# A Modified Fourier Descriptor for Shape Matching in MARS

Yong Rui, Alfred C. She, and Thomas S. Huang *

Image Formation and Processing Lab, Beckman Institute
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
E-mail: yrui@ifp.uiuc.edu, a-she@uiuc.edu, huang@ifp.uiuc.edu

**Abstract.** We propose a Modified Fourier Descriptor and a new distance measure for describing and comparing closed planar curves. Our method accounts for *spatial discretization* of shapes, an issue seldom mentioned, much less addressed in the literature.

The motivating application is shape matching in the Multimedia Analysis and Retrieval System (MARS), our content-based image retrieval system. The application requires a compact and reliable representation of object boundaries in the image database, and a similarity measure that can be computed in real time. We test our shape matching method on a set of Roman characters. Results indicate that our method is a feasible solution for real time shape comparison.

## 1 Introduction

Content-based retrieval (CBR) has gained considerable attention recently [1, 2, 3, 4, 5]. The most commonly researched image features used in retrieval are color, texture, and shape. Color and texture features are explored in [1, 2, 3, 4, 5]. Although shape features have also been studied[1, 5], it is still difficult to obtain a good solution.

To address the challenging issues involved in CBR, the Multimedia Analysis and Retrieval System (MARS) project was started at the University of Illinois [2, 6, 7, 8, 9, 10]. MARS supports user queries based on global color, texture, and shape, as well as queries on the spatial layout of color and texture. The on-line demo for MARS is http://jadzia.ifp.uiuc.edu:8000. MARS uses several shape matching methods, including Modified Fourier Descriptors (MFD), the proposed method presented in this paper. (For information on shape matching methods in MARS other than MFD, see [10], which describes fast algorithms we developed for Chamfer matching and Hausdorff matching. The reference also describes a relevance feedback mechanism that helps the user find the matching method that best fits his/her individual perception of shape feature.)

* This work was supported by the NSF/DARPA/NASA Digital Library Initiative under Cooperative Agreement No. 94-11318.

In general, a CBR system is useful only if it can retrieve acceptable matches in real time. This requires the choice of a suitable set of image features, a method for correctly extracting them, and a feature distance measure that can be computed in real time.

Our focus is on shape matching. We propose that a useful shape representation should satisfy the following four conditions:

1. *Robustness to Transformation* – the representation must be invariant to translation, rotation, and scaling of shapes, as well as the starting point used in defining the boundary sequence.
2. *Robustness to Noise* – shape boundaries often contain local irregularities due to image noise. More importantly, *spatial discretization* introduces distortion along the entire boundary. The representation must be robust to these types of noise.
3. *Feature Extraction Efficiency* – feature vectors should be computed efficiently.
4. *Feature Matching Efficiency* – since matching is done on-line, the distance measure must require a *very* small computational cost.

Some simple shape features are the perimeter, area, number of holes, eccentricity, symmetry, etc. Although these features are easy to compute, they usually result in too many false positives to be useful in a CBR system, thus they are excluded from our discussion.

Advanced methods that can represent more complex shapes fall into two categories. Region-based methods are the first category. A typical representative is the Moment-Invariants Method (MIM) [11]. The disadvantage of the MIM is its high computational cost (features are computed using the *entire* region, including interior pixels), and low discriminatory power. The descriptor tends to return too many false positives.

Boundary-based methods are the second category, and include the Turning Angle Method (TAM) [12] and Fourier Descriptors (FD) [13, 14]. These methods provide a much more complete description of shape than MIM; however, they are sensitive to the starting point of the shape boundary. They can discount the effect of the starting point only by solving a non-linear optimization problem, which is not feasible in a real-time CBR system. Also, to the extent of our knowledge, little research has been done on how to deal with the problem of *spatial discretization* when using these methods. We discuss this in detail in section 4.

We propose the Modified Fourier Descriptor (MFD), which satisfies our four conditions above. The FD method is the most closely related work, so we give a brief review of it in section 2. We discuss the proposed MFD in section 3. Comparisons between MFD and existing methods are given in section 4. Experimental results and conclusions are in sections 4 and 5, respectively.

