Active Browsing using Similarity Pyramids

Jau-Yuen Chen[†], Charles A. Bouman[†] and John C. Dalton[‡ *]

[†]School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-1285
{jauyuen,bouman}@ecn.purdue.edu

[‡]Synthetik
San Francisco, CA 94103
johnd@synthetik.com

ABSTRACT

In this paper, we describe a new approach to managing large image databases which we call active browsing. Active browsing integrates relevance feedback into the browsing environment, so that users can modify the database's organization to suit the desired task. Our method is based on a similarity pyramid data structure which hierarchically organizes the database so that it can be efficiently browsed. At coarse levels, the similarity pyramid allows users to view the database as large clusters of similar images. Alternatively, users can "zoom into" finer levels to view individual images.

We discuss relevance feedback for the browsing process, and argue that it is fundamentally different from relevance feedback for more traditional search-by-query tasks. We propose two fundamental operations for active browsing: pruning and reorganization. Both of these operation depend on a user defined relevance set which represents the image or set of images desired by the user. We present statistical methods for accurately pruning the database, and we propose a new "worm hole" distance metric for reorganizing the database so that members of the relevance set are grouped together.

Keywords: Browse, Image database, Pyramids, Pruning, Cross Validation

1. INTRODUCTION

Many approaches for content based management of image databases have focused on query-by-example methods[1] in which an example image is presented and the database is searched for images with similar visual content. However, query-by-example techniques tend to quickly converge to a small set of images that may not be of interest to the user.[2]

Browsing environments offer an alternative to conventional query-by-example, but have received much less attention. Zhang and Zhong[3] proposed a hierarchical self-organizing map (HSOM) which used the SOM algorithm to organize a complete database of images into a 2-D grid. MacCuish *et al*[2] used multidimensional scaling (MDS) to organize images returned by queries while Rubner, Guibas, and Tomasi[4] used MDS for direct organization of a complete database. However, a disadvantage of both SOM and MDS is that they are vary computationally intensive when applied to large databases of images. In addition, the MDS algorithms do not impose any hierarchical structure on the database. Most recently, Craver, Yeo and Yeung[5] described a browsing method based on space-filling curves.

Previously, we proposed a data structure for browsing large databases of images which we call a similarity pyramid.[6] The similarity pyramid organizes large image databases into a three dimensional pyramid structure. Each level of the similarity pyramid contains clusters of similar images organized on a 2-D grid. As users move down the pyramid, the clusters become smaller, with the bottom level of the pyramid containing individual images. In addition, users can pan across at a single level of the pyramid to see images or image clusters that are similar. In previous research,[6,7] we described efficient algorithms for constructing good similarity pyramids.

*Appeared in the SPIE/IS&T conference on *Storage and Retrieval for Image and Video Database VII*, vol. 3656, pp. 144-154, San Jose CA, January 26-29, 1999.

Figure 1. An example of a similarity pyramid and its embedded quad-tree.

While relevance feedback has long been recognized as useful for data retrieval problems,[8] more recently it has attracted attention in content based image retrieval.[9–11] However, these studies have focused on using relevance feedback in search-by-query rather than browsing.

In this paper, we describe an *active browsing* approach which combines relevance feedback into the browsing environment so that users can more effectively manage and search large databases of images or video.[12,13] Our active browsing environment uses a similarity pyramid to organize the database into a 3 dimensional structure that the user can move through. But in addition, the environment allows the user to give feedback through the selection of a set of desirable images which we call the relevance set. The relevance set is then used to adaptively prune and reorganize the pyramid to best suit the user's task. In order to perform these basic pruning and reorganization functions, we use statistical estimation techniques based on cross-validation. The cross-validation approach is shown to give reliable performance independently of the database content and specific choice of similarity measures. We also introduce a new distance metric which groups images from the relevance set together in the similarity pyramid. We call this distance metric the "worm hole" distance because it connects all points in relevance set.

2. ACTIVE BROWSING

The structure of a similarity pyramid is illustrated in Fig. 1. Each level of the similarity pyramid contains clusters of similar images organized on a 2-D grid. As one moves down the pyramid, the clusters become smaller, with the bottom level of the pyramid containing individual images. In addition, users can pan across at a single level of the pyramid to see images or image clusters that are similar.

