Automatic Feature Mining for Personalized Digital Image Retrieval

Kyoung-Mi Lee
Department of Computer Science
The University of Iowa
Iowa City, IA 52242, USA
email: klee1@cs.uiowa.edu

W. Nick Street
Department of Management Sciences
The University of Iowa
Iowa City, IA 52242, USA
email: nick-street@uiowa.edu

ABSTRACT
Computing the distance between two features often measures similarity between objects. 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 an incremental method to automatically obtain feature weights based on both the clustered database and on relevance feedback. We show that using cluster information for an initial search gives better results than using the standard distance. In contrast to existing image database systems, the system can store user feedback into feature weights and reduce redundant search steps. We present shape-based indexing and retrieval results that demonstrate the efficacy of our technique.

KEY WORDS
Image retrieval, Incremental clustering, Weighted feature distance, Prototype refinement, Shape-based query

1. Introduction

In general, queries based on content similarity to an example (object or images) in terms of features (such as color, texture, shape, etc) are known as Query-by-Example. These features are extracted from each example to represent its content and then the image 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 differential weight 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.

Extracted features cannot always capture similarity in human perception. To give the user more control over the search criteria, some systems allow the user to weight the features [6, 2]. This manual adjustment is not generally applicable because it is time consuming and the user does not usually have a detailed understanding of the features.

Recently, relevance feedback has been the most commonly applied method to calculate feature weights using 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. 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 [8]. Peng *et al.* proposed probabilistic feature relevance learning to update weights iteratively, also based on relevance feedback [7]. 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 [1].

In this paper, we propose an incremental learning to automatically obtain feature weights based on both the clustered database and on relevance feedback. Feature weights with cluster information give better initial search than flat feature weights. This feature weights is updated and adapted using relevance feedback by clustering relevant examples into the corresponding cluster. Because users specify their particular preferences by providing feedback, the adapted feature weights can provide personalized retrieval. Section 2 presents the proposed method: incremental clustering, weighted feature distance, and prototype refinement. The indexing and retrieval performance of the shape-based query system are presented in Section 3. Finally, a conclusion will be presented in Section 4.

2. Method

In this section, we propose a novel method that estimates each feature's weight in the distance metric. The target of this paper is to get personalization with feature weights by incorporating relevant feedback.

2.1 Incremental Clustering

A major issue that databases are now facing is the extremely high rate of update. Typically, clustering is used to accelerate query processing by considering only a small number of representatives of the clusters, rather than the entire database. However, existing clustering algorithms are not suitable for maintaining clusters in ever-increasing volume of data, and they have been struggling with the problem of updating clusters without frequently performing complete reclustering [3].

A given set of N data, $\mathbf{V} = \{\mathbf{v}^1,\mathbf{v}^2,\cdots,\mathbf{v}^N\}$ will be partitioned into C clusters $\{\mathbf{P}^1,\mathbf{P}^2,\cdots,\mathbf{P}^C\}$ such that data 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 $(\mathbf{P}^c)^{c=1\cdots C}$. The ith feature of the prototype, $\mathbf{P}^c = (P_1^c, P_2^c, \cdots, P_I^c)$ is the center of the ith features of data in the corresponding cluster and can be computed by averaging the features that belong to the cluster **c**. To measure the spread of a set of data around the center of the data in the cluster, we use the standard deviation (σ^c).

In this paper, we use an incremental learning method [4] as follows. If an input data is similar to an existing cluster or is grouped by the user explicitly, the input data is assigned to one of the existing clusters. If the input data is dissimilar to all existing clusters or is designed as a new cluster by a user, the new cluster is created with the input data.

Incrementally, whenever a new datum **u** is indexed and assigned to an existing cluster **c**, P_i^c and $\sigma_i^{c'}$ are updated as follows:

$$P_i^{c'} = \frac{N_c P_i^c + u_i}{N_c + 1} \quad \text{and } \sigma_i^{c'} = \sqrt{\frac{N_c s_i + (P_i^{c'} - u_i)^2}{N_c + 1}},$$

where N_c is the number of data in the cluster **c** before assigning **u** and is increased by 1. Also $s_i = (\sigma_i^c)^2 + (P_i^{c'}$.

2.2 Weighted Feature Distance

In this section, we consider the problem of similarity measurement used for retrieval. The most common measure of similarity between data, **v** and **u**, is the distance between them as $\mathbf{d}_p(\mathbf{v},\mathbf{u}) = \|\mathbf{v} - \mathbf{u}\|_p$ where $\|\cdot\|_p$ is, e.g., the Manhattan (p=1), Euclidean (p=2), or max (p=∞) norm. In the I dimensional space, the distance between a prototype $\mathbf{P}^c$ and a given query **q** is computed as

$$\mathbf{d}_p(\mathbf{P}^c, \mathbf{q}) = \left(\sum_{i=1}^{I} |P_i^c - q_i|^p\right)^{\frac{1}{p}} \tag{1}$$

where P_i^c and q_i are the ith features of $\mathbf{P}^c$ and **q**, respectively.