**List of symbols:**

- $N_V$: number of vertices of a polygon;
- $N_B$: number of boundary points of a shape;
- $N_C$: number of FD coefficients used in shape reconstruction;
- $V_i$: the $i$th vertex of a polygon;
- $N_{dense}$: number of *dense* samples used in resampling in the MFD method;
- $N_{unif}$: number of *uniformly* spaced samples used in MFD method;
- $\alpha, \beta, \gamma$: planar curves (shape boundaries).

## 2 Fourier Descriptors

There are two commonly known FD's, described in [13] and [14], which we denote as "FD1" and "FD2", respectively. FD1 has low efficiency in reconstructing the shape, so we discuss FD2 only.

Let $\gamma$ be a clockwise-oriented simple closed planar curve with representation $z(l) = [x(l), y(l)]$, where $l$ is the arc length along $\gamma$. A point moving along the boundary generates the complex function $u(l) = x(l) + jy(l)$. FD2 is defined as:

$$a_n = \frac{1}{L\left(\frac{2\pi n}{L}\right)} \sum_{k=1}^{N_V} (b_{k-1} - b_k) e^{-j\left(\frac{2\pi n l_k}{L}\right)} \tag{1}$$

where $L$ is the total length of $\gamma$; $l_k = \sum_{i=1}^{k} |V_i - V_{i-1}|$ (for $k > 0$ and $l_0 = 0$); and $b_k = \frac{V_{k+1} - V_k}{|V_{k+1} - V_k|}$.

The distance metric is defined as the Euclidean distance in FD coefficient space. Let $\{a_n\}$ and $\{b_n\}$ denote the FD's of two curves $\alpha$ and $\beta$, respectively, and assume only $N_C$ harmonics are used; the distance metric

$$d(\alpha, \beta) = \sqrt{\sum_{n=-N_C}^{N_C} |a_n - b_n|^2} \tag{2}$$

Now assume $\beta$ is identical to $\alpha$ except for a translation, scale, and rotation, and that the curve is defined using a different starting point. Ideally, the distance between the two shapes should be zero. Translation is easily dealt with by omitting $a_0$ and $b_0$ when taking the sum in (2).

However, to account for the effects of scale ($s$), rotation ($\phi$), and starting point ($p$), we must minimize the distance metric

$$d^*(\alpha, \beta) = \min_{s,\phi,p} \sum_{n=-M, n\neq 0}^{M} \left| a_n - s e^{j(np+\phi)} b_n \right|^2 \tag{3}$$

over the parameters $(s, \phi, p)$. This is a computationally expensive optimization problem and makes FD2 impractical for shape matching in a real-time CBR system, especially when the image database is large. Similarly, TAM has the same disadvantage.

## 3 Proposed Method – A Modified Fourier Descriptor

A point moving along the shape boundary generates a complex sequence (4-neighbor chain code):

$$z(n) = x(n) + jy(n), \quad n = 0, ..., N_B - 1 \tag{4}$$

where $x(n)$ and $y(n)$ are the $x$ and $y$ coordinates of the $nth$ boundary points. The MFD is defined as the Discrete Fourier Transform (DFT) of $z(n)$:

$$Z(k) = \sum_{n=0}^{N_B-1} z(n) e^{-j\frac{2\pi nk}{N_B}} = M(k) e^{j\theta(k)} \tag{5}$$

where $k = 0, ..., N_B - 1$.

Next, we examine the properties of MFD and propose a distance measure which is both reliable and easy to compute.

Let $z'(n)$ be a boundary sequence obtained from $z(n)$: $z'(n)$ is $z(n)$ translated by $z_t$, rotated by $\phi$, and scaled by $\alpha$, with the starting point shifted by $l$. We know that $Z(k)$, for $k \neq 0$, is invariant to translation. Next, we examine rotation, scale, and starting point.

Explicitly, $z'(n)$ is related to $z(n)$ by

$$z'(n) = \alpha z(n - l) e^{j\phi} \tag{6}$$

The corresponding MFD of $z'(n)$ is

$$\begin{aligned} Z'(k) &= \sum_{n=0}^{N_B-1} z'(n) e^{-j\frac{2\pi nk}{N_B}} \\ &= \sum_{n=0}^{N_B-1} \alpha z(n-l) e^{j\phi} e^{-j\frac{2\pi nk}{N_B}} \\ &= \alpha e^{j\phi} \sum_{n=0}^{N_B-1} z(n-l) e^{-j\frac{2\pi nk}{N_B}} \end{aligned} \tag{7}$$

