AUTOMATIC MATCHING TOOL SELECTION USING RELEVANCE FEEDBACK IN MARS

Yong Rui, Thomas S. Huang, Sharad Mehrotra and Michael Ortega

Beckman Institute and Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
E-mail:{yrui, huang}@ifp.uiuc.edu and {sharad, ortega-b}@cs.uiuc.edu

ABSTRACT

For a given visual feature, due to the diversity of human's subjective judgment, a visual information retrieval system that supports a single prefixed similarity measure will result in poor retrieval performance. To address this problem, this paper proposes the concept of *similarity matching toolkit* which consists of different similarity measures simulating human's perceptions of the given feature from different aspects. The toolkit supports a *feedback-driven tool selection* mechanism which adapts to the similarity measure that best fits the user's perception.

To illustrate the advantage of the proposed toolkit approach, we apply it to shape-based image retrieval. The paper describes a shape matching toolkit consisting of four transformation-invariant and computationally efficient matching tools and describes how relevance feedback can be used for automatic tool selection. Experimental results validate the flexibility of the matching toolkit and show the effectiveness of the relevance feedback for shape matching tool selection.

1. INTRODUCTION

In the past five years, content-based image retrieval is becoming a very active research area [1, 2, 3]. However, in order for this approach to be of practical use, there are still many research issues need to be solved. One of such research issues is how to incorporate human expertise to improve retrieval performance, as human is already a part of the retrieval process.

For any low-level visual feature, such as color, texture, or shape, there exist dozens of similarity measures. None of them has been agreed on best simulating user's perception of the feature, since different persons, or even the same person under different circumstances, may have different perception criteria. Therefore, a robust Visual Information Retrieval (VIR) system must be capable of supporting multiple similarity measures to flexibly support different perception criteria of different users, rather than prefixing a single similarity measure at the system design stage.

The similarity measures are referred as *matching tools* in this paper and they together define a *matching toolkit* for a particular feature. While it is relatively easy for a user to specify which visual features he is interested in, it is difficult for him to specify which matching tool best fits his perception criterion. This requires the user to have enough knowledge of the properties of the matching tools, which is normally not the case. This difficulty is bypassed by most existing systems by prefixing the similarity measure at the system design stage at the cost of potentially poor retrieval performance.

In MARS [1], the technique of *relevance feedback* is proposed towards solving this difficulty. Specifically, for a given feature that the user is interested in, the best matching tool will be determined via relevance feedback. The user is not required to have any knowledge of the properties of the matching tools. He or she only needs to rank the retrieval returns according to his own perception criterion and feedbacks the ranks to the VIR system. From the user's feedback, the VIR system will *automatically* identify the matching tool that best fits this particular user's perception criterion.

While the proposed approach is valid for automatically identifying the matching tool of any visual feature, shape feature is chosen to illustrate how relevance feedback is used for automatic matching tool selection.

This work was supported in part by NSF/DARPA/NASA DLI Program under Cooperative Agreement 94-11318, in part by ARL Cooperative Agreement No. DAAL01-96-2-0003, and in part by CSE Fellowship, UIUC.

[1] MARS is the Multimedia Analysis and Retrieval System being built at University of Illinois at Urbana-Champaign.

Among the low level visual features, shape is the most challenging and has been implemented in only a few systems[3, 2]. An efficient shape feature *model* (both representation and matching tool) in a VIR system must demonstrate:

- *Invariance to transformation*: The model should be invariant to geometric transformations, such as translation, rotation, and scaling, to be a valid shape model.
- *Compact representation and fast matching speed*: The number of objects stored in a VIR system is normally very large. It is highly desirable to have a compact representation to minimize the storage overhead and have a fast matching tool to minimize the retrieval time.

A Fourier Descriptor (FD) representation and four matching tools are proposed to construct the shape matching toolkit.

This paper will focus on two main aspects, i.e. *shape matching toolkit construction* and *automatic tool selection via relevance feedback*. The rest of paper is developed as follows. FD based shape representation is discussed in Section 2. Section 3 describes four transformation-invariant and fast speed matching tools. The process of automatic tool selection via relevance feedback is discussed in Section 4. Experimental results and conclusions are given in Sections 5 and 6 respectively.