However Eq. (1) does not take into account the influence of the scale of each feature in the distance computation. The scale of a feature can change the overall contribution of that feature in the distance computation [7]. 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. The weight for a feature P_i^c can be defined as

$$w_i^c = \frac{|P_i^c - q_i|}{\sigma_i^c}.$$

That is, a feature's importance inversely proportional to the relative distance between the query and the prototype along that dimension. It is also 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 [8]. 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 σ_i^c or σ_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 weights that q_i exerts 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 the distance between a prototype and a datum. The distance between $\mathbf{P}^c$ and **q** can finally be computed as

$$\mathbf{d}_w(\mathbf{P}^c, \mathbf{q}) = \left(\sum_{i=1}^{n} |w_i^c(P_i^c - q_i)|^p\right)^{\frac{1}{p}}. \tag{3}$$

These weights enable the distance computation to elongate dissimilar features, and, at the same time, to constrict the closest ones.

In addition to $\mathbf{d}_w(\mathbf{P}^c,\mathbf{q})$, for the actual retrieval, we need to compute the distance between **q** and data **v** in $\mathbf{P}^c$ which is similar to **q** as

$$\mathbf{d}_w(\mathbf{v}, \mathbf{q}) = \left(\sum_{i=1}^{n} |w_i^c(v_i - q_i)|^p\right)^{\frac{1}{p}}. \tag{4}$$

Given **q**, selection of data sharing similar characterizing elements starts by comparing **q** to the set of prototypes using Eq. (3). If some prototypes are selected as being similar to **q**, only the data in those prototypes calculate the distance with the query using Eq. (4).

2.3 Prototype Refinement

In the information retrieval literature, it has been well established that retrieval performance can be significantly improved by incorporating the user as part of the retrieval loop. the most popular approach is query refinement which 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 need. When the user looks for the image he found previously, the system can give results using information learned by his feedback, instead of starting an initial search.

1. Initialization: m = 0,
   N_{c_0} = N_c, $\mathbf{P}^{c_0}$ = $\mathbf{P}^c$, and σ^{c_0} = σ^c, where m = 0 means there is no relevance feedback to **q**.
2. Retrieval
   (a) Calculate $w_i^{c_m}$ using Eq. (2).
   (b) Search similar prototypes using Eq. (3).
   (c) Search relevant data in similar prototypes $\mathbf{P}^{c_m}$ using Eq. (4).
3. User relevance feedback: Select relevant data R^j from retrieved results.
4. Prototype refinement: Modify each new prototype and standard deviation using Eq. (5).
5. $m = m + 1$ and go to step 2.

Algorithm 1. Feature weights are updated incrementally by user relevance feedback

In this paper, we refine prototypes by specifying over time a set of relevant results. At each step, the user labels the retrieved data as relevant or irrelevant to the query **q**. The system then clusters the relevant data to the corresponding prototype $\mathbf{P}^c$ and makes it satisfy the user's need better than the original prototype

$$\begin{aligned} N^{c'} &= N_c + N_c^R, \\ P_i^{c'} &= \frac{N_c P_i^c + \sum_{j=1}^{N_c^R} R_i^j}{N_c + N_c^R}, \quad \text{and} \\ \sigma_i^{c'} &= \sqrt{\frac{N_c s_i^c + \sum_{j=1}^{N_c^R} (P_i^c - R_i^j)^2}{N_c + N_c^R}}, \end{aligned} \tag{5}$$

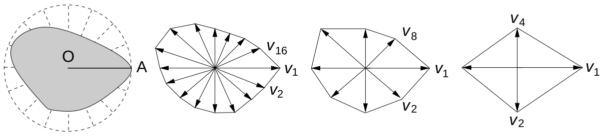
where N_c is the number of shapes and N_c^R is the number of relevant shapes and R_i^j is the ith feature of the jth relevant data, in the cluster **c**, respectively. Also, $s_i^c = (\sigma_i^c)^2 + (P_i^{c'} - P_i^c)^2$. This process is detailed in Algorithm 1.

3. Application to Shape-based Query

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 [5]. 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; it measures the ability of a system to present all relevant images. Precision is the ratio of relevant images retrieved to the total number of images retrieved; it measures the ability of a system to present only relevant images.