Setting $m = n - l$, we get

$$\begin{aligned} Z'(k) &= \alpha e^{j\phi} \sum_{m=-l}^{N_B-l-1} z(m) e^{-j\frac{2\pi mk}{N_B}} e^{-j\frac{2\pi lk}{N_B}} \\ &= \alpha e^{-j(\phi + \frac{2\pi lk}{N})} Z(k) \\ &= M'(k) e^{j\theta'(k)} \end{aligned} \tag{8}$$

where

$$M'(k) = \alpha M(k) \tag{9}$$

$$\theta'(k) = \phi + \theta(k) - \frac{2\pi lk}{N_B} \tag{10}$$

$$\phi = \theta_0 - \theta'_0; \tag{11}$$

where $\theta_0$ and $\theta'_0$ are the orientations of the major axes of the two shapes, defined as

$$\theta_0 = \frac{1}{2}\tan^{-1}\left(\frac{2cm_{11}}{cm_{20} - cm_{02}}\right) \tag{12}$$

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

The magnitude and phase angle of FD coefficients of $z'(n)$ are related to those of $z(n)$ in the way specified in (9) and (10). Based on these relations, we construct two sequences

$$ratio(k) = \frac{M'(k)}{M(k)}; \tag{13}$$

$$shift(k) = \frac{\theta(k) - \theta'(k) + \phi}{k}; \tag{14}$$

$$k = -N_C, ..., N_C, k \neq 0. \tag{15}$$

It is easy to see that if $z'(n)$ is indeed a transformed version of $z(n)$, then the two sequences above (eqs. (13)(14)) would both be constant in $k$. Specifically, $ratio(k) = \alpha$ and $shift(k) = 0$ for all $k$. On the other hand, if $z'(n)$ is completely different from $z(n)$, then $ratio$ and $shift$ will both have high variance with respect to the frequency index $k$. Based on this observation, the standard deviation is a good measure of the difference between two shapes. The distance measure for magnitude ($D_m$) and phase angle ($D_p$) are defined as

$$\begin{aligned} D_m &= \sigma[ratio] \\ D_p &= \sigma[shift] \end{aligned} \tag{16}$$

where $\sigma$ denotes standard deviation.

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

$$Dist_{MFD} = w_m D_m + w_p D_p \tag{17}$$

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 for most of the images we tested.

Note that the proposed distance measure is invariant to translation, rotation, scale, and starting point, making it suitable for on-line matching in a CBR system.

## 4 Comparisons with the Existing Methods

We compare the MFD with FD1 and FD2 in terms of both computational complexity and practical robustness.

### 4.1 Computational Complexity

We mentioned in section 1 that a good shape representation method in the CBR system should be efficient in both feature extraction and feature matching, with

much more emphasis on the latter. This is obvious since a CBR system typically does matching on-line, and may support multiple user queries simultaneously.

Tables 1 and 2 show the computation operation counts for MFD, FD1, and FD2 in feature extraction and feature matching, respectively. A subtract is counted as an add; a divide is counted as a multiply; absolute value is counted as 2 adds; math library functions (e.g., exponential, sine, square root) are counted as 16 multiplies.

We can see that although MFD requires a little bit more computation during feature extraction, it is much faster during feature matching. This is because the MFD distance measure is *intrinsically* invariant to translation, rotation, scale, and starting point. This is a very important advantage for the MFD since feature extraction is done off-line while matching is done on-line.

Table 1. Operation counts for feature extraction

| | FD1 | FD2 | MFD |
|---|---|---|---|
| Adds | $O(N_V^2)$ | $O(N_V^2)$ | $O(N_B \log_2 N_B)$ |
| Mults | $O(N_V)$ | $O(N_V)$ | $O(N_B \log_2 N_B)$ |

Table 2. Operation counts for feature matching

| | FD1 | FD2 | MFD |
|---|---|---|---|
| Adds | $O({N_C}^3)$ | $Huge^*$ | $O(N_C)$ |
| Mults | $O({N_C}^3)$ | $Huge^*$ | $O(N_C)$ |

$Huge^*$: beyond comparison since it requires finding all zeros of a trigonometric polynomial of degree $N_c$.

