OPTIMIZING SIMILARITY BASED VISUALIZATION IN CONTENT BASED IMAGE RETRIEVAL

Giang P. Nguyen and Marcel Worring

Intelligent Sensory Information Systems, University of Amsterdam,
Kruislaan 403, 1098SJ Amsterdam, The Netherlands
{giangnp,worring}@science.uva.nl

ABSTRACT

In any CBIR system, visualization is important, either to show the final result to the user or to form the basis for interaction. Advanced systems use 2-dimensional similarity based visualization which show not only the information of one image itself but also the relations between images. A problem in interactive 2D visualization is the overlap between the images displayed. This obviously reduces the search capability. Simply spreading the images on the screen space will not preserve the relations between them. In this paper, we propose a visualization scheme which reduces the overlap as well as preserves the general distribution of the images displayed. Results show that an effective balance between display of structures and limited overlap can be achieved.

1. INTRODUCTION AND RELATED WORK

Every image retrieval system has a visualization step either to display results or to allow users to interact with the images. Visualization is essential to help the user to gain better understanding of the data in order to give meaningful feedback. Similarity based visualization of an image database does not simply display images on the screen at random positions, but aims at showing the relations between the images in the database.

In literature, the work on similarity based visualization in CBIR can be classified into two main categories. The first one is referred to as 1D visualization, the most common way used in CBIR systems. In such methods, the images are listed to the user based on some pre-defined criteria such as degree of similarity to a given example. Examples of this category are described in [6]. However this way of visualization can only be used if one has an example to begin with. Furthermore, it does not support the user in understanding the distributions of the database as the relations between the images are ignored.

The second class are the recently developed systems with 2D visualization [2], [4], [5]. In [5], it is argued that visualizing images in 2D is essential to bridge the "semantic gap" [6] as the meaning of an image does not depend only on the image itself but also on the context of other images. In this category, the images are displayed in a screen to show the relations between them. Since the 2D visualization shows the relations between images in clusters, this helps the user in concentrating on the right groups of images. Visualization in higher dimensions is also considered [3]. In this paper, we focus on the 2D visualization.

To show images on the screen, we have to find a suitable method for mapping data from the high dimensional feature space to the 2D screen. Most of the methods for dimension reduction in literature are meant for point sets. When such a method is applied to a set of images, the images displayed will partially or fully cover one another. This is emphasized due to the fact that images tend to form clusters in feature space. The overlap reduces the search capability, and introduces the possibility of missing images when they are overlapped by another image. For example, when there are outliers in the systems, images far from all other images, the remainder will build up a crowded cluster which is difficult for the user to interact with. To increase the view, the overlap between images should be considered. It should be noted here that this is also very relevant for display on PDA's as screen space there is very limited.

Randomly spreading images over the visualization space will break the general relations between them. In this paper, we propose a new method satisfying the following criteria. First of all, the set of displayed images should show a general overview of the whole dataset. Secondly, when mapping data from high dimensions to 2D, the original structure of the data in high dimensional space should be preserved. Finally, the selected method should reduce as much as possible the overlap thus making optimal use of the screen space. The paper is organized as follows. In section 2, we give a detailed description of our approach. Section 3 presents our experimental results to demonstrate the effectiveness of the method.

2. OUR APPROACH

2.1. Dissimilarity matrix: a distances preservation method

The main purpose of visualizing a set of images in a 2D space is to preserve as much as possible the intrinsic relations between images, as well as to give the user the best visualization to interact with. Now given a set of N images, assume each image I has a set of features $F = \{f_1, f_2, \cdots, f_{\mathcal{F}}\}$. The result of image comparisons is a matrix $\mathcal{D} = (d_{ij})_{N\times N}$, where d_{ij} is the dissimilarity between image i and image j.

In high dimensional feature space, the data generally has a nonlinear structure. However, in literature, most methods use a dissimilarity matrix based on Euclidean distance or histogram intersection [6]. Hence, they ignore the intrinsic structure of the data given in the high dimensional space. In order to keep the data structure, we employ the IsoMap algorithm [7]. In this method, the inner relations between all objects in the data are taken into account. With given $\mathcal{D}$, a neighborhood graph $\mathcal{G}$ is first built using k nearest neighbors. This graph determines for each object i its neighbors and defines:

$$d^*_{ij}(\mathcal{G}) = \begin{cases} d_{ij} & \text{if } j \text{ is neighbor of } i, \\ \infty & \text{otherwise.} \end{cases}$$

The next step redefines the similarity values. Between object i and j, a shortest path in $\mathcal{G}$ is calculated using Dijkstra's algorithm which defines a new dissimilarity value:

$$d^*_{ij}(\mathcal{G}) \overset{Dijkstra}{\longrightarrow} d'_{ij}(\mathcal{G}).$$

Figure 1 shows an example of the advantage of the IsoMap method. The left image shows an example of data in high dimensional space. For a normal distance computation, such as Euclidean distance, the nonlinear manifold of the dataset as shown in the right image is ignored. This means points far from each other along the manifold can yield short distances. As we want to preserve the real distances between data points in the high dimensional space, the IsoMap algorithm in the right image will obtain distances following the structure of the manifold.

