Incremental Feature Weight Learning and its application to a Shape-based Query System

Kyoung-Mi Lee[1] and W. Nick Street[2]

[1]Department of Computer Science,
The University of Iowa, Iowa City, IA 52242, USA.
E-mail: klee1@cs.uiowa.edu

[2]Department of Management Sciences,
The University of Iowa, Iowa City, IA 52242, USA.
E-mail: nick-street@uiowa.edu

Abstract: Similarity between shapes is often measured by computing the distance between two feature vectors. Unfortunately, the feature space cannot always capture the notion of similarity in human perception. So, most current image retrieval systems use weights measuring the importance of each feature. However, the similarity does not vary with equal strength or in the same proportion in all directions in the feature space. In this paper, we present feature weights based on both clustered objects in the database and on relevance feedback. We show that using variance information from shape clusters to guide cluster information for an initial database search gives better results than using the standard Euclidean distance. To automatically incorporate a user's need, the proposed shape-based query system uses an incremental feature weight learning method that refines prototypes. In contrast to existing image database systems, the system can learn from user feedback. Indexing and retrieval results are presented that demonstrate the efficacy of our technique using the well-known Columbia database.

Key words: Shape-based query, Incremental clustering, Weighted feature distance, Prototype refinement

1 Introduction

The tremendous growth in the number and size of digital image collections is making necessary the development of new tools for indexing and retrieval. In general, queries based on content similarity to an example (object or image) in terms of features (such as color, texture, shape, etc.) are known as Query-by-Example. The image database retrieval procedure looks for the most relevant examples in a multidimensional space defined by the features under a given similarity metric. However, features are unequal in their importance for computing the similarity between examples. So most current image retrieval systems use weights measuring the importance of each feature. However, the similarity does not vary with equal strength or in the same proportion in all directions in the feature space emanating from a given query. Feature weights may change from example to example and from location to location [1].

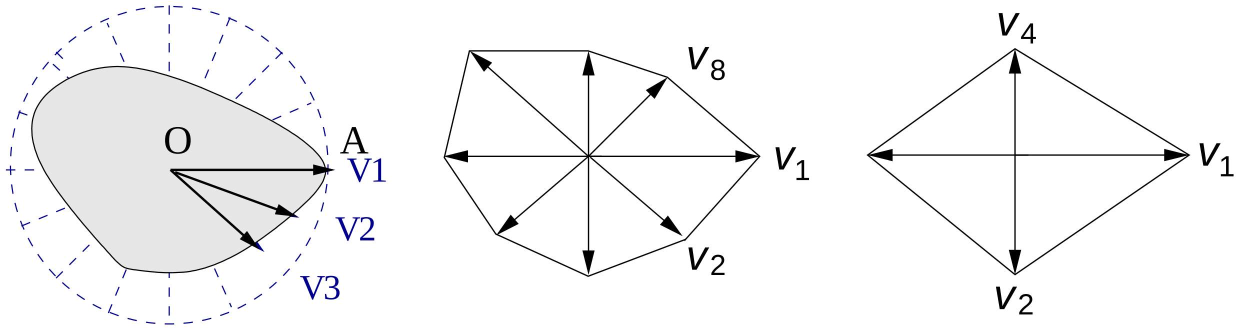
Similarity in human perception cannot always be captured by the extracted features. To give the user more control over the search criteria, some systems allow the user to assign weights to the features [11, 3]. This manual adjustment is not generally applicable because it is time consuming, and because the user does not usually have a detailed understanding of the features. This paper presents an automatic feature mining method for shape-based image retrieval.

Recently, relevance feedback has been the most commonly applied method for calculating feature weights incorporating the user's preference. An initial search is done in the database using the original query input by the user. Upon being presented the results of this search, the user labels some of these results as relevant or irrelevant according to the user's information needs. Cox *et al.* viewed the relevance feedback in a Bayesian framework that incorporates a explicit model of the user's selection process [7]. Rui *et al.* used a vector space model where a new query feature vector is generated as a weighted linear combination of the original feature vector and the feature vectors of the images that were labeled as relevant or irrelevant by the user [13, 14]. Peng *et al.* proposed probabilistic feature relevance learning to update weights iteratively, also based on relevance feedback [12]. Aksoy *et al.* used the ratio of standard deviations of the feature values both for the whole database and also among the images selected as relevant by the user [2].

In this paper, we propose feature weights based on both the clustered database and on relevance feedback. Clustering the database can significantly reduce the retrieval time by limiting the search range. Moreover, feature weights with cluster information give better initial search than flat feature weights. Relevance feedback from previous retrievals improves the retrieval performance by localizing the search within a small number of clusters. The rest of the paper is organized as follows. Section 2 presents a shape model used in this paper. In Sections 3, 4 and 5, incremental clustering, weighted feature distance, and prototype refinement are introduced. The indexing and retrieval performance of the shape-based query system are presented in Section 6. Section 7 concludes the paper.

2 Shape model

To model the shapes of a set of variable objects, the essential components of the proposed learning system use a Point Distribution Model (PDM) [6]. A PDM is a statistical shape model and is generated from a training set of N object descriptions $(\mathbf{v}^n)^{n=1\cdots N}$. An object description, $\mathbf{v}^n$, is simply a labeled set of I points $(v_i^n)_{i=1\cdots I}$. To use shapes as input data of a learning system, these points must be positioned consistently on each object in the training set.



(a) Query (b) 8D (c) 4D

Figure 1: Shape-based query

In this paper, we adopt the centroid-radii model [8, 15]. The shape is represented as a vector, $\mathbf{v} = (v_1, v_2, \cdots, v_I), = (r_{\theta_1}, r_{\theta_2}, \cdots, r_{\theta_I})$, where I is the number of features and v_i is the ith radius from the centroid to the boundary of the shape (Figure 1 (a)). Assume that OA is an arbitrary radius of the shape. Starting from OA and moving clockwise, we divide the circle into I equal arcs to place points around the boundary, with the regular interval being $\theta = \frac{360}{I}$ degrees.

