COMP 775, Fall 2011 Assignment 3: Effect of Image Match on Registration Pizer
Passed out: Monday, 24 October 2011, Due: Monday, 6 November 2011

In this assignment you will create programs to compute 2D image mismatch between a pair of images and will learn how the method of computing image mismatch affects the quality of a registration. The registration that you will be dealing with is the single-parameter registration by translation by Δx in the x direction. You will write a minimization program that minimizes the image mismatch over integer values of Δx by moving Δx by steps of size 1 in the direction of initial decrease (the direction of unit increase or decrease in Δx that yields a decrease of image mismatch) until the next step increases the image mismatch (so it stops at a relative minimum of image mismatch). [That method behaves as a gradient-based optimizer.] That program will call an image mismatch function that you will provide it; you will write multiple image mismatch functions and will be comparing the behaviors of them on images described in the following.

On the blackboard you will be given a 512×512 base image, which will serve as the fixed image. It will have intensities in the range [0, 1024] and will be made up of curly mostly vertical bars of various widths. You will create new moving images by applying one from a set of intensity transformations on the base image and applying an x translation by Δx on the result. You will then plot the image mismatch value as a function of the translation of the moving image, with the image mismatch computed over the interior section of the base image obtained by removing 128 rows and columns from the top, bottom, left, and right of the image (yielding a comparison region of 256×256 pixels). Then you will run the registration program using that image mismatch function and explain its behavior when applied to moving images with transformations of each of the four types below and with translations to be recovered ranging from a few pixels to 100 pixels.

The image mismatch methods that you are to create are as follows:

A) Sum of squared intensity differences
B) Sum over 64×64 squares, into which you divide your image, of the square of the normalized cross correlation over the squares. In the base image comparison region there will be 16 such squares.
C) Sum over the same squares of the Euclidean differences in quantile function on intensity, as modified appropriately to handle both positive and reverse contrast polarity (negative alpha values; see below) relations between the intensities in the fixed and moving images. Explain your modification.
D) Mutual information, obtained by forming the 2D (base intensity, moving image intensity) histogram and computing mutual information from it.

The intensity transformations are as follows. All are of the form: transformed intensity($\underline{x}$) = alpha($\underline{x}$) intensity($\underline{x}$) + beta + zero mean Gaussian distributed pixel intensity noise with standard deviation 10.

1) alpha =1, beta =0
2) alpha =1, beta = 20
3) alpha = -1, beta = 1100
4) alpha($\underline{x}$) = 2D Gaussian with mean at the center of the image and standard deviation 300, i.e. roughly 60% of the linear width of the image, beta=0. The registration program cannot “know” what this function is, only that it is smooth and very gradual.

What to pass in:

For a few images following each type of intensity translation, plots of the image mismatch vs. translation value and the translation value computed by the registration, together with an explanation of the behavior.

A discussion of how image mismatch value and its relation to the intensity relation affect both the range of translations that can be recovered and the likelihood of convergence to the right answer.

The programs for the four types of image mismatch, together with a description of the modification you made in quantile function mismatch for method C.

The minimization program you wrote.