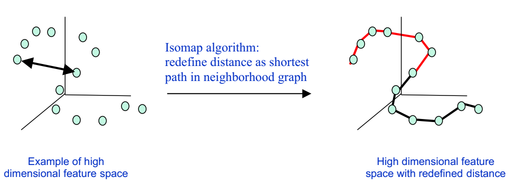

Figure 1: Example of redefining dissimilarity between objects using IsoMap algorithm

2.2. Finding 2D visualized positions

When mapping from a high dimensional data space to a lower dimensional representation space, an exact match between the data driven distances and the screen distances in general is impossible. Methods choose different data characteristics to preserve. Examples are using distance preservation [4], variance preservation [2], and neighbor preservation [1].

As stated above, the overlap between images in visualization space should be reduced to increase the search capability. It is not just that simple to spread images over the representation space as the relations between them in high dimensional space should be taken into account. This will not preserve distance but for visualization structure we believe neighbor preservation is more important. For example, if the $d_{ij} = 2 \times d_{ik}$ in the high dimensional space, the distance between i and j does not have to be twice as large as long as it is larger than the distance between i and k in visualization space. This leads us to the choice of using SNE.

2.2.1. SNE: a neighbor identities preservation method

Given a dissimilarity matrix $\mathcal{D} = \{d_{ij}\}$. For each object, the SNE method [1] computes the probabilities that it would take others to be its neighbors $\mathcal{P} = \{p_{ij}\}$:

$$p_{ij} = \frac{\exp\left(\frac{-d^2_{ij}}{2\sigma^2_i}\right)}{\sum_{k\neq i} \exp\left(\frac{-d^2_{ik}}{2\sigma^2_i}\right)}. \tag{1}$$

with $p_{ii} = 0$ and Gaussian neighborhood σ_i is set by hand. In the screen space, each object is represented as a point. First, the SNE initializes all points in random positions on a 2D screen $\{y_i\}_{i=1}^N$. The *induced probability* $\mathcal{Q} = \{q_{ij}\}$ is then calculated for each pair of points:

$$q_{ij} = \frac{\exp\left(-||y_i - y_j||^2\right)}{\sum_{k\neq i} \exp(-||y_i - y_k||^2}, \quad q_{ii} = 0.$$

Finally, using the Kullback-Leibler distance between two probability distributions $\mathcal{P}$ and $\mathcal{Q}$, the SNE aims at minimizing the following function:

$$C = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

2.2.2. Optimizing the visualization results

Because the original SNE concentrates on local relations between points, points with very small probabilities will be kept far away from the rest. As the size of visualization space is fixed, the rest will build up crowded clusters, this would not be a good choice for the user to interact with. We identify the steps which can be changed in order to improve the overall results. More precisely, this improvement aims at spending as much as possible of visualization space to represent data, as well as reducing the overlap between points.

The first modification is in the initialization step. As described above, this method depends on the initialized points. In [1], points are initialized randomly around the origin.

However, this way of visualization does not use the visualization space optimally. Furthermore, this makes the result of 2D visualization dependent on the initialization. Therefore, we propose a way of setting up the initial positions for all points.

From equation (1), we have a matrix of probabilities $\mathcal{P} = \{p_{ij}\}$, where $\forall i,\ \sum_j p_{ij} = 1.0$. We find a pair of points k and l with minimum values of probability p_{kl}. This means point k^{th} and l^{th} are furthest from each other. Those two points are placed in screen space along a mean line where the distance between them is taken from $\mathcal{D}$. We have:

$$\begin{cases} y_i = (0, \frac{d_{kl}}{2}) \\ y_j = \left(d_{kl}, \frac{d_{kl}}{2}\right). \end{cases}$$

The remainder are initialized along the line based on the sorted distances in the graph $\mathcal{G}$ to k and l. Since the SNE tries to match the two distributions $\mathcal{P}$ and $\mathcal{Q}$, during the process of adjusting positions, those two points always remain with the longest distance. As we initialize them in two furthest positions in visualization space, we guarantee that the data will be able to cover $\mathcal{V}$ as much as possible.