This shape model can facilitate multiresolution representation and multiresolution query-

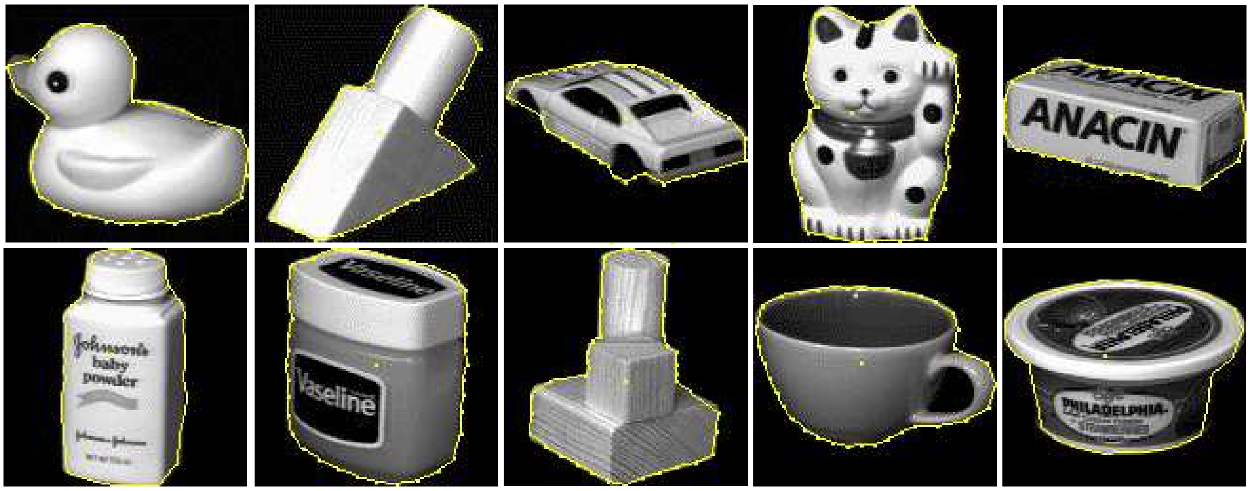


Figure 2: Shape model with 36 radii in the Columbia database

ing. For example, a shape with 8 features (Figure 1 (b)) is more precise than a shape with 4 features (Figure 1 (c)). Clearly, the more features are used to represent a shape, the less shapes match such query. With a sufficient number of radii, dissimilar shapes can be differentiated from each other. Figure 2 shows examples of the shape model with 36 radii.

3 Incremental clustering

Clustering is used to accelerate query processing by considering only a small number of representatives of the clusters, rather than the entire database. A given set of N shapes will be partitioned into C clusters such that shapes within the same cluster have a high degree of similarity, while shapes belonging to different clusters have a high degree of dissimilarity. Each of these clusters is represented by a prototype $(\mathbf{P}^c)^{c=1 \cdots C}$. Given the set of N^c shapes in the cluster $\mathbf{c}$, each feature P_i^c of the prototype can be computed by averaging the shape features that belong to the cluster. To measure the spread of a set of data around the center of the data in the cluster, we use the standard deviation (σ).

A major issue that databases are now facing is the extremely high rate of update. However, existing clustering algorithms are not suitable for maintaining clusters in such an environment, and they have been struggling with the problem of updating clusters without frequently performing complete re-clustering [5]. In this paper, we use an incremental learning method [4, 8] as follows. If an input shape is similar to an existing cluster or is grouped by the user explicitly, the shape is assigned to one of the existing clusters. If the input shape is dissimilar to all existing clusters or is designed as a new cluster by a user, the new cluster is created with the shape. Figure 3 shows the incremental clustering process graphically.

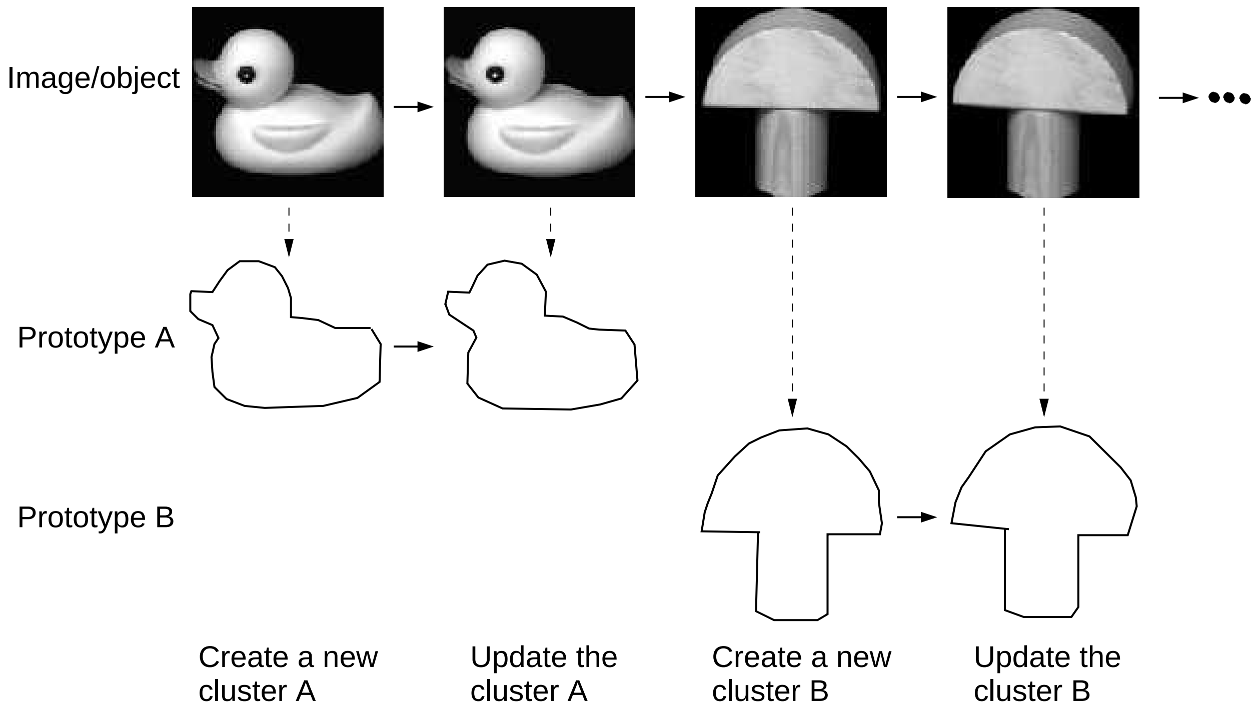


Figure 3: Incremental clustering of shapes

Obviously it is an important issue to measure the similarity between two shapes. The most common measure of similarity between two shapes, $\mathbf{v}$ and $\mathbf{u}$, is the distance between them. In I-dimensional Euclidean space, the distance between a prototype $\mathbf{P}^c$ and a given shape $\mathbf{v}$ is computed as