### 4.2 Robustness: Practice and Theory

Regardless of the different computational costs, FD1, FD2 and MFD are all valid shape representations, at least theoretically. But to be of practical use, a representation must be tested using the following procedure:

1. Use a camera to take two images of the same physical object, but at different scales, rotations, and translations.
2. Segment the two input images to obtain two shape boundaries, with arbitrary starting point.
3. Compare the features obtained from the each image.
4. If the match is good, conclude that the method is valid.

Note that the segmentation occurs *after* the transformation. This is the actual situation when comparing shapes from two different images. If we use this testing procedure, none of the existing methods give good results, including our proposed MFD method. This is because the boundaries used in these methods are sensitive to *discretization noise*. The discretization noise in many cases changes the boundary enough such that the Fourier coefficients become significantly different. Both FD1 and FD2 suffer from this problem.

The boundary extraction method of FD1 (described in [15]) is sensitive to noise. If we rotate the input image, both the number of vertices and the lengths between vertices will change. No boundary extraction method was mentioned in FD2. However, since FD1 was cited as a main reference in FD2, it most likely used the same boundary extraction method, thus, suffering from the same problem.

Since MFD uses the 4-neighbor chain code, it also suffers from discretization noise. A simple example illustrates this point (see Figure 1). We discretize the triangle using two different orientations. Note that the upper figure has *staircase effect* in edge $c$ while the lower figure has *staircase effect* in edges $a$ and $b$. The Fourier transform magnitudes, as well as $ratio(k)$ (defined in section 3) are shown in Figure 2. Note that the plot of $ratio(k)$ shows a large variance, even though the DFT coefficients were obtained from the *same* object.

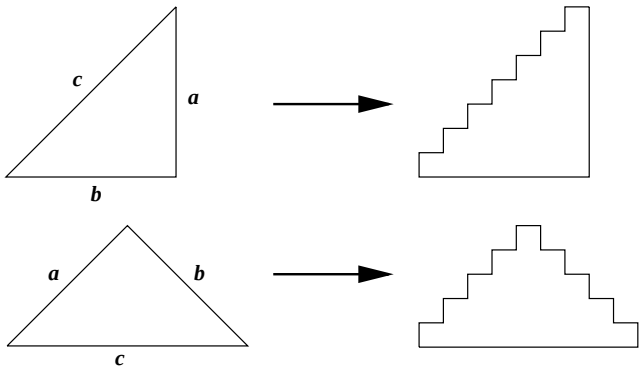



**Fig. 1.** Effect of spatial discretization on the chain code.

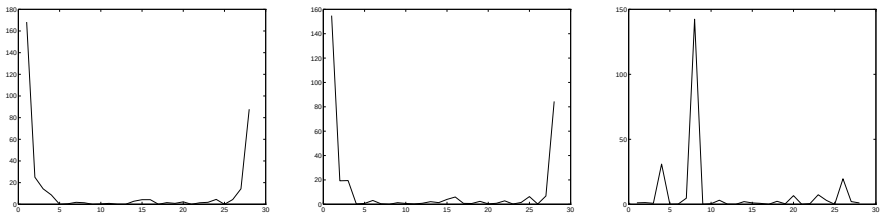



**Fig. 2.** (a) DFT magnitude of the upper triangle in Fig. 1; (b) DFT magnitude of the lower triangle in Fig. 1; (c) $ratio(k)$ vs. $k$.

We want to solve this *spatial discretization* problem while keeping the invariance properties of the MFD; we propose the following procedure:

1. Compute the DFT of the shape boundary $z(n)$, $Z(k)$, using (5);
2. Use the low frequency $[-N_C, +N_C]$ coefficients to reconstruct dense but possibly non-uniform samples $z_{dense}(n)$ of the original boundary:

$$z_{dense}(n) = \sum_{k=-N_C}^{N_C} Z(k) e^{-j\frac{2\pi nk}{N_B}}, \quad n = 0, ..., N_{dense} - 1 \tag{18}$$

3. Use interpolation to trace the dense samples $z_{dense}(n)$ and construct uniform samples $z_{unif}(n)$, n = 0, ..., $N_{unif}$. The uniform samples $z_{unif}(n)$ are *uniformly* spaced on the boundary in terms of arc length;
4. Compute the DFT of $z_{unif}(n)$ to obtain coefficients $Z_{unif}(k), k = -N_C, ..., N_C$.

