VEDANTA, AMRITA, MONISH AND CHITRANSH

SHAPE DISCRIMINATION USING FOURIER DESCRIPTORS

ABSTRACT

Description or discrimination of boundary curves (shapes) is an important problem in picture processing and pattern recognition. Fourier descriptors (FD's) have interesting properties in this respect. First, a critical review is given of two kinds of FD's. Some properties of the FD's are given and a distance measure is proposed, in terms of FD's, that measures the difference between two boundary curves. It is shown how FD's can be used for obtaining skeletons of objects. Finally, experimental results are given in character recognition and machine parts recognition.

INTRODUCTION

One of the problems in picture processing is the classification of objects in a scene.

Such situations arise, for example, in the classification of silhouettes of airplanes, photographed from the ground [8], classification of silhouettes of satellites, and in character recognition.

There are many techniques available to describe closed curves, but there is theoretical and experimental evidence that Fourier Descriptors are a useful set of features

REVIEW OF FOURIER DESCRIPTORS

We assume y is a clockwise-oriented simple closed curve with parametric representation (x(l),y(l)) = Z(l) where l is arc length and 0 < 1 < L. Denote the angular direction of y at point l by the function 0(1). Define then the cumulative angular function 4(1) as the net amount of angular bend between starting point 1 = 0 and point l.

$$\phi^*(t) = \phi\left(\frac{Lt}{2\pi}\right) + t$$

WHERE T RANGES FROM 0 TO 27R. NOTE THAT 0*(T) IS INVARIANT UNDER TRANSLATIONS, ROTATIONS, AND CHANGES OF THE PERIMETER L (SCALE).

$$\phi^*(t) = \mu_0 + \sum_{k=1}^{\infty} A_k \cos(kt - \alpha_k).$$

REVIEW OF FOURIER DESCRIPTORS

Note that we can define an equivalent set of FD's by the equation

$$\phi^*(t) = \sum_{k=-\infty}^{\infty} c_k e^{jkt} \tag{3}$$

where

$$c_k = \frac{1}{2\pi}\int_0^{2\pi} \phi^*(t)e^{-jkt}\,dt.$$

We have again that y is a clockwise-oriented simple closed curve with representation (x(l),y(l)) = Z(l), where l is the arc length along y. A point moving along the boundary generates the complex function u(l) = x(l) + jy(l) which is periodic with period L. The FD's become now

$$a_n = \frac{1}{L}\int_0^L u(l)e^{-j(2\pi/L)nl}\,dl$$

$$u(l) = \sum_{-\infty}^{\infty} a_n e^{jn(2\pi/L)l}.$$

PROPERTIES OF THE FD'S

$$a_n = \frac{1}{L\left(\frac{n2\pi}{L}\right)^2} \sum_{k=1}^{m} (b_{k-1} - b_k) e^{-jn(2\pi/L)l_k}$$

where

$$l_k = \sum_{i=1}^{k} |V_i - V_{i-1}|, \qquad \text{for } k > 0 \text{ and } l_0 = 0$$

and

$$b_k = \frac{V_{k+1} - V_k}{|V_{k+1} - V_k|}, \qquad \text{so } |b_k| = 1.$$

Fig. 2. Example of line patterns.

Fig. 3. Tracing of a line pattern.

Fig. 4. Patterns with thickness.

A. Calculating FD's for a Polygonal Curve

PROPERTIES OF THE FD'SS

B. Line Patterns

A convenient mathematical model of line patterns that are not closed, non overlapping, and have a certain thick-ness.

$$u(l) = f(l) + f'(l)]c(l)$$

C. Optimal Curve Matching

In many pattern recognition applications, the position, size, and rotation of an object are not important. This is because the object can be described by its outer boundary, which is the same regardless of its position, size, or rotation.

$$d^2(\alpha, \beta) = \int_0^1 |u_\alpha(sL_\alpha) - u_\beta(sL_\beta)|^2$$

PROPERTIES OF THE FD'SS

D. Computation of the Area of a Surface

$$S = -\sum_{n=-inf}^{inf} |a_n|^2 n\pi$$

E. Relative Position of a Point with Respect to a Boundary

$$f(s) = Im \sum_{n \neq 0} b_0 * b_n e^{jn2\pi s}$$

EXPERIMENTAL RESULTS

Fig. 8. Skeletons using eight harmonics.

Skeleton Finding Using FD's

EXPERIMENTAL RESULTS

1	2	50		1	3	3	2		10	1
2	1		47						2	
3				46						
4	1				27				2	1
5						47			2	
6	1		1		2		47		2	
7	1		2	1	1			50		2
8	4				4				26	1
9	1			2	7				3	45

error rate = $\frac{76}{500}$

= 15.4%

Fig. 9. Classification result (suboptimal procedure).