$$\mathbf{d}(\mathbf{P}^c, \mathbf{v}) = \sqrt{\sum_{i=1}^{I} |P_i^c - v_i|^2}, \tag{1}$$

where P_i^c and v_i are the ith features of $\mathbf{P}^c$ and $\mathbf{v}$, respectively.

During clustering, shapes are normalized for translation through the use of an object-centered coordinate system and for reflection through the use of the reflected shape. Shapes are normalized for size by scaling based on the longest radius. To avoid simple orientation differences, the distance is computed at different rotations and the final distance is defined to be the minimum of these distances, $\min_{1 \le j \le I} \mathbf{d}(\mathbf{P}^c, \mathbf{v}_{(j)})$ where $\mathbf{v}_{(j)}$ is the query rotated by the jth feature of $\mathbf{v}$.

4 Weighted Feature Distance

Image retrieval is a procedure to search images closest to a given query. In this section, we consider the problem of the distance metric used to measure similarity for shape-based retrieval. The scale of a feature can change the overall contribution of that feature in the distance computation [12]. However Eq. (1) does not take into account the influence of the scale of each feature in the distance computation. Thus, each feature should be rescaled so that all features contribute equally to the distance computation in Eq. (1).

To capture such scale information as weights, we propose a measure using the standard deviation. Figure 4 shows a case where there are four clusters in a two-dimensional feature space. In this example, the v_1 dimension is more important for queries a and b, since a slight move along the v_1-dimension may change the cluster assignment, and the v_2-dimensions of queries c and d is more important. Furthermore, even though the distance along the v_1-

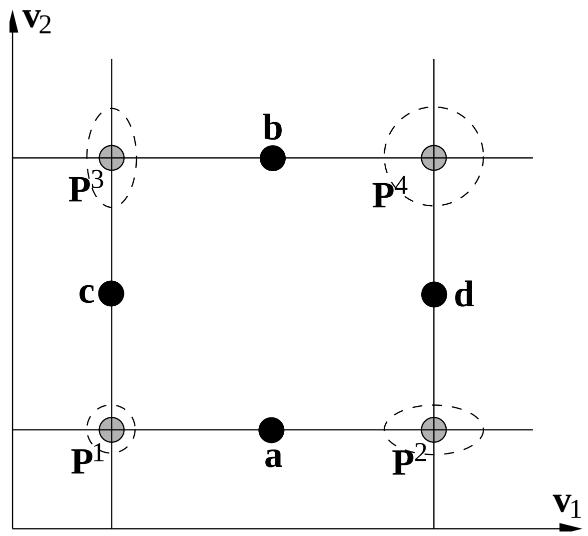


Figure 4: Feature scale varies with query locations in terms of distance. Black points represent queries, gray points represent object clusters, and dotted lines show the standard deviation of each cluster.

dimension from query a to clusters $\mathbf{P}^1$ and $\mathbf{P}^2$ is the same, as is that from b to $\mathbf{P}^3$ and $\mathbf{P}^4$, the calculated distance metrics are different because of the standard deviations of the clusters. Because the standard deviation of $\mathbf{P}^2$ is wider than that of $\mathbf{P}^1$ along v_1 and that of $\mathbf{P}^4$ is wider than that of $\mathbf{P}^3$, a is closer to $\mathbf{P}^2$ than $\mathbf{P}^1$, and b is closer to $\mathbf{P}^4$ than $\mathbf{P}^3$, respectively. Similarly, c is closer to $\mathbf{P}^3$ than $\mathbf{P}^1$, and d is closer to $\mathbf{P}^4$ than $\mathbf{P}^2$.

It is important to normalize each feature in the the shapes to the same range to ensure that each individual feature receives equal weight in determining the similarity between two shapes [13]. Therefore, we propose the following exponential weight

$$w_i^c = \begin{cases} 1 - \exp\left(-\dfrac{|P_i^c - q_i|}{\sigma_i^c}\right), & \text{if } \sigma_i^c \neq 0 \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

It is evident that $0 \leq w_i^c \leq 1$, where $w_i^c = 1$ indicates that q_i is too far from P_i^c or P_i^c is 0, and thus q_i isn't similar to P_i^c at all. On the other hand, $w_i^c = 0$ states P_i^c and q_i are identical, and so q_i is totally similar to P_i^c. Values in between show the degrees of weight that q_i exerts at i, reflecting the influence of $\mathbf{q}$ on the variation of $\mathbf{P}^c$ at i. In the distance function used as a similarity measure, it is desired to give a larger weight to a dissimilar shape. Thus, Eq. (2) can be used as weight for each dimension of the distance between a prototype and a shape. The distance between a prototype $\mathbf{P}^c$ and the query $\mathbf{q}$ can finally be computed as

$$\mathbf{d}_w(\mathbf{P}^c, \mathbf{q}) = \sqrt{\sum_{i=1}^{I} |w_i^c(P_i^c - q_i)|^2}. \tag{3}$$

These weights enable the distance computation to elongate dissimilar features, and, at the same time, to constrict the closest ones. For the actual retrieval, we define the distance between $\mathbf{q}$ and a shape $\mathbf{v}$ in $\mathbf{P}^c$ which is similar to $\mathbf{q}$ as

$$\mathbf{d}_w(\mathbf{v}, \mathbf{q}) = \sqrt{\sum_{i=1}^{I} |w_i^c(v_i - q_i)|^2}. \tag{4}$$

5 Incremental Feature Weight Learning

Relevance feedback is the process of automatically adjusting an existing query using information fed back by the user about the relevance of previously retrieved examples. We want to use relevance feedback to give the user more control on the search criteria. We also want to allow the database system to learn the user's needs. When the user looks for a shape similar to one he found previously, the system can give results using information learned by his feedback, instead of starting an initial search from scratch.