The similarity pyramid combines the advantages of both searching and browsing. In fact, at the bottom levels of the pyramid, an image is surrounded by the images that would be returned by a simple query-by-example. However, the pyramid structure allows the user to "zoom out" so that they can see a larger variety of query returns, or pan across to move toward a different set of images.

In this work, we combine relevance feedback with the browsing environment of a similarity pyramid. We will incorporate relevance feedback into two basic browsing functions: pruning and reorganization. Pruning removes images from the database that are not likely to be of interest to the user, and reorganization changes the structure of the similarity pyramid to facilitate the user's search.

Figure 2 shows the active browsing environment presented to the user. The similarity pyramid is shown to the left, and the set of relevant images, which we call the relevance set, is shown to the right.

The relevance set is a set of images that the user selects as they move through the pyramid. The user can incrementally add or remove images from the relevance set at any time during the browsing process. For browsing, the user's objective may be to locate all images from a desired class. In this case, the relevance set may contain dissimilar groupings of images that represent the variations that can occur among images in the class. As the user browses through the database, the relevance set also becomes a buffer or clip board which stores all the images of interest to the user.

Once users have defined a set of relevant images, they may associate the relevance set with a semantic label. These relevance sets may then be stored and recalled based on the semantic label. Figure 3 shows four examples of relevance sets, corresponding to the semantic labels "car", "women", "wedding", and "sport".

Figure 2. Active Browsing Environment: The top level of the similarity pyramid is shown to the left, and the set of relevant images (relevance set) is shown to the right. The user can incrementally add or remove images from the relevance set as they browse through the database.

Car Women Wedding Sport

Figure 3. The relevance sets / semantic classes of four manual picked class: Car, Women, Wedding, and Sport.

3. PRUNING

The objective of pruning is to remove images from the database that are not likely to be of interest to the user while retaining all or most potentially desirable images. This is useful since the reduced size of the pruned database makes it easier and more effective to browse.

While traditional queries methods attempt to find images that are likely to be of interest, pruning retains all images which are of possible interest, but at the expense of retaining many questionable images. Intuitively, pruning attempts to achieve high recall, but at the expense of precision; whereas traditional queries tend to emphasize precision over recall. The reduced precision of pruning is acceptable because the similarity pyramid structure allows the user to efficiently browse the resulting images.

Let $R = \{x_1, \cdots, x_M\}$ be a set of M relevant images that a user selects to be typical of images of interest. Then we would like to retain all images y in the database such that

$$d(y, R) \leq T$$

where $d(y, R)$ is the distance between y and the set of relevant images R, and T is a threshold. In this paper, we will use the nearest neighbor distance function

$$d(y, R) = \min_{i=1,\cdots,M} d(y, x_i) \tag{1}$$

where $d(y, x_i)$ is a simple histogram based distance function that incorporates color, edge and texture features.[6] The nearest neighbor distance function works well when the set of desired images forms disjoint clusters in the feature space. In this way, the relevance set may represent a diverse set of images that are primarily related through a semantic label.

Our objective is to select a threshold T that will have a high probability of retaining desired images, but will substantially reduce the size of the database. Stated more formally, let S be the set of desired images. Then define the accuracy of the pruning to be

$$\text{Accuracy} = P\{d(y, R) \leq T | y \in S\} \ .$$

Here accuracy has the same meaning as recall, but we use the word accuracy because it has a more intuitive meaning in the application of pruning.

The problem is then to choose T in a way that achieves a fix level of pruning accuracy. To do this, we use a cross-validation strategy to compute the threshold value T. Let $R_i = R - \{x_i\}$ be the set of relevance images with the image x_i removed. Then we define

$$\begin{aligned} d_i &= d(x_i, R_i) \\ \bar{d}_y &= d(y, R_1) \\ d_y &= d(y, R) \end{aligned}$$

where $y \in S$ is a desired image. Furthermore, we will denote the order statistics of d_i as $d_{(i)}$; so that $d_{(i)} \leq d_{(i+1)}$.