2. FD SHAPE REPRESENTATION

Shape representation specifies how the outer boundary of a shape is represented by a set of parameters. We choose the Fourier Descriptor (FD) [4, 5, 6] as our shape representation, since it meets both the requirements discussed in Section 1.

A point moving along the shape boundary generates a complex sequence

$$z(n) = x(n) + jy(n), \quad n = 0, ..., N_B - 1 \tag{1}$$

where $x(n)$ and $y(n)$ are the x and y coordinates of the nth boundary points, and N_B the number of boundary points of the shape. The FD shape representation is defined as the Discrete Fourier Transform (DFT) of $z(n)$.

$$Z(k) = \sum_{n=0}^{N_B-1} z(n) e^{-j\frac{2\pi nk}{N_B}} = M(k)e^{j\theta(k)} \tag{2}$$

where $k = 0, ..., N_B - 1$; $M(k)$ is the magnitude and $\theta(k)$ the phase angle.

Mathematically, $z(n)$'s are floating-point values and can be sampled dense enough to form a continuous boundary (see the left two triangles in Figure 1). We can derive nice transformation-invariant similarity measures based on the mathematical boundary[4, 5]. In practice, however, the continuous boundary is discretized in the image domain and the discretization noise causes the *staircase effect* (see the right two triangles in Figure 1). In Figure 1, although the right two triangles are obtained from the same mathematical triangle (with a 45^o rotation), the FD representations of the upper and lower discritized triangles differ considerably[6]. The transformation-invariant similarity measures based on mathematical boundaries will no longer be invariant to the discretized boundaries.

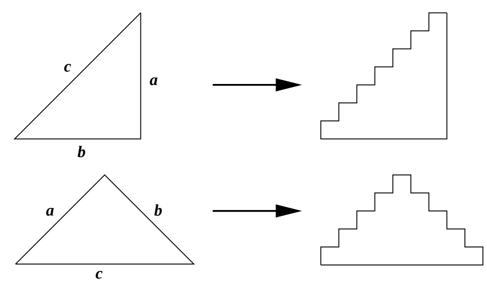


Figure 1: Different discretization of the same triangle

To overcome this difficulty, a much more robust FD representation was developed in our previous research [6]:

1. Compute the DFT of the shape boundary $z(n)$, $Z(k)$, using Equation 2;
2. Use the low frequency $[-N_C, +N_C]$ coefficients, where N_C represents the number of the FD coefficients, to reconstruct dense but possibly non-uniform samples $z_{dense}(n)$ of the original boundary:

$$\begin{aligned} z_{dense}(n) &= \sum_{k=-N_C}^{N_C} Z(k)e^{-j\frac{2\pi nk}{N_B}}, \\ n &= 0, ..., N_{dense} - 1 \end{aligned} \tag{3}$$

   where N_{dense} is the number of dense samples.
3. Use interpolation to trace the dense samples $z_{dense}(n)$ and construct uniform samples $z_{unif}(n)$, n = 0, ..., N_{unif}, where N_{unif} is the number of uniform samples. The uniform samples $z_{unif}(n)$ are *uniformly* spaced on the boundary in terms of arc length;
4. Compute the length of the boundary l_b by summing over all the arc lengths.
5. Normalize the samples $z_{unif}(n)$ to unit-length samples $z'_{unif}(n)$

$$z'_{unif}(n) = z_{unif}(n)/l_b \tag{4}$$

6. Compute the DFT of $z'_{unif}(n)$ to obtain coefficients $Z_{unif}(k), k = -N_C, ..., N_C$.

$Z_{unif}(k)$'s are the final representation of a shape and is stored in the database. Step 2 cuts off the high frequency components, which reduces the noise corruption. Step 3 forms uniform samples, which minimizes the staircase effect. Besides smoothing out the staircase effect, the above procedure (Steps 4 and 5) ensures all the shape boundaries are of the same scale (length); thus making the representation invariant to scaling.