In Figure 3, we have the two triangles re-sampled using the procedure described above. Note that the re-sampled points match more closely to the points that would have been sampled from the original (continuous) triangle.

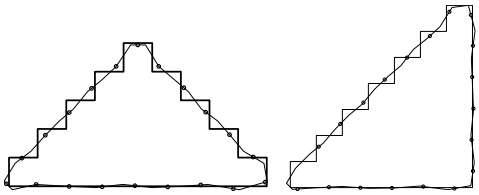


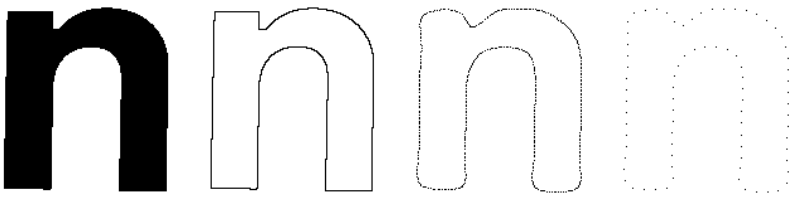

**Fig. 3.** (a) Uniform samples of the upper triangle in Fig. 1; (b) uniform samples of the lower triangle in Fig. 1.

## 5 Experimental Results

We chose to use a set of Roman characters (as opposed to object outlines from the MARS database) to evaluate the proposed method since Roman characters are more commonly available. This will allow other researchers to compare their methods to MFD more easily.

Our test images were created by printing the letters {m, n, u, h, l, t, f} on a laser printer and digitizing the printouts using a scanner. Letters were printed using 256 pt. Helvetica font. To test the robustness of the MFD method, we intentionally misaligned the letters slightly on the scanner, which introduced some boundary noise (Figure 4a).

We tested three aspects of our method: 1. its sensitivity to the choice of parameters $N_C$, $N_{dense}$, and $N_{unif}$, 2. its ability to discriminate between shapes, and 3. its robustness to image transformations.



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

**Fig. 4.** (a) Original image; (b) Extracted boundary; (c) Low frequency reconstruction; (d) Uniform re-sampling.

Figure 4b shows the boundary extracted from the original image of the letter; Figure 4c shows the dense samples reconstructed using 40 MFD coefficients ($N_C = 20$); Figure 4d is the set of uniform samples obtained from Figure 4c.

We can see from Figure 4d that using the first 20 frequencies captures most of information contained in the boundary while reducing segmentation noise.

### 5.1 Sensitivity to choice of parameters

The letters "n" and "f" are used in the following experiments. "n vs. n" denotes the distance between "n" and a rotated version of "n", where the rotation angle is 27 degrees. "n vs. f" denotes the distance between an upright "n" and an upright "f".

1. *Sensitivity to* $N_C$

Table 3 shows Distance vs. $N_C$, where we can see that the MFD is very robust to $N_C$. We have a wide range to choose $N_C$ from – it can range from 5 to 40 without significantly affecting the matching results for the images we used.

Table 3. Distance vs. $N_C$

| $N_C$ | 10 | 15 | 20 | 25 | 30 | 35 |
|---|---|---|---|---|---|---|
| n vs. n | 0.095 | 0.090 | 0.059 | 0.051 | 0.051 | 0.051 |
| n vs. f | 1.984 | 1.806 | 1.930 | 1.713 | 1.907 | 1.705 |

2. *Sensitivity to* $N_{dense}$

$N_{dense}$ is defined as

$$N_{dense} = \frac{boundary\ length}{N_{step}} \tag{19}$$

where $N_{step}$ is the sampling interval. The finer the interval, the larger the number of dense samples. From Table 4 we see that the distance is almost constant for a wide range of $N_{step}$.

Table 4. Distance vs. $N_{step}$

| $N_{step}$ | 2 | 4 | 6 | 8 | 10 | 12 | 14 |
|---|---|---|---|---|---|---|---|
| n vs. n | 0.059 | 0.059 | 0.059 | 0.059 | 0.059 | 0.059 | 0.060 |
| n vs. f | 1.912 | 1.912 | 1.913 | 1.912 | 1.931 | 1.932 | 1.932 |

3. *Sensitivity to* $N_{unif}$
$N_{unif}$ is defined as

$$N_{unif} = (2N_C + 1)multi \tag{20}$$

where *multi* makes $N_{unif}$ a multiple of the number of total frequencies used. *multi* should be at least 1, which corresponds the Nyquist frequency. (see Table 5).