The basic assumption of our analysis will be that the random variables d_i and $\bar{d}_y$ are independent and identically distributed. Under this assumption, it can be shown using standard methods from order statistics that (see Appendix A)

$$P\{\bar{d}_y \leq d_{(n)}\} = \frac{n}{M+1}$$

Since we are using a nearest neighbor distance, the inequality $d_y \leq \bar{d}_y$ always holds. Therefore,

$$P\{d_y \leq d_{(n)}\} \geq \frac{n}{M+1} \ . \tag{2}$$

From (2), we see that if the pruning threshold is set to $T = d_{(n)}$ then the pruned database will contain on average $\frac{n}{M+1}$ of all the desired images. So for example, if the relevance set contains $M = 16$ images, and $d_{(16)}$ is used as the threshold, then on average $\frac{16}{17}$ or 94% of the desired images will be retained in the pruning. Importantly, this result does not depend on the statistics of database or the properties of the metric $d(y, x)$.

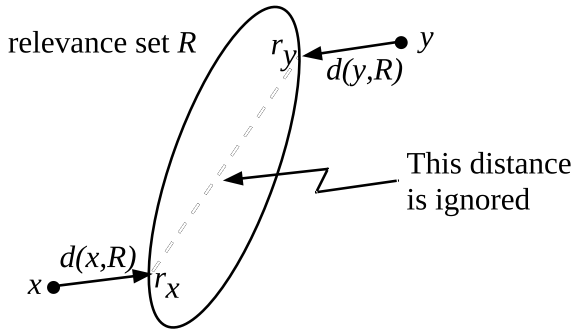

Figure 4. The illustration of "worm hole" distance. The distance from r_x to r_y is ignored.

4. REORGANIZATION

In addition to pruning, relevance feedback can be used to reorganize the database so the images of interest are grouped together. As with pruning, this reorganization makes the browsing process more efficient. Since the desired images often form disjoint clusters that are spread through the feature space, they will often be spread into disjoint groupings in the similarity pyramid. The reorganization step helps to move these desired images near one another in the similarity pyramid structure. While reorganization may be performed at any point in the browsing process, it is typically performed after pruning.

The following sections describe two alternative methods for reorganizing the similarity pyramid. Both methods depend on computation of a new dissimilarity metric based on the images in the relevance set.

4.1. Optimized Distance Function

For this section, we will assume that the distance function $d_\theta(y,x)$ is parameterized by a vector θ. In this work, θ is a 9 component vector containing the weightings corresponding to the L, a and b components of the color, edge and texture histogram features.[6]

The first step to reorganization is to compute the parameter $\hat{\theta}$ which optimizes the performance of the distance function on the relevance set R. To do this, we define a cost function $C(\theta,R)$ which uses cross-validation to estimate how well the distance function clusters the images in the relevance set. With this cost function, we then compute

$$\hat{\theta} = \arg\min_{\theta} C(\theta,R) \ . \tag{3}$$

The minimization of (3) is done using conjugate gradient optimization. The resulting distance function $d_{\hat{\theta}}(y,x)$ is then used to rebuild the similarity pyramid. Because this parameter θ is selected to minimize the cost function $C(\theta,R)$, the reorganized pyramid does a better job of clustering together desired images.

To define $C(\theta,R)$, we first randomly select K images out of the pruned database to form the set $Y=\{y_i,\cdots,y_K\}$. Then define $d_{ij}=d_\theta(y_j,R_i)$ and let $d_{i(j)}$ be the j^{th} order statistic of $\{d_{ij}\}_{j=1\cdots K}$, so that $d_{i(j)} \leq d_{i(j+1)}$. For the boundary case, further assume that $d_{i(0)}=0$. Recalling that $d_i=d(x_i,R_i)$, then we define π_i as the rank of d_i such that $d_{i(\pi_i)} \leq d_i \leq d_{i(\pi_i+1)}$. Using these definitions, we define the continuously valued rank of d_i as

$$r_i = \pi_i + \frac{d_i - d_{i(\pi_i)}}{d_{i(\pi_i+1)} - d_{i(\pi_i)}}. \tag{4}$$

Notice that (4) is chosen so that r_i is a continuous function of d_i. This is important since it will facilitate optimization. The total cost function is then given by

$$C(\theta,R) = \sum_{i=1\cdots M} r_i \tag{5}$$

4.2. Worm Hole Distance

In this section, we propose an alternative method to optimize the organization of the similarity pyramid. Rather then using the linear transformation of Section 4.1, we focus on nonlinear transformations of the dissimilarity metric. Our objective is to nonlinearly warp the feature space so that images in or near the relevance set are near each other. This distorted distance function is then used to rebuild the similarity pyramid so that images similar to the relevance set are grouped together.

