# **Applications**

# **Laboratory 4**

# **Hu’s invariant moments**

# **Student Name –**

# **Student ID -**

# Introduction

The aim here is to understand and calculate some of features used as image descriptors. Hu’s seven moment invariants are insensitive to changes in scale, position and rotation. These moment invariants have been extensively applied to image pattern recognition, image registration, and image reconstruction.

# Calculation of the moment invariants

1. The grayscale test image (image.png), moments.m, and Moment invariants.m files are downloaded. They are saved in the MATLAB working directory. The files are read and the implementation is understood for calculating the seven invariant moments.



Figure 1

1. The image is read and the size of the image is obtained. The image is padded by one-fourth the image size in all directions with zeros. This is used as the basis for next steps (and named Im1 ).



Figure 2

1. The spatial transformation matrix T1 is created for translation. The translated image Im2 is displayed.



Figure 3

1. The spatial transformation matrix T2 is created for scaling to 0.5 (of original size). The translated image Im3 is displayed.



Figure 4

1. The spatial transformation matrix T3 is created to rotate the image by 45 degrees.

The translated image Im4 is displayed.



Figure 5

1. The spatial transformation matrix T4 is created for rotating the image 90 degrees. The translated image Im5 is displayed.



Figure 6

1. The original image Im1 is flipped from left to right and image Im6 is generated. The mirrored image Im6 is displayed.



Figure 7

1. The moment invariants are calculated using the function Moment invariant() and the images created from step 2 to step 7 (Im1-Im6) are passed from the command line.

# Matlab Results :

>> ans

rows =

360

columns =

360

numberOfColorChannels =

1

Moment\_Invariants =

-0.5998 -5.0693 -4.5907 -5.8652 11.0948 -9.7988 12.1610

Moment\_Invariants =

-0.5998 -5.0693 -4.5907 -5.8652 11.0948 -9.7988 12.1610

Moment\_Invariants =

-0.5998 -5.1407 -4.5693 -5.8729 11.1012 9.7169 11.8389

Moment\_Invariants =

-0.6026 -3.5596 -3.5396 -6.8896 13.1192 8.8524 -12.1063

Moment\_Invariants =

-0.5104 -1.5092 -4.0886 -4.0382 -8.6961 -4.8160 8.1162

Moment\_Invariants =

-0.5998 -5.0693 -4.5907 -5.8652 11.0948 -9.7988 -12.1610

>>

# Discussion :

The values of the moment invariants are computed for each image. The similarities or differences are studied in terms of image features. The moment invariants can be used for image pattern recognition because of the invariant features for translation, scaling as well as rotation. In case of a continuous function, the moments are invariant. But practically, the images are discrete in nature.

# Matlab Code :

I = imread('image.png');

figure(1)

imshow(I)

[rows, columns, numberOfColorChannels] = size(I)

for i=1:360

for j=1:90

I(i,j)=0;

end

end

for i=1:90

for j=1:360

I(i,j)=0;

end

end

for i=271:360

for j=1:360

I(i,j)=0;

end

end

for i=1:360

for j=271:360

I(i,j)=0;

end

end

Im1=I;

figure(2)

imshow(Im1);

T1 = maketform('affine', [1 0 0; 0 1 0; 90 90 1]);

Im2 = imtransform(Im1, T1, ...

'XData',[1 size(Im1,1)], 'YData',[1 size(Im1,2)]);

figure(3)

imshow(Im2)

T2 = maketform('affine', [0.5 0 0; 0 0.5 0; 0 0 1]);

Im3 = imtransform(Im1, T2, ...

'XData',[1 size(Im1,1)], 'YData',[1 size(Im1,2)]);

figure(4)

imshow(Im3);

T3 = maketform('affine', [cos(pi/4) sin(pi/4) 0; -sin(pi/4) cos(pi/4) 0; 0 0 1]);

Im4 = imtransform(Im1, T3, ...

'XData',[-269 size(Im1,1)-270], 'YData',[+111 size(Im1,2)+110]);

figure(5)

imshow(Im4)

T4 = maketform('affine', [cos(pi/2) sin(pi/2) 0; -sin(pi/2) cos(pi/2) 0; 0 0 1]);

Im5 = imtransform(Im1, T4, ...

'XData',[-539 size(Im1,1)-540], 'YData',[1 size(Im1,2)]);

figure(6)

imshow(Im5)

Im6=flipdim(Im1,2);

figure(7)

imshow(Im6)

Moment\_invariants(Im1)

Moment\_invariants(Im2)

Moment\_invariants(Im3)

Moment\_invariants(Im4)

Moment\_invariants(Im5)

Moment\_invariants(Im6)