Table 5. Distance vs. *multi*

| multi | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| n vs. n | 0.075 | 0.059 | 0.060 | 0.059 | 0.059 | 0.060 |
| n vs. f | 1.705 | 1.912 | 1.911 | 1.911 | 1.911 | 1.911 |

### 5.2 Discriminatory ability

Tables 6-8 show the ability of MFD to discriminate between shapes. Table 6 shows the MFD distances between the shapes of each letter from the original set. This gives us baseline values on the discriminatory ability of the MFD. The original set is the set of images obtained using scanner as mentioned earlier. Table

Table 6. Distances between letters – original set.

| | m | n | u | h | l | t | f |
|---|---|---|---|---|---|---|---|
| m | 0.000 | 1.802 | 1.809 | 1.625 | 0.893 | 1.802 | 1.512 |
| n | 1.802 | 0.000 | 0.075 | 1.439 | 1.026 | 1.729 | 1.907 |
| u | 1.809 | 0.075 | 0.000 | 1.483 | 0.991 | 1.747 | 1.852 |
| h | 1.625 | 1.439 | 1.483 | 0.000 | 1.081 | 1.583 | 1.557 |
| l | 0.893 | 1.026 | 0.991 | 1.081 | 0.000 | 1.109 | 1.077 |
| t | 1.802 | 1.729 | 1.747 | 1.583 | 1.109 | 0.000 | 1.260 |
| f | 1.512 | 1.907 | 1.852 | 1.557 | 1.077 | 1.260 | 0.000 |

7 shows the MFD distances between letters from the original set and a rotated set. The rotated set was obtained by taking the original set and synthetically rotating each image by 27 degrees (we avoided multiples of 90 degrees since they give the exact same results as in Table 6). Rotations were done using the ImageMagick software package (©1995 E. I. du Pont de Nemours and Company).

Table 7. Distances between original and rotated letters.

| | m | n | u | h | l | t | f |
|---|---|---|---|---|---|---|---|
| m | 0.085 | 1.815 | 1.887 | 1.550 | 0.889 | 1.605 | 1.545 |
| n | 1.795 | 0.079 | 0.133 | 1.448 | 1.035 | 1.860 | 1.736 |
| u | 1.805 | 0.139 | 0.102 | 1.492 | 0.999 | 1.905 | 1.735 |
| h | 1.619 | 1.405 | 1.543 | 0.068 | 1.094 | 1.493 | 1.603 |
| l | 0.837 | 1.154 | 1.034 | 1.077 | 0.016 | 1.109 | 1.070 |
| t | 1.808 | 1.757 | 1.760 | 1.586 | 1.112 | 0.058 | 1.262 |
| f | 1.512 | 1.911 | 1.923 | 1.571 | 1.085 | 1.244 | 0.040 |

Table 8 shows the MFD distances between the original set and a scaled set of images. The scaled set was obtained by scaling the original images by 210%. Scaling was done using the "xv" program (by John Bradley). As expected, "n" and "u" match quite closely, since they are only rotated versions of each other. "h" matches "n" and "u" better than the other letters. We see that discretization (after rotation and scaling) introduces some noise and thus the distances between the same letters are not exactly zero (Tables 7, 8) as is the case in Table 6. But the results indicate that the MFD deals with the discretization effects fairly well. Distances between different letters are always much larger (10 to 100 times) than those between the same letter.

Table 8. Distances between original and scaled letters.

| | m | n | u | h | l | t | f |
|---|---|---|---|---|---|---|---|
| m | 0.025 | 1.873 | 1.848 | 2.081 | 1.762 | 1.674 | 1.602 |
| n | 1.797 | 0.023 | 0.083 | 1.441 | 2.342 | 1.847 | 1.808 |
| u | 1.804 | 0.080 | 0.023 | 1.487 | 2.374 | 1.685 | 1.806 |
| h | 1.621 | 1.324 | 1.333 | 0.022 | 2.017 | 1.582 | 1.601 |
| l | 0.895 | 1.028 | 0.991 | 1.080 | 0.012 | 1.112 | 1.088 |
| t | 1.810 | 1.730 | 1.745 | 1.582 | 1.382 | 0.025 | 1.267 |
| f | 1.518 | 1.911 | 1.884 | 1.574 | 1.232 | 0.891 | 0.034 |