In order to warp the feature space, we will define a new distance metric which can be easily computed from the initial distance metric $d(x, y)$. The new distance metric $d(x, y; R)$ depends on the relevance set R and is defined by

$$d(x, y; R) = \min\{d(x, y), d(x, R) + d(y, R)\} \tag{6}$$

where $d(x, y)$ is the dissimilarity function between images x and y, and $d(x, R)$ is the nearest neighbor distance between x and R defined by

$$d(x, R) = \min_{y \in R} d(x, y) \ .$$

We call this new metric of (6) the "worm hole" distance because points in the set R are considered to be collocated in the space. Figure 4 illustrates how the new metric works. Notice that the shortest distance between two points can include a short cut or worm hole through the relevance set. Since the set R may not be connected in the feature space, points that were originally far apart may be close in the new distorted space. In addition, for any two points $r_1, r_2 \in R$ and $d(r_1, r_2; R) = 0$.

Importantly, we can show that the worm hole distance is a metric whenever the original distance $d(x, y)$ is metric. (Here we treat all the points in R as an equivalence class since $d(r_1, r_2; R) = 0$ for $r_1, r_2 \in R$.) That is $d(x, y; R)$ is positive, symmetric, and obeys the triangle inequality as shown in Appendix B. This is important because it means that the worm hole distance can be used with fast search and clustering methods that depend on the use of the triangle inequality.[7,6]

In some cases, it may be useful to use more than one relevance set to distort the feature space. Let $d(x, y; R_1, \cdots, R_K)$ be a worm hole distance based on K relevance sets R_1 though R_K. The we define this distance recursively using the equation

$$d(x, y; R_1, \cdots, R_K) = \min\{d(x, y; R_1, \cdots, R_{K-1}), d(x, R_K; R_1, \cdots, R_{K-1}) + d(y, R_K; R_1, \cdots, R_{K-1})\}$$

where we use the notation $d(x, R; R_1, \cdots, R_{K-1})$ to mean

$$d(x, R; R_1, \cdots, R_{K-1}) = \min_{y \in R} d(x, y; R_1, \cdots, R_{K-1}) \ .$$

In Appendix B it is shown that this new distance function is also a metric.

5. EXPERIMENTAL RESULTS

We use a Corel Photo CD database consisting of 40,000 images evenly divided among 400 titles or classes. In this work, we assume each title to be a semantic class. For all experiments, the relevance sets were formed by randomly selecting images from each class. Tests were then performed on the remaining images in each class.

Figure 5 compares the recall performance for nearest neighbor distance to the relevance set, and the distance to the centroid of the relevance set. The experiment was performed by averaging over all classes and with relevance set sizes of 1, 4, 16, and 64 images. Notice that nearest neighbor distance not only performs uniformly better than the centroid distance method, but it continues improving with increasing size of the relevance set. This is because these largely semantic classes tend to contain images which form disjoint groupings in the feature space.

Figure 6 illustrates the theoretic and experimental pruning accuracy (recall) when using the order statistic threshold described in Section 3. This experiment is performed by averaging over all classes and with relevance sets of size 16. Notice that the experimental curve is very consistent with the theoretical lower bound predicted by (2).