Besides the FD coefficients, the major axis orientation ϕ is also calculated and stored in the database, which will be used in constructing rotation-invariant matching tools. The orientation of the major axis ϕ is defined as:

$$\phi = \frac{1}{2}\tan^{-1}\left(\frac{2cm_{11}}{cm_{20} - cm_{02}}\right) \tag{5}$$

where cm_{ij} is the $(i,j)^{\text{th}}$ central moment of the shape.

To summarize, the FD shape representation discussed above has the following properties:

- *Compactness in representation*: Instead of storing the whole boundary sequence $z(n)$, only the low frequency FD coefficients and major axis orientation ϕ are stored in the database.
- *Invariance to scaling*: Steps 4 and 5 normalize the shape boundary to a unit-length boundary, which ensures the representation is invariant to scaling. For the matching tools discussed in the next section, only the invariance to translation and rotation needs to be considered.

3. SHAPE SIMILARITY MATCHING TOOLKIT

The FD shape representation described in the previous section has achieved part of the two requirements discussed in Section 1, i.e. *compactness of representation* and *invariance to scaling*. The rest of the two requirements, i.e. *invariance to translation and rotation* and *fast matching speed* will be achieved by the matching tools defined over the FD representation.

In the reminder of the section, we will describe four matching tools that have been implemented in the shape matching toolkit, i.e. Euclidean, Modified Fourier Descriptor (MFD), Chamfer, and Hausdorff. The first two tools are frequency domain tools and the last two are spatial domain tools.

3.1. Euclidean Matching Tool

Based on the data stored in the database, a natural way to compute the similarity between two boundaries $z_1(n)$ and $z_2(n)$ is to compute the (weighted) Euclidean distance in the FD coefficient space:

1. Compute the major axes difference between the two shapes,
$$\psi = \phi_2 - \phi_1$$
2. Rotate $z_2(n)$ such that its major axis aligns with $z_1(n)$'s major axis. This can be achieved easily in the FD coefficient space by rotating the phase angles of $Z_2(k)$ by ψ:
$$Z'_2(k) = M_2(k)e^{j(\theta_2(k)-\psi)}$$
3. Compute the Euclidean distance in the FD coefficient space:
$$Dist_{Euclidean} = \sqrt{\sum_{k=-N_c, k\neq 0}^{N_c} w_k (Z_1(k) - Z'_2(k))^2} \tag{6}$$
where w_k is the weight for the kth FD coefficient, which is normally inverse proportional to the frequency index to emphasize the low frequency components.

Steps 1 and 2 above ensure that $Dist_{Euclidean}$ is invariant to rotation. The condition $k \neq 0$ in Step 3 ensures $Dist_{Euclidean}$ is invariant to translation.

3.2. MFD Matching Tool

Based on the same FD shape representation, in the same frequency domain, the MFD matching tool perceives the similarity between shapes in a *different* way[6].

Let $z_2(n)$ be a boundary sequence obtained from $z_1(n)$: $z_2(n)$ is $z_1(n)$ translated by z_t, rotated by ψ, and scaled by α. Explicitly, $z_2(n)$ is related to $z_1(n)$ by

$$z_2(n) = \alpha z_1(n) e^{j\psi} \tag{7}$$

The corresponding DFT of $z_2(n)$ is

$$\begin{aligned} Z_2(k) &= \sum_{n=0}^{N_B-1} z_2(n) e^{-j\frac{2\pi nk}{N_B}} && (8)\\ &= \alpha e^{j\psi} \sum_{n=0}^{N_B-1} z_1(n) e^{-j\frac{2\pi nk}{N_B}} && (9)\\ &= M_2(k) e^{j\theta_2(k)} && (10)\end{aligned}$$

where

$$\begin{aligned} M_2(k) &= \alpha M_1(k), && (11)\\ \theta_2(k) &= \theta_1(k) + \psi && (12)\end{aligned}$$

The magnitude and phase angle of FD coefficients of $z_2(n)$ are related to those of $z_1(n)$ in the way specified in Equations 11 and 12. Based on these relations, we construct two sequences

$$\begin{aligned} ratio(k) &= \frac{M_2(k)}{M_1(k)} \quad (13) \\ shift(k) &= \theta_2(k) - \theta_1(k) - \psi \quad (14) \\ k &= -N_C, ..., N_C, \quad k \neq 0 \end{aligned}$$

