

First select a pair of input image vectors with largest intersample geometric distance from the input images. Initialize the weights of the neurons on the lower left corner and the upper right corner (i.e., $w_{41}$ and $w_{4M}$ in Fig. 2) with the pair of vector. Then initialize the neuron on the upper left corner (i.e., $w_{41}$ in Fig. 2) with the image vector selected from the remaining input image vectors, which is farthest to the two previously selected image vectors according to the geometric distance. In the end the neuron on the lower right corner (i.e., $w_{4M}$ in Fig. 2) is set to be the image vector farthest to the previously three selected samples similarly.

2) Initialize the Neurons on the Four Edges.

Always select four image vectors from the remaining sample using the methods in step 1 to initialize the neuron four edges (i.e., $w_{1k}$, $w_{M(M+1-k)}$, $w_{(M+1-k)}$, $w_{kM}$ $k = 2, \cdots, M-1$ in Fig. 2).

3) Initialize the Remaining Neurons Using the Corner and Edge Neurons’ Weight.

We initialize the remaining neuron from left to right and from top to bottom using linear interpolation and the formula is represented below:

$$w_i = \left( w_{M} - w_{i1} \right) \frac{(j-1)}{(M-1)} + w_{i1}$$

$$i = 2, \cdots, M-1; j = 2, \cdots, M-1$$ (3)

![Fig. 2. The arrangement of the GDBSOM’s map.](image)

Respect to the proposed initialization, the GDBSOM’s map effectively samples the images space according to the low dimensional manifold and well reflects the structure of the manifold embedded in the image space. As a result it allows the GDBSOM algorithm to converge faster to the solution.

Then, the training of the GDBSOM’s map follows. Two important steps, winner search and weight updating, are involved in the training process. In the winner search step, the similarities of the input image vector with each neuron vector in the GDBSOM’s map are measured by the geodesic distance instead of the Euclidean distance. We defined the Approximate Geodesic Distance for calculating the similarity. As in the batch process, each neuron corresponds with a cluster of trained image vectors, and the Approximate Geodesic Distance is the minimum of all geodesic distances between the input image vector and the trained image vectors. In the weight updating step the batch updating is adopted for more swiftly training. There are two important updating coefficients $\alpha(t)$.
(the learning rate) and \(\beta(t)\) (the neighborhood function) used to adjust the training convergence. The detail of the GDBSOM algorithm is presented as follows:

1) Initialization
Use the approach depicted in section 4 to calculate the geodesic distances between all pairs of image vectors \(\{X_1, \ldots, X_n\}\). Then initialize the weights of neurons using efficient initialization methods mentioned above.

2) Winner Search
For each input image vectors \(X_i\), find the winning neuron \(i^* j^*\), using the minimum geodesic manifold distances instead of using the Euclidean distance:

\[
i^* j^* = \arg\min_s \text{ApproximateGeodesicDistance}(X_i, w_j)
\]

where the size of the GDBSOM’s map is \(M \times M\). And then the \(X_i\) is classified into the weight vector \(w_{i^* j^*}\).

The definition of the Approximate Geodesic Distance (AGD) between \(X_i\) and \(w_j\) is:

\[
\text{AGD}(X_i, w_j) := \min_{t} \text{GeodesicDistance}(X_i - X_t)
\]

where \(X_t\) is the image vector classified into the weight vector \(w_j\).

3) Weights Updating
Adjust the weights of neurons in batch model, using the following rule:

\[
w_{ij}^{\text{new}} = w_{ij}^{\text{old}} + \alpha(t)(\sum_{i \in S_y} X_i - w_j^{\text{old}})
\]

where the components of set \(S_y\) are input vectors classified into \(w_j\) satisfying:

\[
i - \beta(t) \leq i' \leq i + \beta(t)
\]

\[
j - \beta(t) \leq j' \leq j + \beta(t)
\]

and \(N_y\) is the number of components of \(S_y\).

The two parameters: \(\alpha(t)(0 < \alpha(t) < 1)\), \(\beta(t)(0 \leq \beta(t))\) are learning coefficients for the \(t\)th cycle defined below:

\[
\alpha(t) = \max\{0.01, \alpha_{\text{init}}(1 - t / \tau_{\alpha})\}
\]

\[
\beta(t) = \max\{0, \beta_{\text{init}}(1 - t / \tau_{\beta})\}
\]

where \(\alpha_{\text{init}}\) and \(\beta_{\text{init}}\) are the initial values, \(\tau_{\alpha}\) and \(\tau_{\beta}\) are the time constants.

4) Updating Topological Neighbors on the GDBSOM Map and Repeat.