Figure 7 illustrates the mean of pruned database size with theoretical accuracy = 50% for four typical classes: *Car Racing, Chicago, Steam Train, Wedding.* Each plot was formed by averaging 10 random experiments. Notice that the size of the pruned database generally decreases with the size of the relevance set, but it decreases more

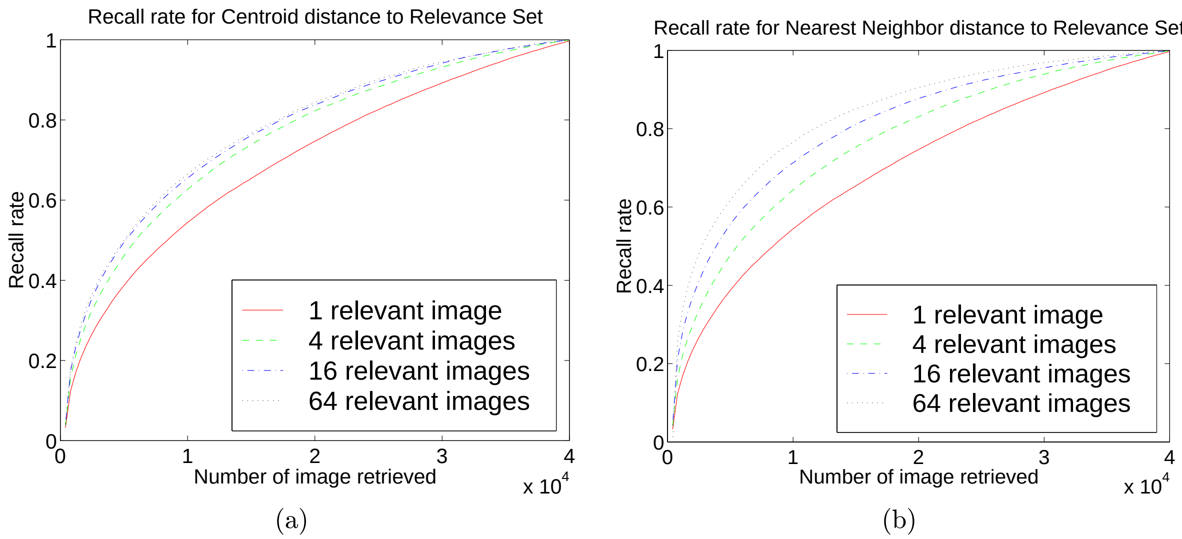

Figure 5. The recall accuracy for (a) centroid distance, and (b) nearest neighbor distance. The performance of nearest neighbor distance improves with the increasing size of the relevance set.

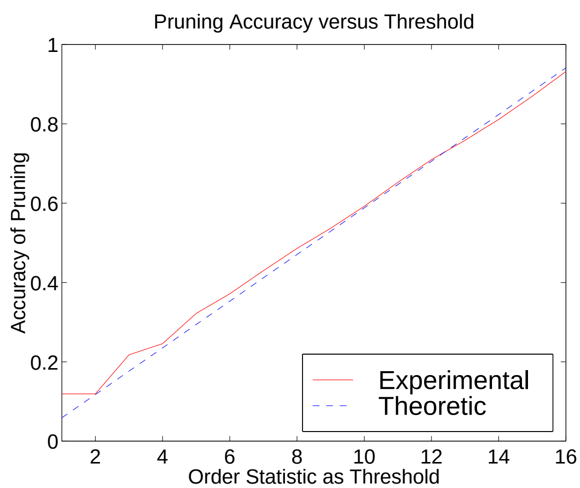

Figure 6. The theoretic and experimental accuracy versus n the rank of the order statistic used for pruning.

rapidly for some classes than others. For example, classes such as "car racing" tend to be more semantically related, so a larger relevance set is useful. Alternatively, classes such as "wedding" only require a small number of images in the relevance set to accurately model the distribution in the feature space.

To evaluate the performance of reorganization, we applied the *dispersion* measure described in[6] for evaluating the quality of pyramid organization. The dispersion measures the average normalized distance between similar images in the pyramid. Figure 8 shows the dispersion of the relevance set in the similarity pyramid before and after reorganization. Notice that the dispersion after reorganization is reduced, indicating that the images of interest are grouped more closely in the reorganized pyramid. This experiment was performed by averaging over all classes in the database and with relevance set sizes of 16.

Figure 9 shows the results of reorganization of the similarity pyramid using the worm hole distance of Section 4.2. The black rectangle is drawn around the 16 images contained in the relevance set R. Notice that the images near the relevance set match both the semantic and visual content of the relevance images quite closely. Interestingly, this happens even though the members of the relevance set differ substantially in both color and texture.