It is easy to see that if $z_2(n)$ is indeed a transformed version of $z_1(n)$, then the above two sequences would be two constant sequences. Specifically, *ratio* sequence will consist of all α's and *shift* sequence will consist of all 0's. On the other hand, if $z_2(n)$ is very different from $z_1(n)$, the two sequences will have high variances. Based on this intuition, the standard deviation is a good measure of the similarity. The similarities for magnitude (D_m) and phase angle (D_p) are defined as

$$\begin{aligned} D_m &= \sigma[ratio] \\ D_p &= \sigma[shift] \quad (15) \end{aligned}$$

where σ denotes standard deviation.

The overall similarity distance is defined as the weighted sum of D_m and D_p:

$$Dist_{MFD} = w_m D_m + w_p D_p \quad (16)$$

where w_m and w_p are weighting constants. Empirically, we find that $w_m = 0.9$ and $w_p = 0.1$ gives good results to most of the images.

The condition $k \neq 0$ in Equations 13 and 14 ensures the matching tool is invariant to translation. Furthermore, Equation 14 takes the major axis orientation into account and makes the matching tool invariant to rotation.

3.3. Chamfer Matching Tool

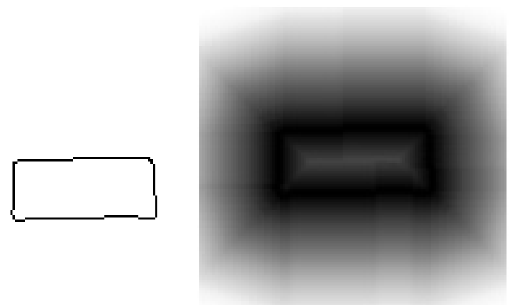
Chamfer matching tool is a spatial domain similarity measure. The original Chamfer algorithm is not invariant to transformations; thus requires intensive computation[7]. A transformation-invariant Chamfer algorithm is proposed in this paper based on the FD representation, which will be discussed in Section 3.3.2.

3.3.1. The original Chamfer algorithm

For the two images that are to be matched, one is called *pre-distance image* and the other called *pre-polygon image*. A *distance image* and a *polygon image* are then constructed from the corresponding pre-distance and pre-polygon images, before the matching is performed. For most applications, the choice for the pre-distance or pre-polygon image is arbitrary. However, the complexity for constructing distance image is much higher than that for polygon image. Therefore, if the matching speed is a major consideration, the to-be-matched image should be chosen as the pre-distance image, and the matching images be chosen as the pre-polygon images. In our image database application, it is obvious that the query image should be chosen as the pre-distance image.

In the pre-distance image, each non-boundary pixel is given a value that is a measure of the distance to the nearest boundary pixel. The boundary pixels get the value zero. To compensate different distance values of horizontal (vertical) neighbors and diagonal neighbors, 3 is used as the distance for the former and 4 the latter.

Following the algorithm described in [7], the distance image is constructed from the pre-distance image, as shown in Figure 2. In the distance image, the darker the pixel's intensity is, the closer it is to the boundary.



(a) (b)

Figure 2: (a) The pre-distance image; (b) The distance image.

The edge image corresponding to the pre-polygon image is called the *polygon image*. In our case, the original pre-polygon image is already a edge (boundary) image. The polygon image is just the pre-polygon image itself. When we match the two boundaries, the polygon image is superimposed on the distance image. An average of the distance image pixel values that are hit by the boundary pixels in the polygon image is the Chamfer distance:

$$Dist_{Chamfer} = \frac{1}{3}\sqrt{\frac{1}{N_B}\sum_{n=1}^{N_B} V_n^2} \quad (17)$$

where V_n is the distance value hit by boundary pixel $z_2(n)$, and N_B the number of boundary pixels in the polygon image.

If the pre-distance image and the pre-polygon image are arbitrary images containing boundaries of any scale and orientation, multiple rounds of matching need to be performed. To find the real similarity value between two boundaries, the polygon image has to be moved over the distance image at different scales and orientations.

