**Report for Assignment 3 – Part I: Effect of Image Match on Registration**

**Ravikiran Janardhana**

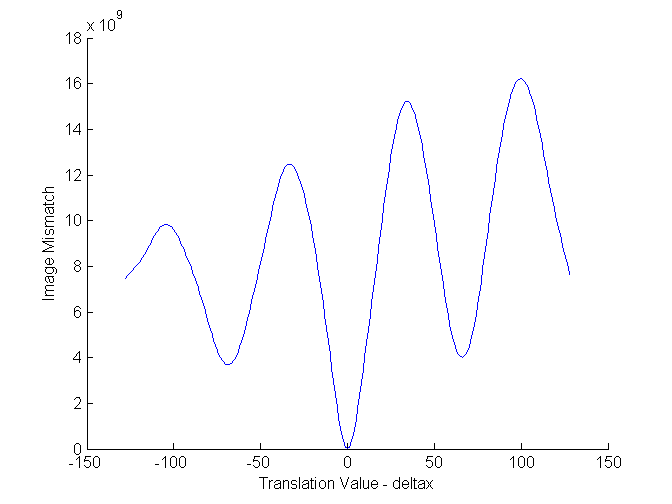
**Medical Image Analysis – COMP 775**

**Fall 2011**

**Tasks completed**

1. Implemented a matlab program to perform intensity transformation of the given fixed image and computed the moving image by applying an ‘x translation (Δx)’ on the result.
2. By varying the ‘x translation from -128 to +128’, the image mismatch value i.e., sum of squared intensity differences was computed for each of the cases.
3. Plotted the image mismatch value on y axis against the x translation (Δx) on x-axis.

**Result**

****

**Code**

1) img\_mismatch.m

% Image mismatch program

% Read the base image

baseimg = double(imread('FixedImage.png'));

maxx = size(baseimg,1);

maxy = size(baseimg,2);

% Generate the intensity transformation

alpha = 1;

beta = 0;

gauss\_sigma = 10;

movimg = intensity\_transformation(baseimg, alpha, beta, gauss\_sigma);

% Compute the moving image which is intensity transformed and translated

tempimg = zeros(maxx,maxy);

startx = 128;

starty = 128;

deltax = -128:128;

mismatch\_arr = zeros(size(deltax,2));

for i = 1:size(deltax,2)

T = [1 0 0; 0 1 0; deltax(i) 0 1];

tform = maketform('affine', T);

[tempimg, xdata, ydata] = imtransform(movimg, tform, ...

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

mismatch\_arr(deltax(i)+129) = sum\_of\_squared\_intensity\_differences( baseimg, ...

tempimg, startx+1, startx+256, starty+1, starty+256);

end

figure;

axis on;

xlabel('Translation Value - deltax');

ylabel('Image Mismatch');

hold on;

plot(deltax, mismatch\_arr(:,1));

disp('Done');

2) sum\_of\_squared\_intensity\_differences.m

function sum = sum\_of\_squared\_intensity\_differences( img1, img2 , startx, endx, starty, endy)

%sum\_of\_intensity\_differences - Computes the sum of squared intensity

%differences from (startx, starty) to (endx, endy)

if nargin < 6

disp('Please pass all the required parameters to sum\_of\_squared\_intensity\_differences, Exiting');

quit;

end

sum = 0.0;

t1 = 0.0;

t2 = 0.0;

term = 0.0;

maxterm = 0.0;

for i = starty : endy

for j = startx : endx

t1 = img1(j,i);

t2 = img2(j,i);

term = t1 - t2;

term = term \* term;

if maxterm < term

maxterm = term;

end

sum = sum + term;

end

end

end

3) intensity\_transformation.m

function output = intensity\_transformation(img, alpha, beta, sigma)

%intensity\_transformation - produces the intensity transformation image

if nargin < 4

disp('Please pass all the required parameters to intensity\_transformation function, Exiting');

quit;

end

length = size(img, 1);

width = size(img, 2);

gauss\_noise = imnoise(img, 'gaussian', 0, sigma\*sigma);

output = zeros(length, width);

for i = 1 : length

for j = 1 : width

output(j,i) = alpha \* img(j,i) + beta + gauss\_noise(j,i);

end

end

end