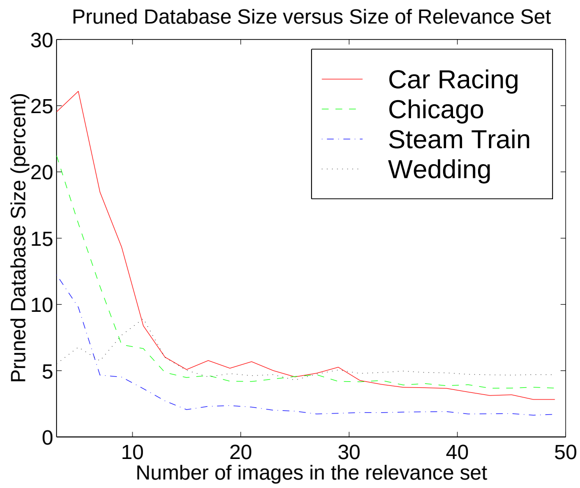

Figure 7. The mean of the pruned database size versus the number of images in the relevance set. All examples use a pruning accuracy (recall) of 50%.

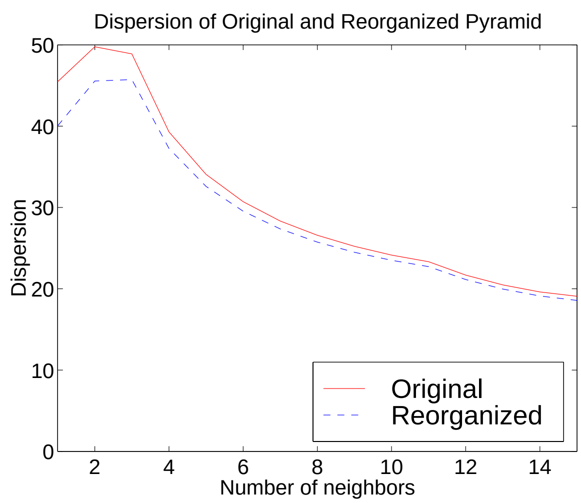

Figure 8. The dispersion of similarity pyramid before and after reorganization.

6. CONCLUSION

In this paper, we propose methods incorporating relevance feedback into the browsing process. Our approach is to allow the user to select images as they move through the database and copy them to a clipboard. We call this collection of images a relevance set, and use it to both prune and reorganize the database. Pruning is done using a nearest neighbor distance and a threshold based on cross-validation. We propose two methods for reorganization. Both methods depend on the rebuilding of the pyramid with a new, more relevant distance function. The first method optimizes the distance metric to best separate the relevant images from the remaining images in the database. The second method is based on a “worm hole” distance which groups images of the relevance set together.

APPENDIX A. ORDER STATISTIC

In this section, we compute $P\{\bar{d}_y \leq d_{(n)}\}$, the probability function we discussed in Section 3.

Assume $\bar{d}_y$ and $d_i, i = 1 \cdots M$ are i.i.d. with $P\{d_i \leq t\} = F(t)$ and $d_{(i)}$ is the i^{th} order statistic of $\{d_i\}_{i=1 \cdots M}$, with $P\{d_{(n)} \leq t\} = F_{d_{(n)}}(t)$.

Figure 9. A result of reorganization with worm hole distance. The images inside rectangle are members of relevance set.

It is well known that

$$dF_{d_{(n)}}(d_{(n)}) \;=\; \frac{M!}{(n-1)!(M-n)!}F(d_{(n)})^{n-1}(1-F(d_{(n)})^{n-j}dF(d_{(n)})$$