We introduce an incremental feature weight learning method which enables the user to refine prototypes by specifying over time a set of relevant results (Algorithm 1). At each step, the system begins by initializing the current database. Given a query $\mathbf{q}$, created by

1. Initialization: $m = 0$,

   $N^{c_0} = N^c$, $\mathbf{P}^{c_0} = \mathbf{P}^c$, and $\sigma^{c_0} = \sigma^c$,

   where $m = 0$ means there is no relevance feedback to $\mathbf{q}$.

2. Retrieval

   (a) Calculate $w_i^{c_m}$ using Eq. (2).

   (b) Search similar prototypes using $\mathbf{d}_w(\mathbf{P}^{c_m},\mathbf{q})$, Eq. (3).

   (c) Search relevant shapes in similar prototypes $\mathbf{P}^{c_m}$ using $\mathbf{d}_w(\mathbf{q},\mathbf{v})$, Eq. (4).

3. User relevance feedback: Select relevant shapes from retrieved images.

4. Prototype refinement: Modify each new prototype and standard deviation:

   $N^{c_{m+1}} = N^{c_m} + N_R^{c_m}$,

   $$P_i^{c_{m+1}} = \frac{N^{c_m} P_i^{c_m} + \sum_{k=1}^{N_R^{c_m}} R_i^{m_k}}{N^{c_{m+1}}}, \text{ and } \quad \sigma_i^{c_{m+1}} = \sqrt{\frac{N^{c_m} s_i^{c_m} + \sum_{k=1}^{N_R^{c_m}} (P_i^{c_{m+1}} - R_i^{m_k})^2}{N^{c_{m+1}}}},$$

   where N^{c_m} is the number of shapes in the cluster $\mathbf{c}$, $N_R^{c_m}$ the number of relevant shapes corresponding to the cluster $\mathbf{c}$, $R_i^{m_k}$ the ith feature of the kth relevant shape, and $s_i^{c_m} = (\sigma_i^{c_m})^2 + (P_i^{c_{m+1}} - P_i^{c_m})^2$, after the mth feedback, respectively.

5. $m = m + 1$ and go to step 2.

Algorithm 1: Incremental Feature Weight Learning

the user who draws a boundary on a sketch or an image, selection of shapes sharing similar characterizing elements starts by comparing $\mathbf{q}$ to the set of prototypes using Eq. (3). If some prototypes are selected as being similar to $\mathbf{q}$, only the shapes matching those prototypes are compared to the query using Eq. (4). The user labels the retrieved examples as relevant or irrelevant to the query $\mathbf{q}$. The system then has to estimate user relevance feedback, i.e. for each retrieved shape, decide its level of relevance to the query. The relevant shapes are clustered to the prototype $\mathbf{P}^c$ to make it satisfy the user's need better than the original query shape. Since the prototype ($\mathbf{P}^{c_{m+1}}$) is refined by the user's need, we refer to the process as *prototype refinement*. The next time the system is used, the user starts a search using $\mathbf{P}^{c_{m+1}}$, not $\mathbf{P}^c$. Through relevance feedback, each prototype can include information on what the user preferred previously. Whenever the system performs retrieval, the system recalculates Eq. (2) using the refined prototypes and thus reduces the search step.

6 Application to Shape-based Query System

We present experiments using the well-known Columbia database which contains 1440 images of 20 different objects: 72 images per object taken at 5^o in pose [10]. We used a 36-dimensional data space for the database, that is, each shape is represented by 36 radii (see Figure 2). **The performance in the experiments is evaluated using two measures: precision and recall. Recall is the ratio of relevant images retrieved to the total number in the database and precision is the ratio of relevant images retrieved to the total number of images retrieved.**

6.1 System Overview

Figure 5 shows the functional architecture of the proposed shape-based query system which is implemented in Java and so the user at the client can access the clusters at the server through the Internet. The user at the client can access the object database and prototypes at the server through the Internet to copy them initially. The user presents a query from a local or remote (www) file system. The client formulates the query, performing shape feature extraction. The query is submitted to the server. Based on the retrieved results, the user can supply his interests by selecting relevant results and the client uses this information to learn feature relevance.