$w_k \backslash w_i$	0	1	2	3	4	5	6	7	8	9
0	42				4		1		1	
1	1	50			1	2	2		10	1
2	1		47						2	
3				47						
4	1				43	1	1		3	2
5						47			2	
6			1				46		1	
7			2	1				50		2
8	4								31	1

w_i original class

w_k class classified to

error rate = $\frac{53}{500}$

= 10.6%

Character Recognition

EXPERIMENTAL RESULTS

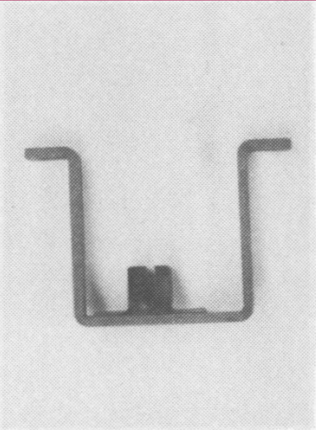


Fig. 14. U-shaped part.

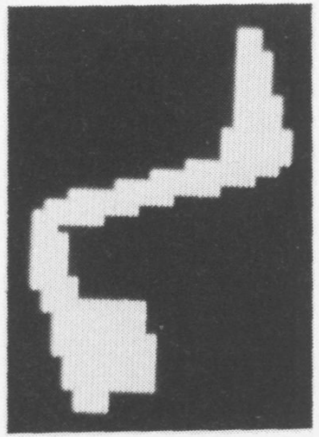


Fig. 15. Silhouette of S-shaped part.

Machine Parts Recognition

DISCUSSION

In this paper, the FD's were computed using (4) or (7). In (4) one can also interpret the parameter l as time and scan the boundary with uniform velocity.

It is possible then to study the effect of moving along the boundary with nonuniform velocity. It is shown in [18] that, when appropriate velocity patterns are chosen, it requires fewer harmonics to store the boundary curve accurately.

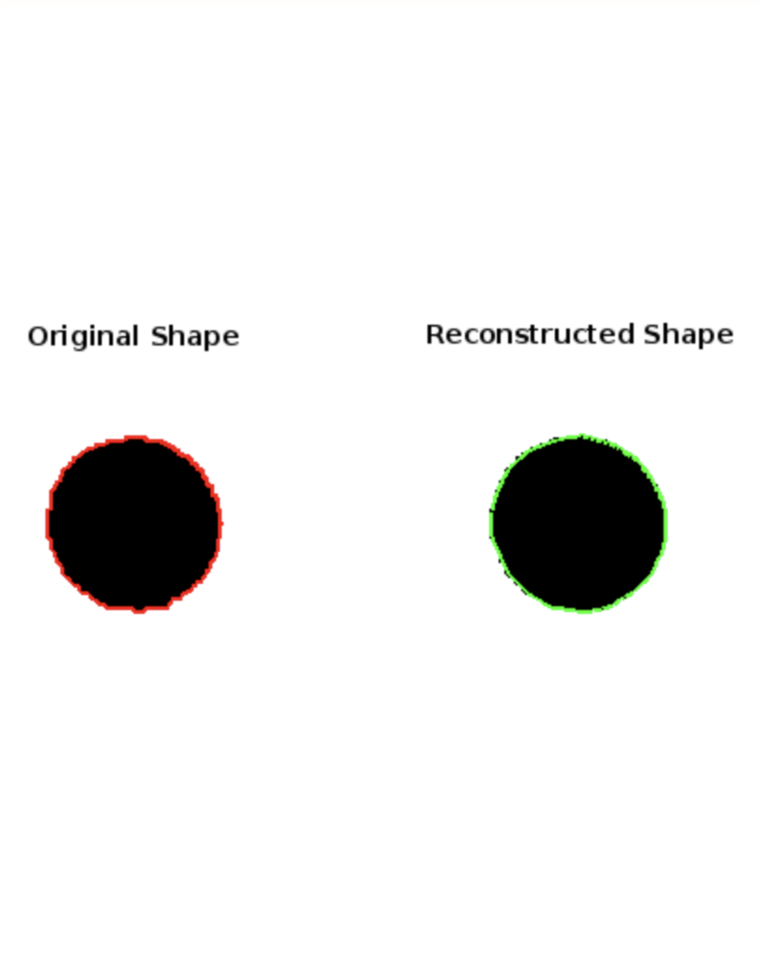
From the experimental results in Section IV, it is evident that FD's are very useful for the problems defined in the Introduction mainly because they allow an easy recognition for rotated and scaled patterns having some noise on their boundaries.