Under these assumption, $P\{\bar{d}_y \leq d_{(n)}\}$ can be found by:

$$\begin{aligned}
P\{\bar{d}_y \leq d_{(n)}\} &= E_{d_{(n)}}[P\{\bar{d}_y \leq d_{(n)}|d_{(n)}\}] \\
&= E_{d_{(n)}}[F(d_{(n)})] \\
&= \int F(d_{(n)})dF_{d_{(n)}}(d_{(n)}) \\
&= \int F(d_{(n)})\frac{M!}{(n-1)!(M-n)!}F(d_{(n)})^{n-1}(1-F(d_{(n)})^{n-j}dF(d_{(n)}) \\
&\quad \text{(But } y = F(X) \text{ is a uniform distribution on [0, 1])} \\
&= \int_0^1 t\frac{M!}{(n-1)!(M-n)!}t^{n-1}(1-t)^{n-j}dt \\
&\quad \text{(This is the expectation of beta(n, n-j+1) distribution)} \\
&= \frac{n}{M+1}
\end{aligned}$$

APPENDIX B. WORM HOLE DISTANCE

In this appendix, we show that the worm hole distance introduced in Section 4.2 is a distance metric. Let Ω be a space, let $R \subset \Omega$, and let Ω_R be a smaller space in which the points of R are treated as an equivalence class.

THEOREM B.1. *Let $d(x,y)$ be a metric on the space Ω and let $R \subset \Omega$ be a finite set of points in the space. Then $d(x,y;R) = \min\{d(x,y), d(x,R) + d(y,R)\}$ is a metric on the space Ω_R.*

proof:

We need to show that $d(x,y;R) \geq 0$,

- $\forall x \neq y \in \Omega_R,\ d(x,y;R) > 0$
- $\forall x \in \Omega_R,\ d(x,x;R) = 0$
- $d(x,y;R) = d(y,x;R)$
- $d(x,y;R) + d(y,z;R) \geq d(x,z;R)$

The first three properties are easily verified. However, we need to verify that the triangle inequality holds. First notice that

$$d(x,y;R) + d(y,z;R) = \min\{\underbrace{d(x,y)}_{(term1)}, \underbrace{d(x,R) + d(y,R)}_{(term2)}\} + \min\{\underbrace{d(y,z)}_{(term3)}, \underbrace{d(y,R) + d(z,R)}_{(term4)}\} \tag{7}$$

There are four cases for the two *min* functions in (7). For each case, we will show that $d(x,z;R)$ is a lower bound.

Case 1: Assume the minima are achieved by terms 1 and 3.

$$\begin{aligned} d(x,y;R) + d(y,z;R) &= d(x,y) + d(y,z) \\ &\geq d(x,z) \\ &\geq \min\{d(x,z), d(x,R) + d(z,R)\} \\ &= d(x,z;R) \end{aligned}$$

where the first inequality is simply the triangle inequality of metric $d(x,y)$ and the second inequality is the result of *min* operator.

Case 2: Assume the minima are achieved by terms 2 and 4.

$$\begin{aligned} d(x,y;R) + d(y,z;R) &= d(x,R) + d(y,R) + d(y,R) + d(z,R) \\ &\geq d(x,R) + d(z,R) \\ &\geq \min\{d(x,z), d(x,R) + d(z,R)\} \\ &= d(x,z;R) \end{aligned}$$

Case 3: Assume the minima are achieved by terms 2 and 3.

Let r_x, r_y and r_z be chosen so that

$$\begin{aligned} d(x,r_x) &= \min_{t \in R} d(x,t) \\ d(y,r_y) &= \min_{t \in R} d(y,t) \\ d(z,r_z) &= \min_{t \in R} d(z,t)\ . \end{aligned}$$

Then we have that

$$\begin{aligned} d(x,y;R) + d(y,z;R) &= d(x,R) + d(y,R) + d(y,z) \\ &= d(x,r_x) + d(y,r_y) + d(y,z) \\ &\geq d(x,r_x) + d(r_y,z) \\ &\geq d(x,r_x) + d(r_z,z) \\ &= d(x,R) + d(z,R) \\ &\geq \min\{d(x,z), d(x,R) + d(z,R)\} \\ &= d(x,z;R)\ . \end{aligned}$$

Case 4: The minima are achieved by terms 1 and 4.

This proof is similar to case 3.
QED

We next show that the distance function $d(x, y; R_1, \cdots, R_K)$ is a distance metric on the space $\Omega_{R_1,\cdots,R_K}$. Here $\Omega_{R_1,\cdots,R_K}$ is a smaller version of the space Ω where each set of points R_k is treated as an equivalence class.