3.3.2. *A fast and transformation-invariant Chamfer algorithm*

The Chamfer algorithm described in Section 3.1.1 is not invariant to transformation. Although a hierarchical matching algorithm (HMA) was proposed[7], the matching speed is still far from tolerable in image database application.

Based on the FD representation and the availability of major axis orientation ϕ, a much better approach is to normalize the boundaries before the Chamfer algorithm is applied. The normalizing and matching procedure is summarized as:

1. Reconstruct the query image's shape boundary $z_1(n)$ from the FD coefficients stored in the database.
$$z_1(n) = IDFT(M_1(k)e^{j(\theta_1(k))}), \quad k \neq 0 \quad (18)$$
where $IDFT$ denotes the inverse DFT.
2. Reconstruct a rotated version of shape boundary, $z_2'(n)$, for each of other images, by using both the FD coefficients and the major axis orientation ϕ.
$$z_2'(n) = IDFT(M_2(k)e^{j(\theta_2(k)-\psi)}), \quad k \neq 0 \quad (19)$$
where $\psi = \phi_1 - \phi_2$.
3. Construct the distance image from $z_1(n)$ (see Figure 2). The polygon images are the same as the pre-polygon images, i.e. $z_2'(n)$'s.
4. Superimpose the polygon images on the distance image and compute the distance by using Equation 17.

The condition $k \neq 0$ in steps 1 and 2 ensures the centroids of the boundaries in both distance and polygon images are at the origin; thus is invariant to translation. In step 2, the polygon image's major axis is aligned with the distance image's major axis; thus the similarity measure is invariant to rotation.

Before the matching is applied, the distance and polygon images are normalized. When they are superimposed, their centroids coincide, and orientations are aligned. Only one round of Equation 17 is needed. No moving-around is necessary, and the matching is done in one scaling and one orientation. This proposed Chamfer matching algorithm is much faster than the original algorithm.

3.4. Hausdorff Matching Tool

Hausdorff matching tool is a spatial domain measure and finds many applications in Fractals[8]. Define A and B are the two boundaries to be matched. A consists of boundary pixels $z_1(n)$'s and B consists of boundary pixels $z_2(n)$'s. For a pixel on A, i.e. $z_1(n)$, the distance from $z_1(n)$ to B is defined as

$$d(z_1(n), B) = min(d(z_1(n), z_2(n)) : z_2(n) \in B) \quad (20)$$

The distance from boundary A to boundary B is defined as

$$d(A, B) = Max(d(z_1(n), B) : z_1(n) \in A). \quad (21)$$

Note that this distance metric is asymmetric. To make it symmetric, the final Hausdorff distance between boundaries A and B is defined as

$$Dist_{Hausdorf} = Max(d(A, B), d(B, A)) \quad (22)$$

The algorithm for computing the Chamfer distance can be easily adapted to compute the Hausdorff distance, except that we now need to compute two distances $d(A, B)$ and $d(B, A)$. When we compute $d(A, B)$ we use A as the distance image and B as the polygon image. When we compute $d(B, A)$, we switch the role of A and B. The distance from A to B is defined as

$$d(A, B) = \frac{1}{3}\sqrt{\max_n V_n^2} \quad (23)$$

Comparing the two spatial matching tools, i.e. Chamfer and Hausdorff, Chamfer is a norm-2 distance, which gives a balanced consideration among all the boundary pixels. Hausdorff is a norm-∞ distance, which penalizes the similarity more than Chamfer does, if only a few boundary pixels do not match well. This is illustrated in Figure 3. Chamfer will give the two boundaries high similarity while Hausdorff will penalize the upper-left *bump* on the second boundary by giving a relatively low similarity.