The other modification to SNE aims at reducing the overlap. In the original SNE, if there exists a point k such that the summation of all probabilities that other points would take k as their neighbor $\sum_j p_{jk}$ is very small, the method will keep point k far from the rest. The remainder will build up crowded clusters.

Assume point k is far from the rest. To reduce the distances between k and $i \neq k$, the probabilities value p_{ki} should be increased. As the $\sum_j p_{jk} = 1 = const$, to increase value p_{kt} with ϵ, where ϵ is a small value added to increase the probability. This will reduce p_{kl} for $l \neq t$. On the other hand, in order to keep the neighbor identities, the general relations between points should be preserved. This means if $p_{kt} < p_{kl}$ then $p_{kt} + \epsilon < p_{kl}$. This way of changing is very complicated. We decide to change the values of p_{ii} for all $i = 1, \cdots, N$. By changing the value p_{ii} we can adjust point i around its current position.

Now, assume $p_{ii} = \lambda$, since the total of probabilities is 1.0, increasing the value of λ will reduce the remaining p_{ij} with scale $(1 - \lambda)$ where $\lambda \in [0, 1]$, $j \neq i$. This will push other points around i further from it. This means that all the points are more separate from each other. Clearly as keeping neighbor relations and reducing overlap are conflicting criteria, λ defines a parameter for balancing the two.

2.3. Giving an overview of the database

In an interactive image search process, to let the user define search queries, the system first shows some images to the user. The user then selects the ones that are close to what he/she is looking for. This step can be done randomly or based on some predefined criteria to choose a set of images to be displayed. Randomly selecting images from a large image set is not a proper way as the displayed set of images will not be able to cover all the possibilities of the dataset. Therefore, we use the k-means cluster algorithm in order to classify the dataset into m clusters. This process takes the features of all images. Each image will be represented as a vector of feature values. The k-means algorithm is then applied using the dissimilarity matrix. After this, the current dataset is divided into m groups of images. In the visualization process, a set of m images is displayed to the user each time. This assures that the user can have some ideas of how the dataset is distributed.

Our complete method is described as shown in figure 2. In the next section, we show some experimental results of our approach.

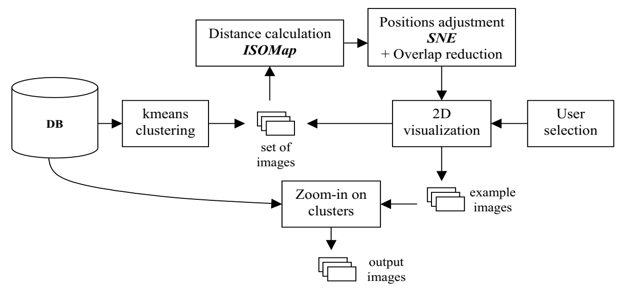

Figure 2: Combination scheme of the proposed 2D visualization with user interaction.

3. EXPERIMENTAL RESULTS

In our experiment, we use a dataset of 5000 keyframes which is extracted randomly from a very large video database. The Lab color histogram is used as feature (32 bins for each color channel) for each keyframe. Similarity values are based on histogram intersection. In the similarity calculation step, the IsoMap is used to redefine the dissimilarity. The matrix $\mathcal{D}$ is used as input for the SNE algorithm. The initialization of SNE is as mentioned above.

In the experiment, we test with λ in $[0, 1]$. If $\lambda = 0$ the method is the original SNE, which means the neighbor preservation is maximum. If $\lambda = 1$, all $p_{ij} = 0, j \neq i$, hence there will be no neighbor preservation, all points are pushed far away from others. This means the viewing area of data in screen space reaches its maximum. In figure 3, we show the relation between viewing area and neighbor preservation for varying values of λ. The neighbor preservation cost is taken from [1]. The overlap is the sum of all the area one image is covered by the others.

Figure 4 shows some results, which compare the result using the proposed method with $\lambda = 0$ (left images), $\lambda = 0.1$ (middle images), and $\lambda = 1.0$ (right images) in case of 50 and 500 images. From the results, it is observed that with changing λ the overlap can be reduced. However, when increasing λ the structure of the data is lost. With $\lambda = 0.1$

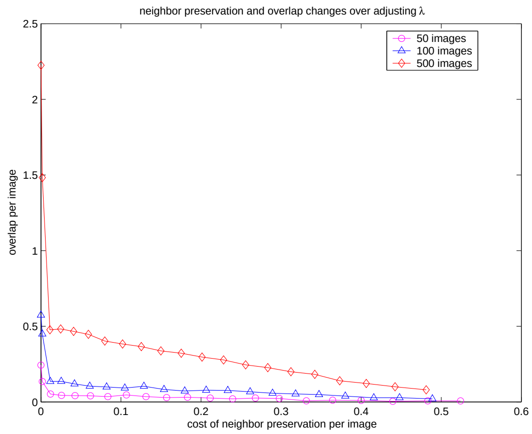

Figure 3: The changing of overlap and neighbor preservation cost value by adjusting value of λ in case of showing 50, 100, 500 images.

we observe that the general relations between images are preserved, while the overlap between them is significantly reduced.

