INTRODUCTION TO DIGITAL IMAGE PROCESSING

PRACTICE PROBLEMS – GEOMETRIC OPERATIONS

1. (a) Try out the `rotateImage` function that was discussed in class and is posted on the course Web site. Using the following commands, apply a 30° clockwise rotation about the centre of the image `pout.tif` and display the original and rotated images:

```
im = imread('pout.tif');
subplot(221), imshow(im), title('Original image')
im1 = rotateImage(im, 30);
subplot(222), imshow(im1), title('After rotateImage')
```

   (b) Type the following into MATLAB:

```
im2= imrotate(im, -30, 'nearest', 'crop');
subplot(223), imshow(im2)
```

   to compare the output of the built-in MATLAB function `imrotate` to our function `rotateImage`. Why is there a negative on the 30? What is the effect of leaving out the parameter `crop`? Use the `help` function to figure these out.

2. Please read about spatial transforms (i.e., geometric operations) in MATLAB by typing '`doc images`' at the MATLAB prompt and selecting "Geometric Transformation and Image Registration".

   Now that you've finished reading, load the `pout.tif` image into a MATLAB variable called `im`. Do the following using the `maketform` and `imtransform` commands.

   (a) State what would you type at the MATLAB prompt to translate the image by 17 pixels upwards and 20 pixels to the right while making sure that the output image matrix has the same origin coordinates as the input image and is the same size as the input. Plot the input (`im`) and output images side-by-side.

   (b) State what would you type at the MATLAB prompt to rotate `im` by 30 degrees clockwise around the origin while making sure that the output image matrix has the same origin coordinates as the input image and is the same size as the input. Plot the input (`im`) and output images side-by-side.

   (c) State what would you type at the MATLAB prompt to scale `im` by a factor of 0.5 in the *j* direction and a factor of 0.8 in the *i* direction? The scaling should be done with respect to the origin while making sure that the output image matrix has the

same origin coordinates as the input image and is the same size as the input. Plot the input (`im`) and output images side-by-side.

3. The function `register.m` that is on the course Web site implements the algorithm for landmark-based registration that was described in class. The function reads in user-specified source and target images, displays them side-by-side, allows the user to select corresponding landmarks on the source image and then the target using the `ginput` function, and then computes and returns the parameters `theta`, `di`, `dj` and `s` that represent the rotation angle, *i*-component of translation, *j*-component of translation and a single scaling factor.

   Apply the `register` function to the images `'maple2.jpg'` (source) and `'maple.jpg'` (target). The images are available on the course Web site.

   What did you estimate the parameters of the transformation to be?

   What features of the image did you use as landmarks and why?

   Would you expect the parameters you estimated to be identical to what your classmates found? Explain why or why not.

   What would you type at the MATLAB prompt to apply the geometric transformation that you estimated to `maple2.jpg` in order to register it to `maple.jpg`?

   NOTE: The latest versions of MATLAB do have landmark-based registration built-in as well as advanced registration functions (intensity-based, demons, etc) that are taught in the second semester course. The `register.m` function developed in class illustrates the mathematical foundation of the simplest registration function.