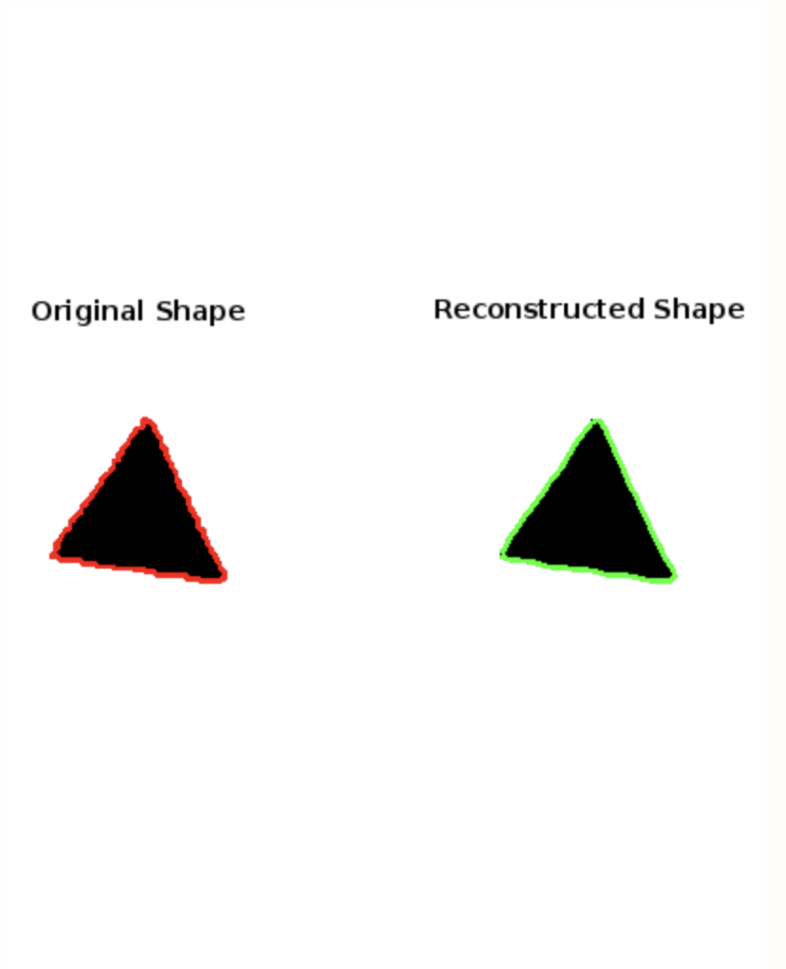
Polygonal approximation and chain encoding should be used in other types of problems where the patterns are best described in terms of specific features present in the boundary

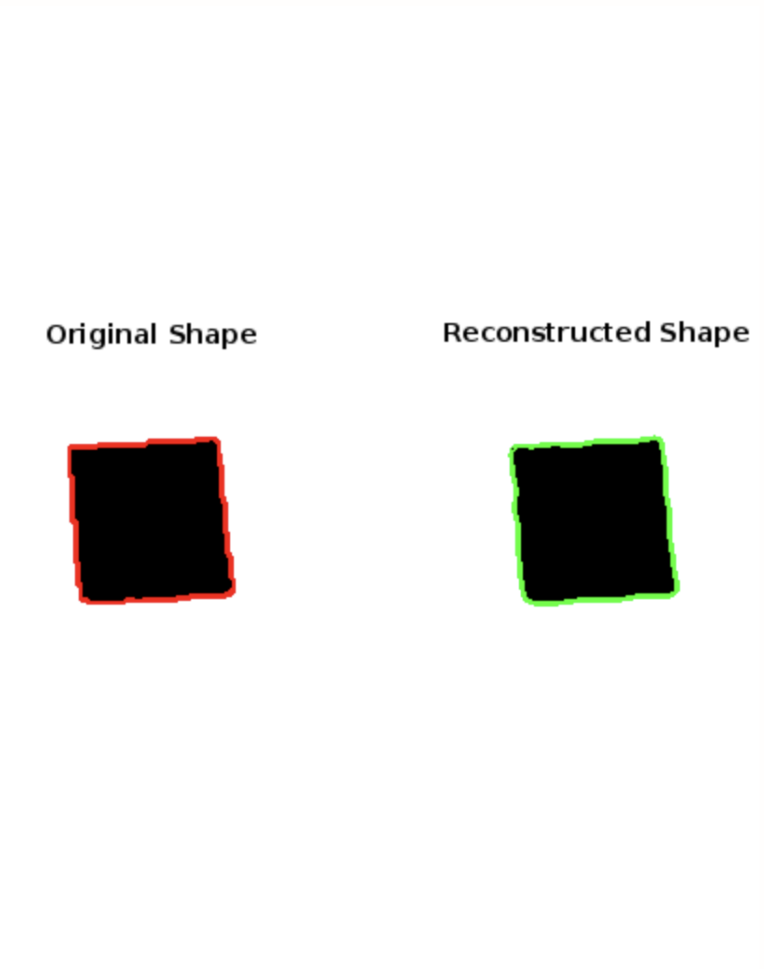
IMPLEMENTATION

```
% Load the image containing the shapes
image = imread('2.png');
image = im2bw(image, 0.5); % Convert to binary image

% Find the boundaries of the shapes
boundaries = bwboundaries(image);

% Iterate through each shape
for k = 1:length(boundaries)
    boundary = boundaries{k};

    % Compute Fourier descriptors
    complex_descriptor = complex(boundary(:, 2), boundary(:, 1));
    fourier_descriptor = fft(complex_descriptor);

    % Remove low-frequency components
    num_coefficients = 20;
    fourier_descriptor(num_coefficients+1:end-num_coefficients) = 0;

    % Reconstruct shape from Fourier descriptors
    reconstructed_descriptor = ifft(fourier_descriptor);
    reconstructed_boundary = [real(reconstructed_descriptor), imag(reconstructed_descriptor)];

    % Plot original and reconstructed shapes
    figure;
    subplot(1, 2, 1);
    imshow(image);
    hold on;
    plot(boundary(:, 2), boundary(:, 1), 'r', 'LineWidth', 2);
    title('Original Shape');
    subplot(1, 2, 2);
    imshow(image);
    hold on;
    plot(reconstructed_boundary(:, 1), reconstructed_boundary(:, 2), 'g', 'LineWidth', 2);
    title('Reconstructed Shape');
end
```