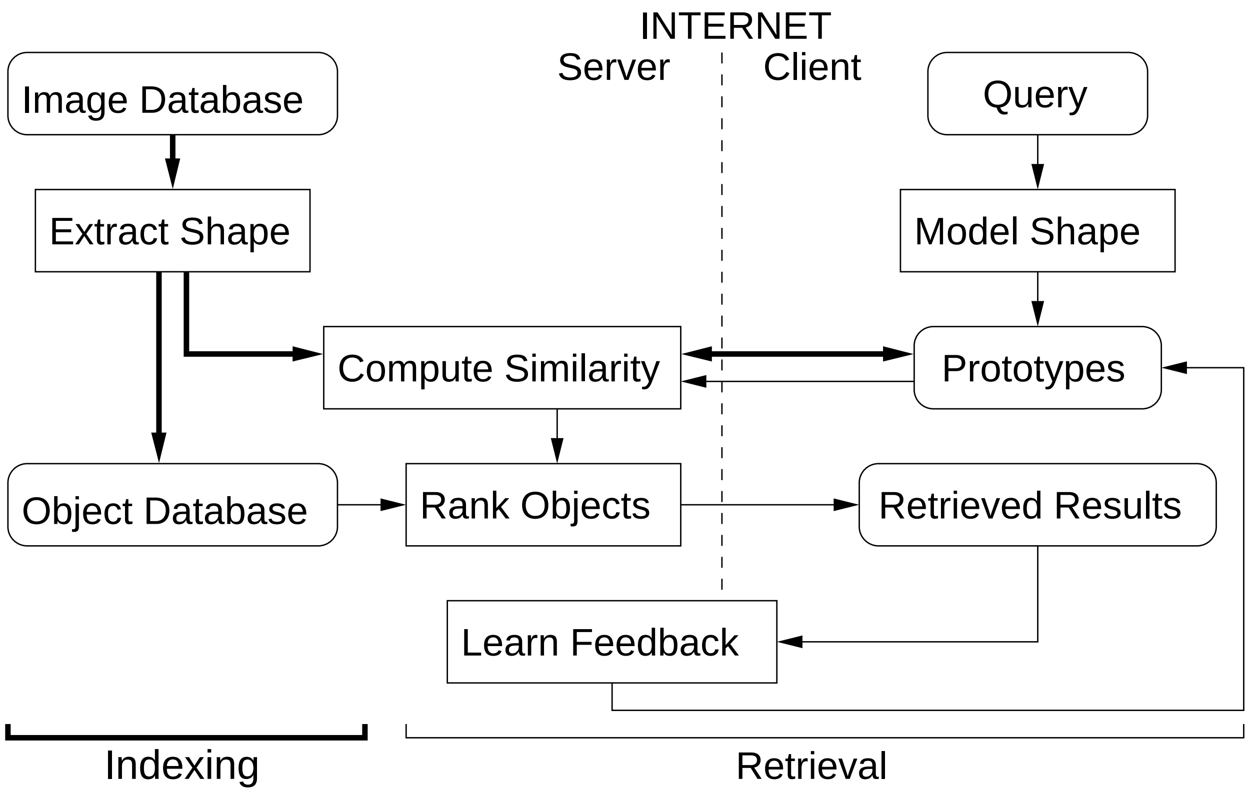


Figure 5: System overview: Thick arrows indicate the indexing process, thin arrows the retrieval process,

square boxes operations, and round boxes data.

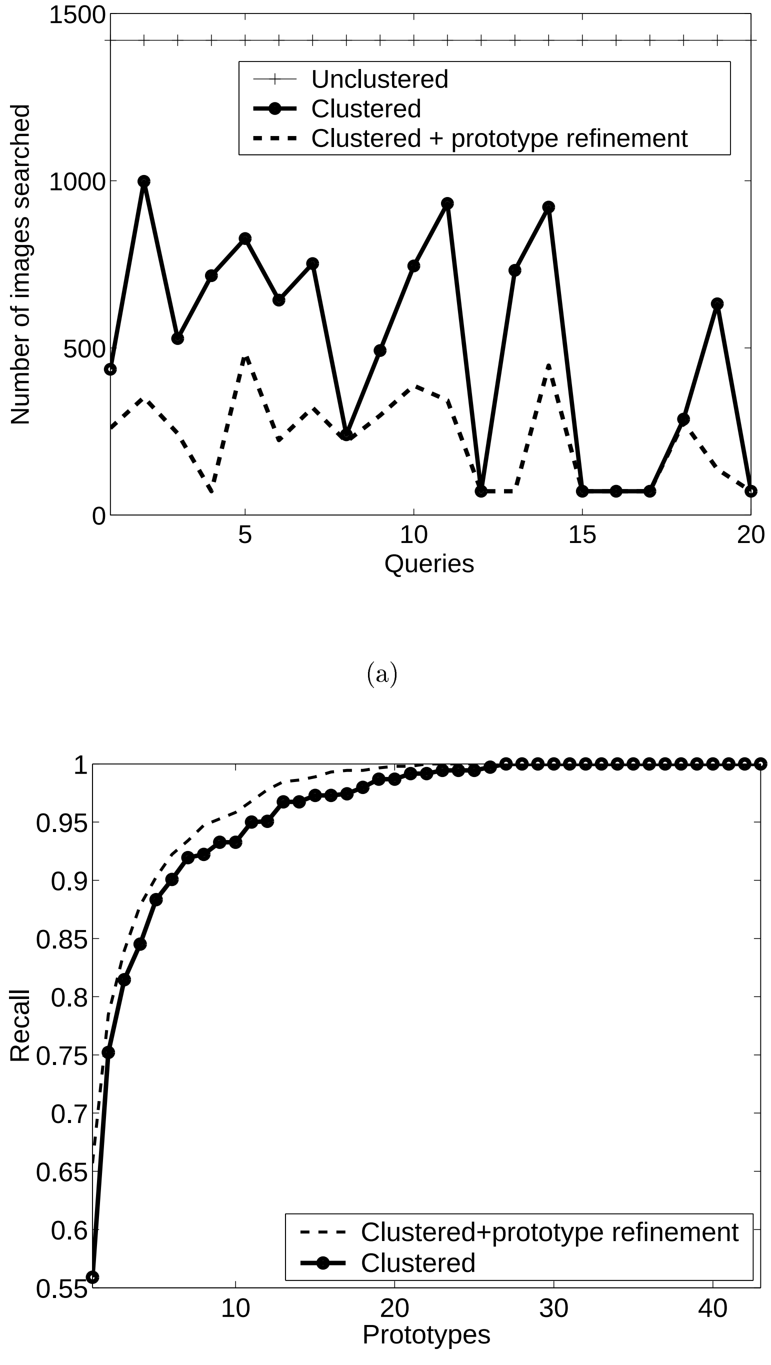
The server consists of two parts: the indexing and retrieval modules. Database indexing is performed off-line, incorporating incremental learning which aids indexing of an ever-increasing volume of data. The system carries out retrieval based on current similarity between the query and all prototypes first, and then between the selected prototypes and all shapes in them. From the user feedback, the system computes and incrementally learns feature weights in terms of similarity.

6.2 Shape-based Indexing

We used 1420 images for indexing and 20 images (45^o for each object) for query. The database was incrementally clustered and created 45 prototypes. These prototypes are used as an indexed key of the two-level indexed tree which consists of a prototype layer and an object layer. The prototype layer of the tree is a simple filter that reduces the search space quickly, discriminating the objects based on shape. To evaluate indexing effectiveness, we tested the retrieval performance after clustering.