Since the visualization space is fixed, increasing the number of displayed images will reach to the limitation of screen space. Let us assume that an image is called viewable if its visible area occupies $k\%$ of the image. Every displayed images has the same size of w and h, and the visualization space has size $\mathcal{W}$ and $\mathcal{H}$. The maximum number of images that the proposed method can afford to reduce the overlap will be $\frac{\mathcal{H}\times\mathcal{W}}{h\times w}k$. In fact, if there are too many images displayed it will be confusing for the user. Hence the number of displayed images should be chosen such that it is sufficient to represent the dataset as well as suit for interaction.

4. CONCLUSION

In this paper, we have discussed an improvement in visualizing data in CBIR systems. 2-dimensional visualization is good in showing relations between images. However, the problem of overlap limits the capability of searching. The original SNE method preserves the neighbor identities which can show the data in 2D without losing their neighborhood. By adjusting some factors of the methods, we can represent data in lower dimensions which still has the advantage of SNE and solves the problem of overlap. IsoMap is used as a tool to redefine the dissimilarity matrix, since it takes into account the intrinsic structure of data in high dimensional space. The combination of those two gives better visualization result in the sense that it is able to keep general relations between images, while the overlap between them is reduced remarkably.

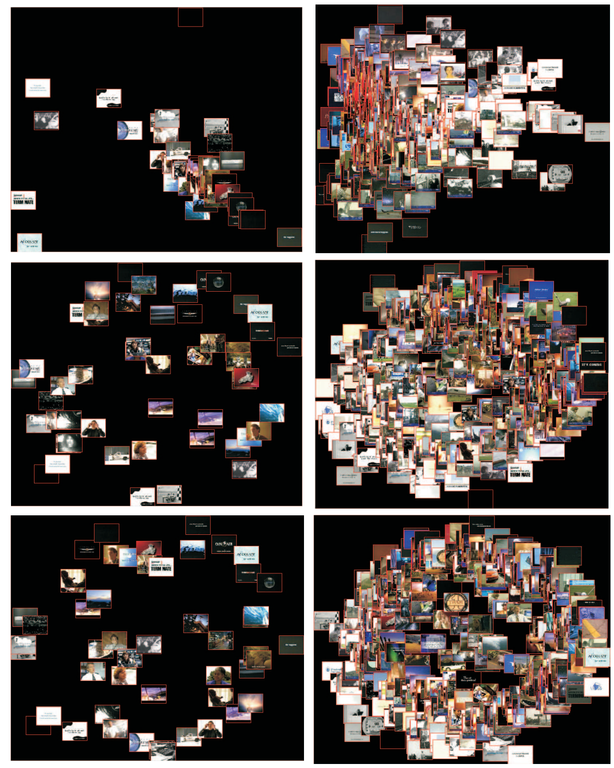

Figure 4: Results of visualizing 50, and 500 images in 2D space. From top down are results with original $\lambda = 0$, $\lambda = 0.1$, and $\lambda = 1.0$.

5. ACKNOWLEDGMENT

We would like to thank Sjaak Verbeek for his discussion on the use of SNE algorithm and the IOP project I'mIK.

6. REFERENCES

[1] G. Hinton and S. Roweis. Stochastic neighbor embedding. *Neural Information Processing Systems 15.*, 2002.

[2] B. Moghaddam, Q. Tian, N. Lesh, C. Shen, and T. Huang. Visualization and user-modeling for browsing personal photo libraries. *IJCV*, 56(1-2):109–130, 2004.

[3] M. Nakazato and T. Huang. 3D MARS: Immersive virtual reality for content-based image retrieval. *Proc. of ICME*, 2001.

[4] Y. Rubner, C. Tomasi, and L. Guibas. A metric for distributions with applications to image databases. *Proc. of IEEE ICCV*, pages 59–66, 1998.

[5] S. Santini, A. Gupta, and R. Jain. Emergent semantics through interaction in image databases. *IEEE trans. of Knowledge and Data Engineering*, 2001.

[6] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of early years. *PAMI*, 22(12):1349–1380, 2000.

[7] J. Tenenbaum, V. Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science 290 (5500)*, pages 2319–2322, 2000.