IMPLEMENTATION







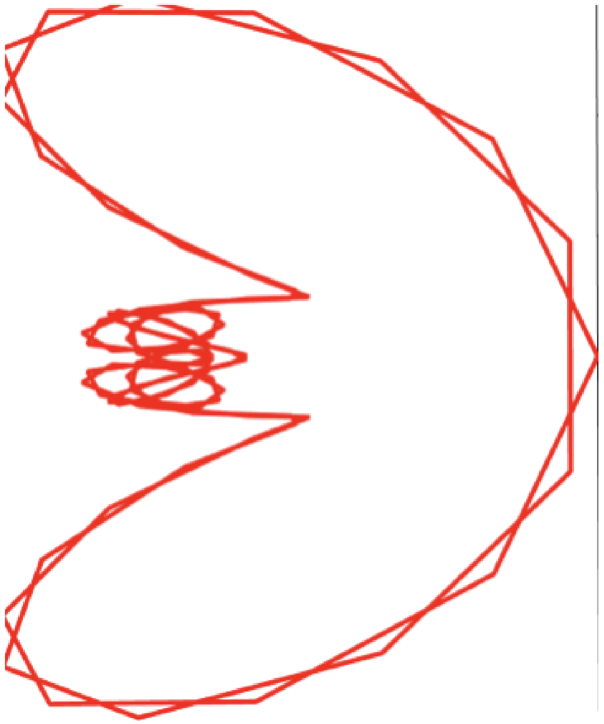
CIRCLE

TRIANGLE

SQUARE

IMPLEMENTATION

Number 3 in eight harmonics equation:



SQUARE

IMPLEMENTATION

Code for the number 3 using eight harmonics :

```
>> % Skeleton Generation of Number 3 using Eight Harmonics

% Parameters
numPoints = 100;       % Number of points in the skeleton
numHarmonics = 8;      % Number of harmonics

% Generate time values
t = linspace(0, 2*pi, numPoints);

% Initialize skeleton matrix
skeleton = zeros(numPoints, 2);

% Generate skeleton using eight harmonics
for harmonic = 1:numHarmonics
    frequency = harmonic * 2;  % Increase the frequency with each harmonic

    % Compute X and Y coordinates of the skeleton points
    x = sin(frequency * t);
    y = cos(frequency * t);

    % Adjust the X and Y coordinates to form the number 3 shape
    if harmonic == 1 || harmonic == 2 || harmonic == 4 || harmonic == 5
        x = x * 1.2;
        y = y * 0.6;
    elseif harmonic == 3
        x = x * 1.5;
        y = y * 0.5;
    elseif harmonic == 6 || harmonic == 7
        x = x * 0.6;
        y = y * 1.2;
    elseif harmonic == 8
        x = x * 0.5;
        y = y * 1.5;
    end

    % Add the current harmonic to the skeleton
    skeleton = skeleton + [x', y'];
end
```

SQUARE


THANK YOU