THEOREM B.2. *Let $d(x, y)$ be a metric on the space Ω and let $R_k \subset \Omega$ be a finite set of points in the space for $k = 1, \cdots, K$. Then $d(x, y; R_1, \cdots, R_K)$ is a metric on the space $\Omega_{R_1,\cdots,R_K}$.*

proof:

We only need to verify that $d(x, y; R_1, \cdots, R_K)$ obeys the triangle inequality. The other properties are easily checked.

Our proof is by induction on K. We have already shown this property for $K = 1$. Assume it holds for $K - 1$, and define $\bar{d}(x, y) = d(x, y; R_1, \cdots, R_{K-1})$. Then $\bar{d}(x, y)$ is a metric, and $d(x, y; R_1, \cdots, R_K) = \bar{d}(x, y; R_K)$ must therefore be a metric by the previous theorem.
QED

REFERENCES

1. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image content: The QBIC system," *IEEE Computer* **28**, pp. 23–31, September 1995.
2. J. MacCuish, A. McPherson, J. Barros, and P. Kelly, "Interactive layout mechanisms for image database retrieval," in *Proc. of SPIE/IS&T Conf. on Visual Data Exploration and Analysis III*, vol. 2656, pp. 104–115, (San Jose, CA), Jan 31-Feb 2 1996.
3. H. Zhang and D. Zhong, "A scheme for visual feature based image indexing," in *Proc. of SPIE/IS&T Conf. on Storage and Retrieval for Image and Video Databases III*, vol. 2420, pp. 36–46, (San Jose, CA), February 9-10 1995.
4. Y. Rubner, L. Guibas, and C. Tomasi, "The earth mover's distance, multi-dimensional scaling, and color-based image retrieval," in *Proceedings of the ARPA Image Understanding Workshop*, May 1997.
5. S. Craver, B.-L. Yeo, and M. M. Yeung, "Image browsing using data structure based on multiple space-filling curves," in *To appear in the Proceedings of the Thirty-Sixth Asilomar Conference on Signals, Systems, and Computers*, (Pacific Grove, CA), November 1-4 1998.
6. J.-Y. Chen, C. A. Bouman, and J. Dalton, "Similarity pyramids for browsing and organization of large image databases," in *Proc. of SPIE/IS&I Conf. on Human Vision and Electronic Imaging III*, vol. 3299, (San Jose, CA), January 26-29 1998.
7. J.-Y. Chen, C. A. Bouman, and J. P. Allebach, "Fast image database search using tree-structured vq," in *Proc. of IEEE Int'l Conf. on Image Proc.*, vol. 2, pp. 827–830, (Santa Barbara, CA), October 26-29 1997.
8. G. Salton, *The SMART Retrieval System*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971.
9. I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos, "Pichunter: Bayesian relevance feedback for image retrieval," in *Proceedings of International Conference on Pattern Recognition*, vol. 3, pp. 361–369, (Vienna, Austria), August 1996.
10. L. Taycher, M. L. Cascia, and S. Sclaroff, "Image digestion and relevance feedback in the Imagerover WWW search engine," in *Proceedings of International Conference on Visual Information*, (San Diego, CA), December 15-17 1997.
11. Y. Rui, T. S. Huang, and S. Mehrotra, "Relevance feedback techniques in interactive content-based image retrieval," in *Proc. of SPIE/IS&T Conf. on Storage and Retrieval for Image and Video Databases VI*, vol. 3312, pp. 25–36, (San Jose, CA), January 26-29 1998.
12. J.-Y. Chen, C. Taşkiran, E. J. Delp, and C. A. Bouman, "ViBE: A new paradigm for video database browsing and search," in *Proc. of IEEE Workshop on Content-Based Image and Video Databases*, pp. 96–100, (Santa Barbara, CA), June 21 1998.
13. C. Taskiran, J.-Y. Chen, C. A. Bouman, and E. J. Delp, "A compressed video database structured for active browsing and search," in *Proc. of IEEE Int'l Conf. on Image Proc.*, vol. III, (Chicago, IL), October 4-7 1998.