3.1 Shape-based Query

In this paper, we use only shape for indexing and retrieval and adopts the centroid-radii model [4, 9] to represent the shape. The shape is represented as a vector, **v** = $(v_1, v_2, \cdots, v_I)$, = $(q_{\theta_1}, q_{\theta_2}, \cdots, q_{\theta_I})$, where I is the number of shape features and v_i is the ith radius from the centroid to the boundary of the shape. With a sufficient number of radii, dissimilar shapes can be differentiated from each other. We used a 36-dimensional data space for two databases. Shapes are normalized for translation, size, orientation and reflection during the clustering.



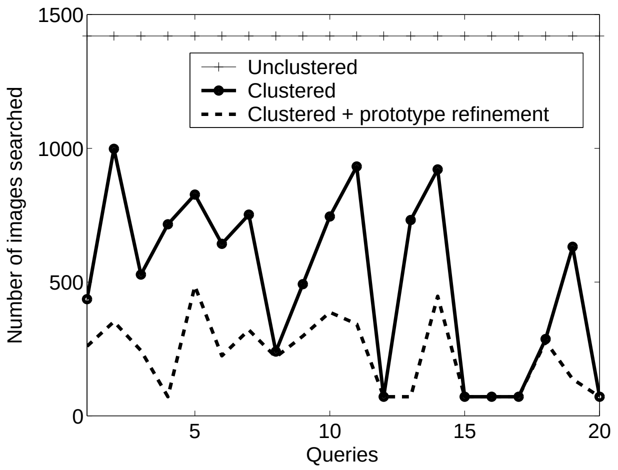
(a) Query (b) 16D (c) 8D (d) 4D

Figure 1. Shape-based query

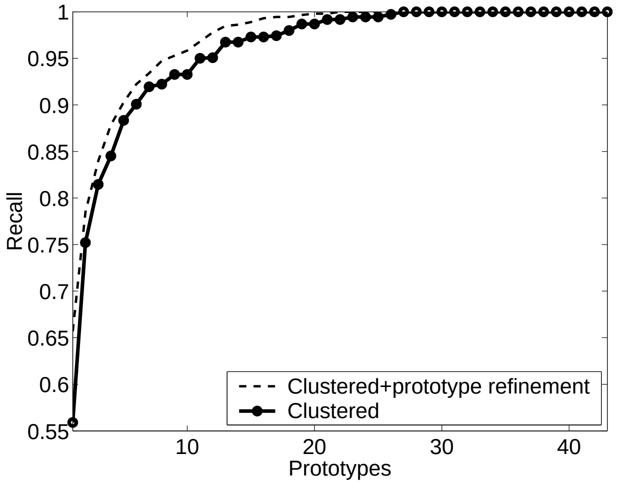
The three representations in Figure 1 ((b), (c) and (d)) show this model can facilitate multiresolution representation and multiresolution querying. For example, for a database whose shapes are represented using I features, a shape query with $\frac{I}{2}$ features can be of the form $(v_1, v_3, \cdots, v_{I-1})$. This is a less precise description of the query. To facilitate query retrieval with such query, the query can be mapped to a I-dimension vector where values of those unspecified features are filled with "*don't care*" values [9]. Clearly, more shapes match such query than when all the features are specified.

3.2 Shape-based Indexing

We used 1420 images for indexing and 20 images (45^o for each object) for query. For object-based images, locating and recognizing objects from the images is a challenging goal. There is no general solution to the difficult task of object segmentation, and successful solutions are limited to specific domains. In cases where good segmentation is possible, the object shape is a characteristic which can contribute enormously in further analysis. In this paper, we use an adaptive thresholding algorithm to extract object boundaries. The object database then was incrementally clustered and created 45 prototypes. These prototypes are used as an indexed key to reduce search time. To evaluate indexing effectiveness, we tested the retrieval performance after clustering.



(a)



(b)

Figure 2. Indexing Effectiveness of Columbia database: (a) Number of images searched for recall of 1.0 per query and (b) Average retrieval performance vs. the number of prototypes

Figure 2 (a) shows the number of images searched for a recall of 1.0 in the 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. Figure 2 (a) shows that the clustering method reduces search range. Figure 2 (b) shows how the number of prototypes affects the average performance for 20 queries performed on the database of the 1420 images. With only the one most relevant prototype, images from the same object were judged as most similar 56.7% of the time in average. As the number of prototypes reached six, performance reaches approximately 90%.