III. EXPERIMENT

In this section, the experience results are given. We implemented the GDBSOM algorithm in MATLAB, and two different types of object image databases were used to test the algorithm: images of various 3D objects and images of human faces. There are some differences between them: the surface texture and profile of 3D objects varies greatly while human faces have much limited variation in surface texture. All the object images used in our experience were taken under varying viewing pose, so images from one object could be different a lot in conventional similarity, but we need to cluster them together as one object. The result of the experiment shows that the GDBSOM algorithm did extremely well in this condition.

The images of various 3D objects are from the Columbia Object Image Library (COIL-100) [15], which contains pictures of 100 everyday objects (see Fig. 3). This database consists of 7200 colour images: 72 different views of every object. The images used in the experiment were obtained by placing the objects on a turntable and taking a view every 5 degrees. Experiments are performed on the shape-images of the object. Here “shape-images” means the converted 32 x 32 grey-level images from the original images, and the conversion formation used is : grey-value = \(0.31R+0.59G+0.10B\). Then the \(m = 1024\) dimensional image vector we got were used in our experiment. Several sample shape-images converted are shown in Fig. 4.

And the images of human faces comes from the UMIST Face Database [16] which consists of 575 images of 20 people, each covering a range of poses from profile to frontal views, and the subjects cover a range of race, sex and appearance. And the number of face images from each person varies, for example the lowest is 24 while the largest is 84. The images are all in PGM format, approximately 220 x 220 pixels in 256 shades of grey. All the images are resized to 64 x 64 pixels. Then we got \(m = 4096\) dimensional image vectors.

Fig. 3. Several original COIL images.
The results of the GDBSOM algorithm are presented in TABLE I. Error rates are calculated as the ratio of the number of misclustered images over the number of total images. As is clear in TABLE I, our algorithm produces very good clustering results. Most clusters resulted from the GDBSOM algorithm could perfectly represent objects based on the image database. Fig. 5 and Fig. 6 show the resulting clusters generated by the GDBSOM algorithm for the two datasets respectively.

<table>
<thead>
<tr>
<th>Image Databases</th>
<th>GDBSOM Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Images</td>
</tr>
<tr>
<td>COIL100</td>
<td>7200</td>
</tr>
<tr>
<td>UMIST</td>
<td>564</td>
</tr>
</tbody>
</table>

We also compare GDBSOM with three other approaches. The first approach is SOM without using geometric distance, which performs extremely poorer than GDBSOM. Then we also use K-means methods based on the geodesic distance and the Euclidean distance for comparison, the results obtained are detailed in TABLE II. The results show that the proposed GDBSOM algorithm could perfectly clustering images of objects acquired under varying viewpoints.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Image Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>COIL100</td>
</tr>
<tr>
<td>GDBSOM</td>
<td>7.3%</td>
</tr>
<tr>
<td>SOM</td>
<td>37.9%</td>
</tr>
<tr>
<td>Geodesic distance + K-means</td>
<td>12.7%</td>
</tr>
<tr>
<td>Euclidean distance + K-means</td>
<td>43.8%</td>
</tr>
</tbody>
</table>

Moreover, we did another experiment on various dense samples for testing the performance of the GDBSOM algorithm and other methods: five databases formed by selecting images from the converted COIL100 images database, simply they are acquired by taking 10, 15, 20 and 30 degrees apart as the object is rotated on a turntable. Then with the converted images which taking 5 degrees apart, there are six image databases for testing. The result is showed in Fig. 7. From the results, it can be found that, when images are more densely sampled, the GDBSOM algorithm could greatly reduce its error rate and get better results than other methods. Namely when images from objects become denser, the manifold embedded in the image space could be learned more exactly from the image space and get better result.
IV. CONCLUSION

We proposed the GDBSOM approach for clustering object images database. We generally view the SOM as the brain of the GDBSOM algorithm, and apply the geometric distance measurement to observe the image space. Experiments demonstrated that the GDBSOM algorithm is indeed effective for clustering images of 3D objects undergoing large pose variation or face images covering a wide range of poses. We proposed several adjustments for more smoothly and swiftly running of the GDBSOM algorithm. Results from the GDBSOM algorithm show that we achieve effective recognition of the different cognitive concepts (objects or human faces) from large proper image database. For broader and more practical applications, the combination with other image features such as colour histogram representation and coordinates for image positions will be studied in future work.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation of China (No. 60435010), the National Science Foundation of China (No. 90604017), the National Basic Research Priorities Programme (No. 2003CB317004), and the Nature Science Foundation of Beijing (No. 4052025).

REFERENCES