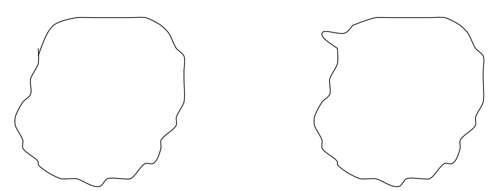


Figure 3: Norm-2 vs Norm-∞

4. AUTOMATIC TOOL SELECTION VIA RELEVANCE FEEDBACK

As described in the previous section, there are many matching tools for shape comparison; each of which try to simulate human's perception from a particular aspect. For example, we can make the following observations about the four matching tools described in Section 3:

- Euclidean and MFD simulate the human's perception from frequency domain, while Chamfer and Hausdorff simulate the human's perception from spatial domain.
- In the frequency domain, low frequency components give a rough general description of the boundary, while high frequency components give a detailed, but possibly noisy, description of the boundary. Euclidean is a norm-2 distance, which gives a balanced consideration among different frequency components. MFD is a standard deviation based distance, which penalizes the similarity more than Euclidean does, if only a single component does not match.
- In the spatial domain, Chamfer is a norm-2 distance, which gives a balanced consideration among all the boundary pixels. Hausdorff is a norm-∞ distance, which penalizes the similarity more than Chamfer does, if only a few boundary pixels do not match well.

While the shape matching toolkit supports different tools which simulate human's perception from different aspects, a user needs to specify which tool best matches his perception before the retrieval can proceed. The technique of *relevance feedback* is proposed such that the user is exempt from specifying the matching tool. That is, the user is not required to have any knowledge of the properties of the matching tools. He or she only needs to rank the retrieval returns according to his own perception criterion and feedbacks the ranks to the VIR system. From the user's feedback, the VIR system will *automatically* identify the matching tool that best fits this particular user's perception criterion.

In the TIR literature it has been well established that retrieval performance can be significantly improved by incorporating the user as part of the retrieval loop[9]. *Relevance feedback* is the mechanism supported by the TIR systems to enable users to guide the computer's search for relevant documents. In TIR domain, this technique has been extensively studied and used in the vector-based retrieval model to adjust the term (keyword) weights to improve the retrieval performance[9].

Our previous work has generalized this technique of automatic query weights adjustment to content-based image retrieval[10]. In this section we describe how the relevance feedback can also be used for automatic tool selection. Since this relevance feedback procedure is valid for any visual feature, we will describe it in a general setting. The application of it in shape feature will be discussed in Section 5.

To simplify the notations, define P to be the matching toolkit consisting of T matching tools, $p_1, ..., p_t, ..., p_T$.

For a given visual feature, a set of useful p_t's are identified and represented in P. The procedure of automatic p_t selection is summarized as follows:

1. The user specifies how many retrieval returns he wants to have. Let this number be N_r.
2. For an arbitrary given query, for each image I_n in the collection, $n = 1, ..., N_c$, where N_c is the number of images in the collection, compute the similarity distance $dist_{I_n,t}$ for each p_t in P.
3. For each p_t, based on $dist_{I_n,t}$'s, sort the image id's and construct a length-αN_r rank list l_t:
$$l_t = [I_{1,t}, ..., I_{m,t}, ..., I_{\alpha N_r,t}] \tag{24}$$
where α is a small positive integer greater than one, and $I_{m,t}$ is the image id for the mth most similar image to the query image when p_t is used. The reason we maintain a length-αN_r, not a length-N_r, rank list, is that these rank list l_t's are intermediate entities, a longer rank list will ensure better final precision. Experimentally we find that $\alpha = 2$ gives good final precision and has fast enough computation speed. Therefore, in the remaining of the procedure $\alpha = 2$ is used.
4. Define a rank-of operator $RANK_t(I_n)$, which finds the rank of image I_n, when p_t is used:
$$\begin{aligned} RANK_t(I_n) &= rank\ of\ I_n\ in\ l_t, && (25)\\ & \quad if I_n \in l_t && (26)\\ RANK_t(I_n) &= 2N_r + 1, && (27)\\ & \quad if I_n \notin l_t && (28) \end{aligned}$$
In Equations (8)-(11), for simplicity, we assign the same rank $2N_r + 1$ to all the images who are not in l_t.
5. For each image, compute the overall rank $rankAll_{I_n}$. Since only N_r images, where N_r is normally a small number, need to be returned to the user, there is no need to compute the overall rank for all the images in the database. To achieve fast retrieval speed, only the $rankAll_{I_n}$'s of the images appearing in some l_t's are computed. This approach results in a significant improvement in retrieval speed, while causing almost no retrieval miss.
$$rankAll_{I_n} = \sum_{t=1}^{T} RANK_t(I_n) \tag{29}$$
where T is the number of elements in P, and I_n appears in at least one of l_t's.
6. Based on $rankAll_{I_n}$'s, construct a length-N_r combined rank list l_c, which contains the overall most similar N_r images to the query image:
$$l_c = [I_{1,c}, ..., I_{m,c}, ..., I_{N_r,c}] \tag{30}$$

and send the retrieved image $I_{m,c}$'s to the user in the order specified in l_c;