3.3 Shape-based Retrieval

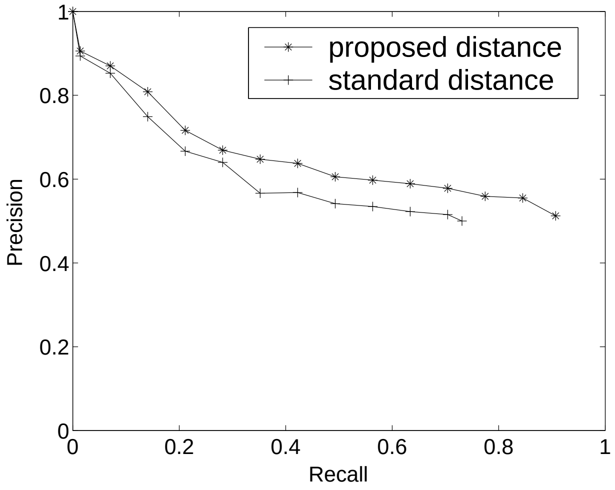


Figure 3. Average precision-recall of Columbia database

The proposed system first finds six most similar prototypes and then ranks shapes in these six clusters based on weighted distance from the queries. We measure the precision and recall over a number of images in six clusters. The performance using the standard distance measure is also presented by directly calculating $\mathbf{d}_p(\mathbf{q},\mathbf{v})$, the unweighted distance between a query $\mathbf{q}$ and a shape $\mathbf{v}$ in database. Figure 3 shows the average result of 20 queries on the database at the initial search without feedback. It shows that the proposed weighted distance is better than the standard distance, in both precision and recall. Figure 5 (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 2 (a) and (b) in Section 3.2 shows that our prototype refinement give a better guide for search, thus 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 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 4 shows the average results over 20 queries of the proposed relevance feedback algorithm and shows that the precision keeps getting better

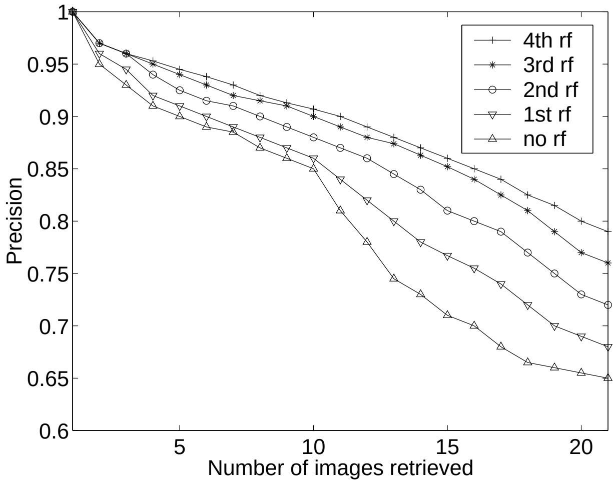


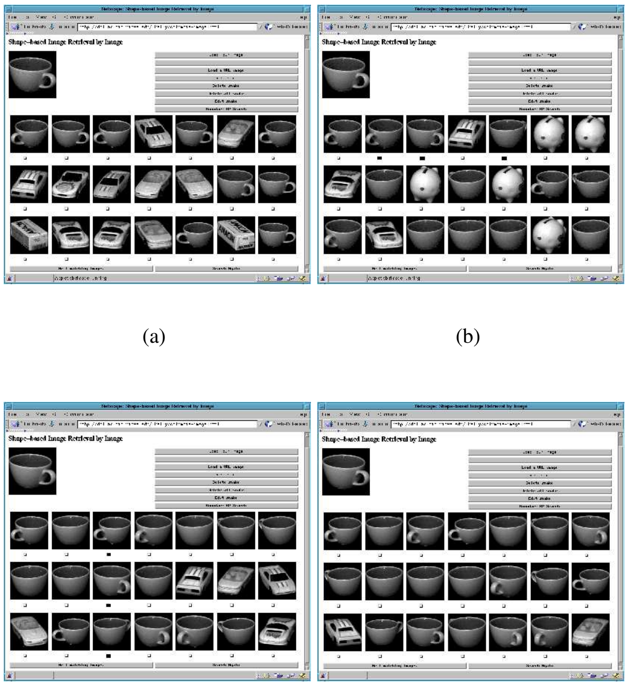
Figure 4. Average precision of Columbia database for the first two relevance feedbacks (rf)

than with no user feedback. Figure 5 (a) and (b) show an example of results with two distances.

Figure 5 presents the retrieval result of a query (the upper left image) on the database. The user selected what he wanted to retrieve (small black boxes) in Figure 5 (b). Figures 5 (c) and (d) show the results based on the user relevance feedback. One important thing of this paper is the proposed system returns Figures 5 (d) when the user wants to find the same query after the second relevance learning, not starting again with Figure 5 (b).

4. 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. The experimental results shows 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.



(a) (b) (c) (d)

Figure 5. 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

References

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] J. Peng, B. Bhaunu, and S. Qing. Probabilistic feature relevance learning for content-based image re-

trieval. *Computer Vision and Image Understanding*, 75(1/2):150–164, 1999.

[8] 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.

[9] 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.