### 5.3 Robustness to transformation

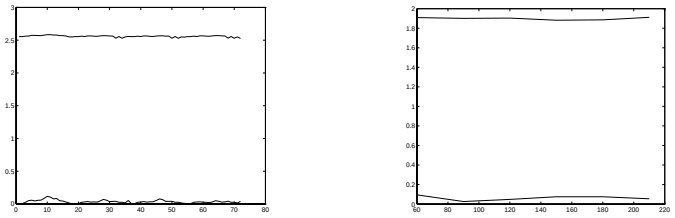
- Translation
  No discretization noise involved. Zero error.

- Rotation
  We plot the distance vs. rotation angle in Figure 5a. The upper curve is the distance between "f" and rotated versions of "n". The lower curve is the distance between "n" and its rotated version. The rotation step is five degrees.

Note how discretization noise affects the distance (the curves are not exactly constant, but have small ripples). However, even if the noise changes each distance a small amount, the overall robustness of the MFD distance is still very good – the average magnitude of the upper curve is about 20 times that of the lower curve.

– Scale
We plot distance vs. scale factor in Figure 5b. The upper curve is the distance between "f" and scaled versions of "n" (from 30% to 210%, with a step size of 30%). The lower curve is the distance between "n" and scaled versions of "n". The magnitude difference is also about a factor of 20, indicating that the MFD is scale invariant.

– Starting point
No discretization noise involved. Zero error.



(a) (b)

**Fig. 5.** "n vs. n" and "n vs. f" for various (a) rotation angles; (b) scale factors

## 6 Conclusions

We presented a new method of shape representation and its distance measure. We compared it with existing FD methods in terms of both computational cost and practical robustness. The main features of our method are:

1. The method is in variant to translation, rotation, scale, and starting point.
2. The method takes into account spatial discretization.
3. The computational cost for feature extraction is low, and for feature matching the cost is extremely low, making the method suitable for real-time multi-user CBR systems.
4. The representation is able to describe complex shapes while remaining relatively compact, reducing the disk space and memory required in the CBR system.

## References


1. C. Faloutsos, "Efficient and effective querying by image content," tech. rep., IBM Research Report, 1993.
2. 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.
3. J. R. Smith and S.-F. Chang, "Tools and techniques for color image retrieval," in *IS & T/SPIE proceedings Vol.2670, Storage & Retrieval for Image and Video Databases IV*.
4. J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C. fe Shu, "The virage image search engine: An open framework for image management," in *SPIE Storage and Retrieval for Still Image and Video Databases IV*.
5. H.-J. Zhang, "Retrieval and browsing: An integrated and content-based solution," in *ACM Multimedia'95*.
6. S. Mehrotra, Y. Rui, M. Ortega-B., and T. S. Huang, "Supporting content-based queries over images in MARS," in *Proc. of IEEE Int. Conf. on Multimedia Computing and Systems*, 1997.
7. Y. Rui, A. C. She, and T. S. Huang, "Automated shape segmentation using attraction-based grouping in spatial-color-texture space," in *Proc. IEEE Int. Conf. on Image Proc.*, 1996.
8. Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega, "A relevance feedback architecutre in content-based multimedia information retrieval systems," in *Proc of IEEE Workshop on Content-based Access of Image and Video Libraries*, 1997.
9. 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.
10. Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega, "Automatic matching tool selection using relevance feedback in MARS," in *The 2nd Int. Conf. on Visual Information Systems*, 1997.
11. M. K. Hu, *Visual Pattern Recognition by Moment Invariants, Computer Methods in Image Analysis*. IEEE Computer Society.
12. E. M. Arkin, "An efficiently computable metric for comparing polygonal shapes," *IEEE Trans on PAMI*, 1991.
13. C. T. Zahn and R. Z. Roskies, "Fourier descriptors for plane closed curves," *IEEE Trans. on Computers*, 1972.
14. E. Persoon and K. S. Fu, "Shape discrimination using fourier descriptors," *IEEE Trans. Sys. Man, Cyb.*, 1977.
15. C. T. Zahn, "A formal description for two-dimensional patterns," in *Proc Int. Joint Conf. Artificial Intelligence*.