Figure 6 (a) shows the number of images searched for a recall of 1.0 in the Columbia database. A recall of 1.0 is the case of finding 71 images of each object, though there may be different shapes at some angles. Instead of searching and ranking all shapes, the system searches only a small set of clusters and ranks shapes in the clusters. If some positive examples are missed, the system searches the next most relevant prototypes. So, if the 71 images of an object have the same shape with the query (as in the $12th$, $15th$, $16th$, $17th$, and $20th$ queries), all relevant images can be retrieved with the small number of retrieved images. If shapes of the object varies at different rotations, there some irrelevant images are retrieved. However, Figure 6 (a) illustrates that the clustering method usually reduces the number of images searched, often dramatically.

Figure 6 (b) shows how the number of prototypes affects the average performance for 20 queries performed on the database of the 1420 images in the Columbia database. With only the single most relevant prototype, 56.7% (about 40 images) of the relevant object (71 images) was retrieved on average. As the number of prototypes reached six, performance reaches approximately 90%.



(a)

(b)

Figure 6: Indexing effectiveness on Columbia database: (a) Number of images searched for recall of 1.0 per query and (b) Average retrieval performance vs. the number of prototypes searched

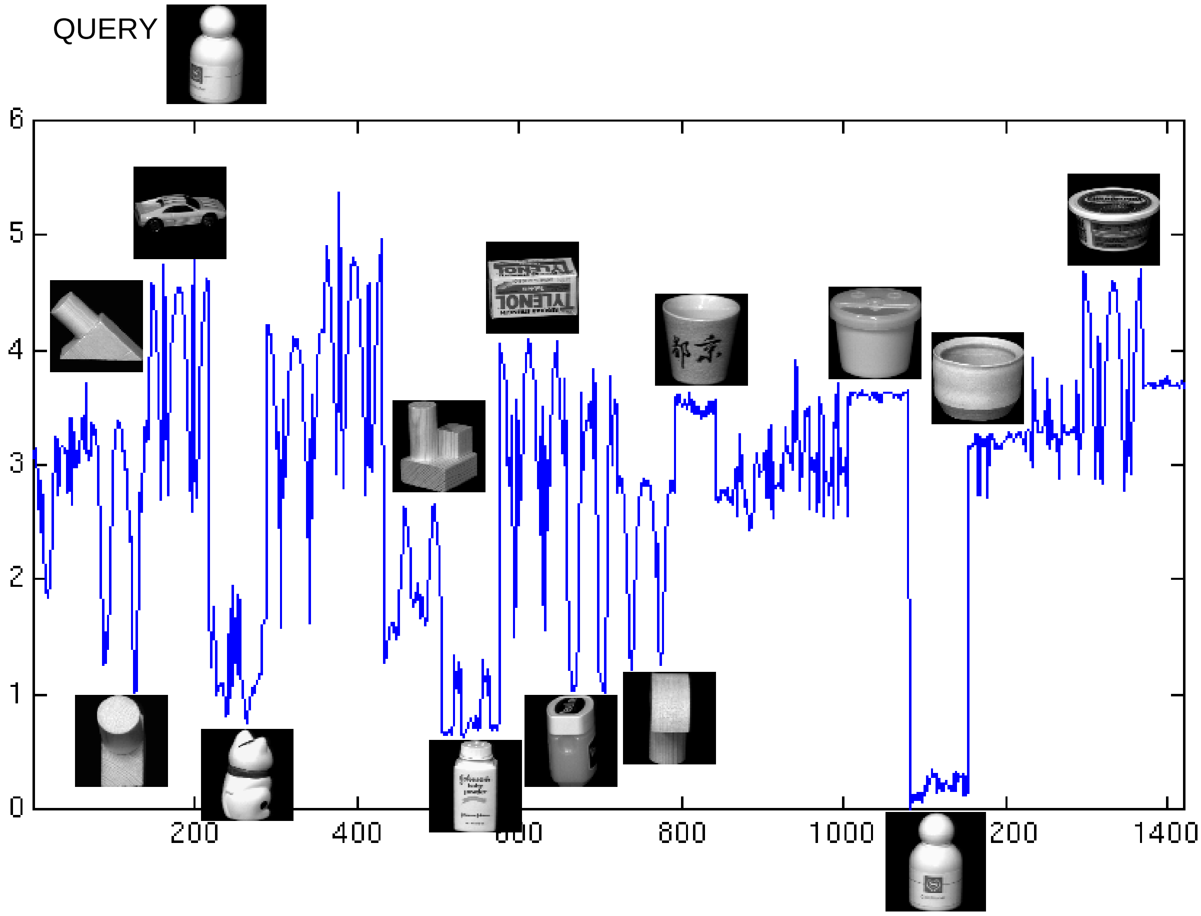


Figure 7: Distance between the Conditioner prototype and all indexed images

To visually evaluate this indexing, we plot distance between the sixteenth query (the Conditioner) which is a single shape for all views and all 1420 images using Eq. (4). Figure 7 shows that there are several different objects which have similar shapes at specific angles. However, all 71 images in the Conditioner prototype have a smaller distance to their own prototype than any of the other shapes.

6.3 Shape-based Retrieval

The proposed system first finds the six most similar prototypes to a query and then ranks shapes in these six clusters based on weighted distance from the queries. We measure the

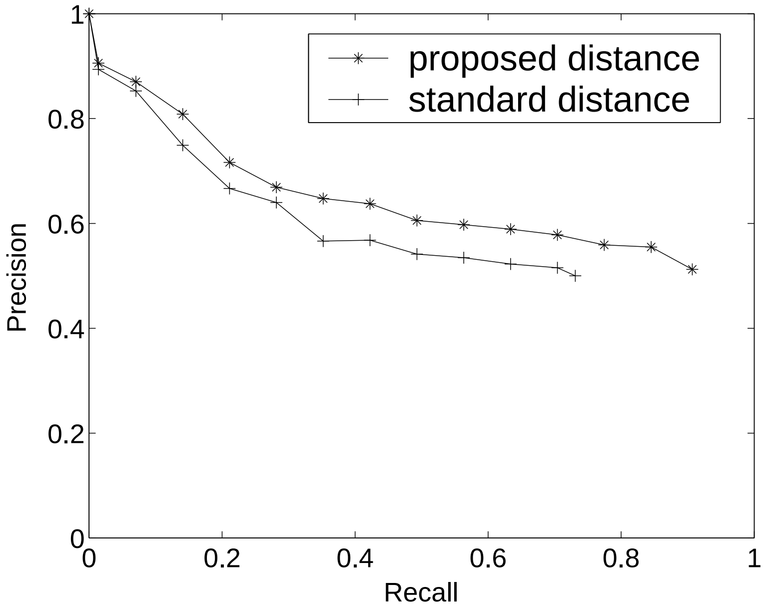


Figure 8: Average precision-recall of Columbia database: the proposed weighted distance improves the precision