7. The ranks for the retrieved images in l_c might not be the same as the user's perception and the user sends back a modified feedback rank list l_f:

$$l_f = [I_{1,f}, ..., I_{m,f}, ..., I_{N_r,f}] \tag{31}$$

8. For each l_t, compute the rank difference rd_t

$$rd_t = \sum_{m=1}^{N_r} abs(RANK_f(I_{m,f}) - RANK_t(I_{m,f})) \tag{32}$$

where abs denotes taking absolute value.

9. Return to the user the best p_{t^*}:

$$t^* = arg\ min(rd_t) \tag{33}$$

where arg denotes the index-selecting operator.

Usually this feedback procedure needs to be done only once and the subsequent retrieval is based on p_{t^*} just identified. Here, we assume a user's perception criterion stays relatively stable during the query process, which is normally a short period. If a user does find his perception is changing, a new round of feedback can be performed.

An alternative to the above standard procedure is to use multiple p_t's with different weights. Instead of selecting the best p_t with the minimum rank difference, we can use the inverse rank difference as the weight for each p_t. By incorporating multiple p_t's, although the retrieval speed is not as good as the above procedure, the retrieval precision is normally higher.

In both the standard and alternative relevance feedback procedure, the user is not required to have any knowledge of the characteristics of the perception criteria p_t's. He or she only needs to rank the retrieval returns according to his own judgment, and feedback the ranks to the VIR system. The good perception criteria p_t's will be automatically determined by the system based on the user's feedback.

5. EXPERIMENTAL RESULTS

To address the challenging issues involved in VIR, a Multimedia Analysis and Retrieval System (MARS) project was started at University of Illinois[3, 6, 10, 11, 12]. MARS-1 is accessible via internet at http://jadzia.ifp.uiuc.edu:8000. The relevance feedback procedure discussed in Section 4 has been implemented in a shape-based image retrieval subsystem in MARS-2. The subsystem is accessible via internet at http://quark.ifp.uiuc.edu:8080.

As part of the DLI content-based retrieval test bed, there are about 300 images in the database, which are a collection of ancient African artifacts from the Getty Museum. For the experiments, users from various domains, including users from Computer Vision, Art, Computer Science, and non-technical users, are asked to submit queries and feedback their ranks to the VIR system. Extensive experiments were performed and we have the following observations:

1. Different users, or even the same user under different circumstances, have different judgment for the similarities, which justifies the need of the matching toolkit and relevance feedback process.

2. All of the 4 matching tools have been selected as the best tools for some users, according to user's feedback.

3. If a user emphases the rough general aspect of the shape boundary, Chamfer and Euclidean are more likely to be chosen as his best tool. If a user emphases the detailed aspect of the shape boundary, Hausdorff and MFD are often chosen as his best tool. This fact matches well with the mathematical definitions of the 4 matching tools.

An example feedback process is illustrated in Figures 4 and 5. In Figure 4, the upper-left image is the query image. After the query is submitted, the combined rank list l_c is constructed, as described in Section 4. Retrieved images are then returned to the user in the order specified in l_c. The numbers in the input areas in Figure 4 are the combined ranks for the corresponding images.

