Image Registration Techniques
Homework 3
Due: Wednesday July 7/22 before class

This homework explores Lectures 4 to 8. Answer each of the following questions clearly. Submit your work at the start of lecture. For questions 2 and 3, you are required to write code using the Insight Toolkit and run the corresponding executable using specific images as input. The material that you should return is composed of your source code and the output images generated from the execution of your program. Include your CMakeLists.txt files along with the source code.

1. (**20 points**) Write an algorithm to blend the moving image, I_m, and the fixed image, I_f, based on the backwards affine transformation $\mathbf{A}$ and $\mathbf{t}$. Note that I am not asking for a working implementation here, just a description of the algorithm. Assume the images are 2d, that each image has m rows and n columns, and (for simplicity) that image coordinates and physical coordinates coordinates coincide. Use the notation $I(x, y)$ to indicate the pixel value at row y and column x, with the first pixel at $x = y = 0$.

   There are two main parts to this.

   (a) Determine the size and origin of the resulting image (call it I_b). These will generally be different from the origin and size of either image. This will therefore necessitate transforming both the fixed and moving image, even though the fixed image will be just a translation.

   (b) Step through I_b, determining the intensity of each pixel. The intensity value will depend on whether the pixel is from I_m, from I_f, from neither, or from both. When the pixel is outside both images, the intensity should be 0. When the pixel is inside both images, the intensity should be the average of the two (mapped) intensities.

2. (**20 points**) Write a program that will read a 2D image from a file, apply a ResampleImageFilter and save the resulting image into a file. Execute this program using as input the image:

   Insight/Examples/Data/BrainProtonDensitySlice.png

   Use the IdentityTransform as spatial transform, and a NearestNeighbor interpolator. Execute this program for the following cases

   (a) Output image size equals Input image size, Output image pixel spacing equals Input pixel spacing, Output image origin equals central point of the Input image.

(b) Output image size equals half of Input image size, Output image pixel spacing equals double of the Input pixel spacing, Output image origin equals Input image origin.

(c) Output image size equals double of Input image size, Output image pixel spacing equals half of the Input pixel spacing, Output image origin equals Input image origin.

3. (**20 points**) Modify the program from the previous exercise in order to use a CenteredRigid2DTransform and a LinearInterpolator. Execute this program using as input the image:

   Insight/Examples/Data/BrainProtonDensitySliceBorder20.png

   Set the origin of the output image to (0, 0), the output pixel spacing equal to the input image pixel spacing, the output image size in pixels equal to the input image size in pixels.

   Execute this program for the following cases

   (a) Set the rotation center of the transform to (0, 0), set the translation to (0, 0) and the rotation angle to 10 degrees.

   (b) Set the rotation center of the transform to the middle point of the image, set the translation to (0, 0) and the rotation angle to 10 degrees.

   (c) Set the rotation center of the transform to the middle point of the image, set the translation to (20, 0) and the rotation angle to 10 degrees.