precision and recall over the images in these six clusters. The performance using the standard distance measure is also presented by directly calculating $\mathbf{d}(\mathbf{q},\mathbf{v})$, the unweighted distance between a query $\mathbf{q}$ and a shape $\mathbf{v}$ in database. Figure 8 shows the average result of 20 queries on the Columbia database on the initial search without feedback. It shows that the proposed weighted distance is better than the standard distance, in both precision and recall. Figure 10 (a) and (b) show examples to compare the two distances.

In addition to weighted distance, relevance feedback is an important issue in this paper. In the experiment of prototype refinement, we tested the same query with and without refinement. Figure 6 (a) and (b) in Section 6.2 shows that our prototype refinement gives a better guide for search, as the system works more efficiently as it learns the shapes. For retrieval experiments, the system performs a new search in the database and retrieves 21

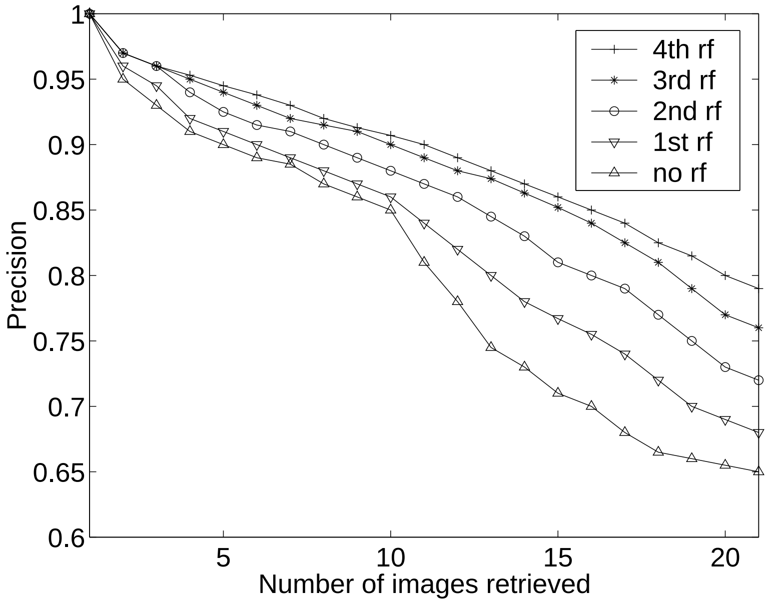
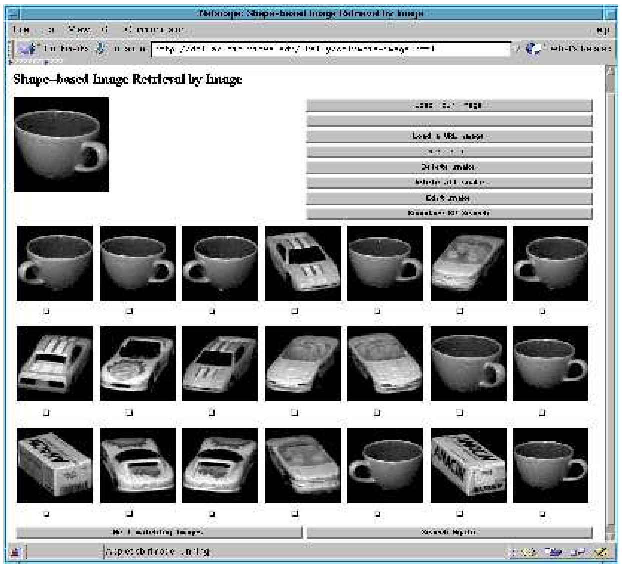


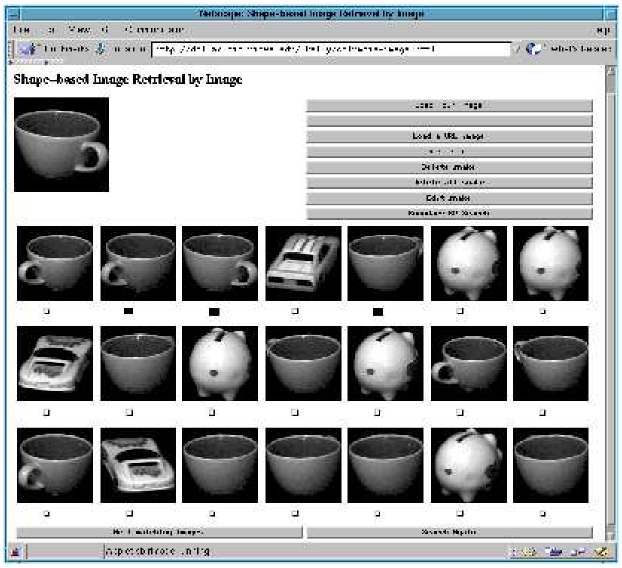
Figure 9: Average precision of Columbia database: relevance feedbacks increase the precision.

images in every incremental step. In order to take into account only the feedback effect, the system eliminates the shapes that have already been learned by the user in the first retrieval. The purpose of the experiment is to compare the retrieval performance with relevance feedback vs. the performance without relevance feedback. Figure 9 shows the average results over 20 queries of the proposed algorithm and shows that the precision continues to improve with more relevance feedback.