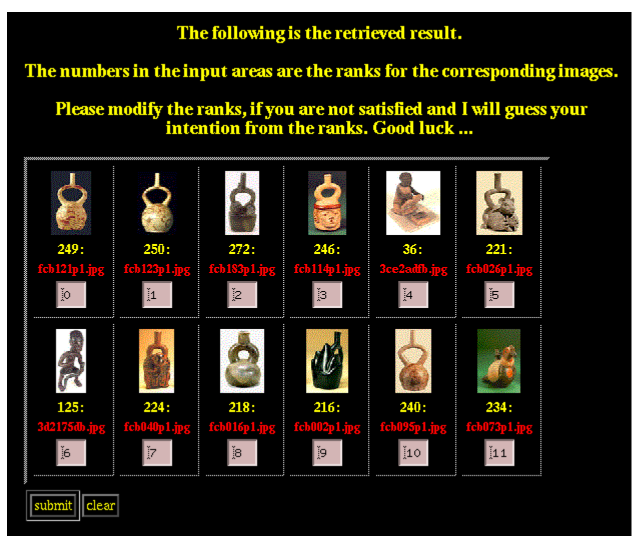


Figure 4: Relevance feedback process (a)

If the user is not satisfied with the rank order, he can modify the rank order according to his own judgment. For example, the user does not like images 36, 125, 216, and 234, which are ranked by the VIR system as 4, 6, 9, and 11 respectively (see Figure 4). The user modifies their ranks to, for example, 12, 14, 15, and 16 respectively and feedbacks the modified ranks to the system. Based on the user's feedback rank list l_f, the system determines that the best matching tool for this user is Chamfer.

Using Chamfer as the matching tool, the new retrieval results are in Figure 5. As expected, the images that the user does not like are no long in the figure; and Chamfer matches the user's perception criterion.

Figure 5: Relevance feedback process (c)

Via the relevance feedback process, the VIR system is capable of flexibly supporting different judgment criteria of different users and thus better meet the user's information need.

6. CONCLUSIONS

For a given visual feature, due to the diversity of human's subjective judgment, a visual information retrieval system that supports a single prefixed similarity measure will result in poor retrieval performance. To address this problem, this paper proposed the concept of *similarity matching toolkit* which consists of different similarity measures simulating human's perceptions of the given feature from different aspects, and the concept of *feedback-driven tool selection* where the matching tool selection is done automatically by the system with user's feedback. The main contributions of this paper are:

- The concept of *similarity matching toolkit* to flexibly support different perception criteria of different users.
- Development of a shape matching toolkit that consists of four transformation-invariant and computationally efficient matching tools.
- Development of the feedback-driven tool selection mechanism that adapts to the similarity measure that best fits the user's perception of a given feature.

Experimental results validated the flexibility of the matching toolkit and showed the effectiveness of relevance feedback.

7. REFERENCES

[1] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafine, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The qbic system," *IEEE Computer*, 1995.

[2] A. Pentland, R. Picard, and S.Sclaroff, "Photobook: Content-based manipulation of image databases," *International Journal of Computer Vision*, 1996.

[3] T. S. Huang, S. Mehrotra, and K. Ramchandran, "Multimedia analysis and retrieval system (MARS) project," in *Proc of 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval*, 1996.

[4] C. T. Zahn and R. Z. Roskies, "Fourier descriptors for plane closed curves," *IEEE Trans. on Computers*, 1972.

[5] E. Persoon and K. S. Fu, "Shape discrimination using fourier descriptors," *IEEE Trans. Sys. Man, Cyb.*, 1977.

[6] Y. Rui, A. C. She, and T. S. Huang, "Modified fourier descriptors for shape representation – a practical approach," in *Proc of First International Workshop on Image Databases and Multi Media Search*, 1996.

[7] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Trans. Patt. Recog. and Mach. Intell.*, 1988.

[8] M. Barnsley, *Fractal Everywhere*. Academic Press, Inc., 1988.

[9] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1983.

[10] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," in *Proc. IEEE Int. Conf. on Image Proc.*, 1997.

[11] Y. Rui, T. S. Huang, and S. Mehrotra, "Human perception subjectivity and relevance feedback in multimedia information retrieval," in *(submitted to) Storage and Retrieval of Images/Video Databases VI, EI'98*, 1998.

[12] M. Ortega, Y. Rui, K. Chakrabarti, , S. Mehrotra, and T. S. Huang, "Supporting similarity queries in MARS," in *Proc. of ACM Conf. on Multimedia (to appear)*, 1997.