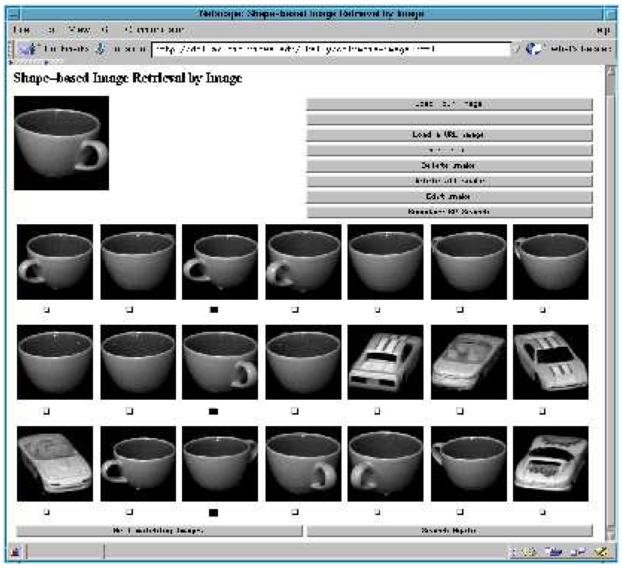
Figure 10 presents the retrieval result of a query (the upper left image) on the Columbia database. **Images are ordered from left to right, top to bottom according to the similarity measured by the distance metrices. By marking the relevant images as small black boxes in Figure 10 (b), the user selected what he wanted to retrieve.** Figures 10 (c) and (d) show the results based on the user relevance feedback. One important point of this paper is that the proposed system returns Figures 10 (d) when the user submits the same query after the second relevance learning, instead of starting again with the results



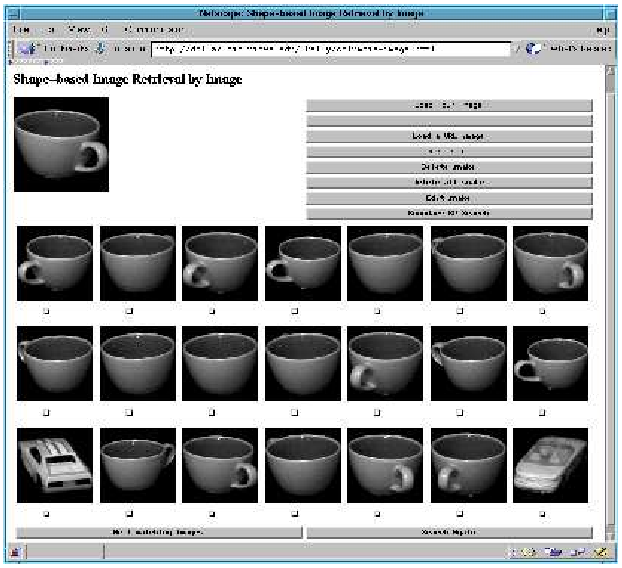
(a)



(b)



(c)



(d)

Figure 10: Retrieval results of the Columbia database: Images are ranked from left to right, top to bottom by increasing distance: (a) with the standard distance and (b) with the weighted distance for initial search, respectively, (c) with first relevance learning, and (d) with the second relevance learning

in Figure 10 (b).

7 Conclusion

This paper presents an incremental feature weight learning method based on both the clustered database and on relevance feedback for efficient shape-based image indexing and retrieval. This can be used directly in the automatic feature mining. The contribution of this paper is to learn relevance feedback and incorporate it into the feature weights. So the feature weights can include user's preference for personalization as the system is used. The experimental results show convincingly that the weighted distance using cluster information achieves better results even in the initial search. We also showed that learning feature weights based on user feedback can indeed improve retrieval performance of the proposed database system. One issue to be addressed further is to consider additional features, such as the color histogram inside of a shape, for local feature weight.

Acknowledgments

This work was partially supported by NIH grant CA64339-04 and NSF grant IIS-99-96044. A short version [9] of this paper will appear in the Second International Wokrshop on Multimedia Data Mining, Augustm 2001, San Francisco, CA.

References

[1] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithm. *Machine Learning*, 6:37–66, 1991.

[2] S. Aksoy, R. M. Haralick, F. A. Cheikh, and M. Gabbouj. A weighted distance approach to relevance feedback. In *Proceedings of the IAPR International Conference on Pattern Recognition*, pages 812–815, 2000.

[3] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C. F. Shu. The Virage image search engine: An open framework for image management. In *Proceedings of SPIE*, volume 2076, pages 76–87, 1996.

[4] M. Charikar, C. Chekur, T. feder, and R. Motwani. Incremental clustering and dynamic information retrieval. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 626–635, 1997.

[5] B. B. Chaudhuri. Dynamic clustering for time incremental data. *Pattern Recognition Letters*, 15(1):27–34, 1994.

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[7] I. Cox, M. Miller, S. Omohundro, and P. N. Yianilos. PicHunter: Bayesian relevance feedback for image retrieval. In *Proceedings of International Conference on Pattern Recognition*, volume C, pages 361–369, 1996.

[8] K.-M. Lee and W. N. Street. Automatic segmentation and classification using on-line shape learning. In *Proceedings of the 5th IEEE Workshop on the Application of Computer Vision*, pages 64–70, 2000.

[9] K.-M. Lee and W. N. Street. Automatic feature mining for personalized digital image retrieval. In *To appear in the Proceedings of the Second International Workshop on Multimedia Data Mining*, 2001.

[10] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-20). Technical Report CUCS-005-96, Columbia University, 1996. http://www.cs.columbia.edu/CAVE/research/softlib/coil-20.html.

[11] W. Niblack, R. Barber, W. Equitz, M. Flicker, E. Glasman, D. Petkovic, P.Yanker, and C. Faloutsos. The QBIC project: Query images by content using color texture and shape. In *Proceedings of SPIE Conference on Storage and Retrieval for Image and Video Databases V*, volume 1908, pages 173–187, 1993.

[12] J. Peng, B. Bhaunu, and S. Qing. Probabilistic feature relevance learning for content-based image retrieval. *Computer Vision and Image Understanding*, 75(1/2):150–164, 1999.

[13] Y. Rui, T. S. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in MARS. In *Proceedings of the IEEE International Conference on Image Processing*, volume II, pages 815–818, 1997.

[14] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrota. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Transanctions on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.

[15] K.-L. Tan, B. C. Ooi, and L. F. Thiang. Indexing shapes in image databases using the centroid-radii model. *Data and Knowledge Engineering*, 32(3):271–289, 2000.