Image Registration Techniques
Homework 5
Due: Wednesday July 8/5 before class

This homework explores Lectures 11 and 12—the introductory lectures on vxl and rgrl. Submit your code and any output images.

1. (**20 points**) Write a vxl program that reads in a list of points, one point per line, and computes the center of the points and the principal directions (2nd moments). The first line of the input file will specify the dimension of the space (at least 2) the points reside in. The name of the input file will be the only command-line argument to your program.

   If $\mathbf{p}_i, i = 1, ..., N$, is the sequence of vectors, then the center (mean) of the points is

   $$\mu = \frac{1}{n} \sum_{i=1}^{n} \mathbf{p}_i$$

   The moments and principal directions are found by computing the eigenvalues and eigenvectors of the scatter matrix:

   $$X = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{p}_i - \mu)(\mathbf{p}_i - \mu)^T$$

   Use functions and classes from **vnl** and **vnl_algo** to do the calculations. Two test files will be posted on the course website (**hw5_q1a.pts** and **hw5_q1b.pts**). Submit the source code for this example as well as the output from the two test files.

2. (**20 points**) Write a program that applies Gaussian smoothing to an image and then computes the gradient magnitude using vertical and horizontal Sobel kernels. Use code from **vil** and **vil_algo** as much as possible! Your program should have 3 command-line arguments: the standard deviation (σ) of the Gaussian, the input image and the output image. Submit your source code and the results of running the program on the two posted example images (cell.png and retina_ex.png) with $\sigma = 1$, $\sigma = 2$, $\sigma = 4$.

3. (**20 points**) Write a program to register two images of size 600×600 using the registration engine **rgrl_feature_based_registration** in two "resolutions". The original image is down-sampled by a factor of 2 to generate the coarser image. The fixed image is generated by rotating the moving image by 10, and translating by (5,-3) units of the same resolution. Features from the 4 images are in **hw5_q3_moving.pts**, **hw5_q3_moving_coarse.pts**, **hw5_q3_fixed.pts**, and **hw5_q3_fixed_coarse.pts**. The program has to first register the features of the coarser resolution with a translation transformation, which is initially an identity transform. The result is then taken

as the initial guess for the registration of features of the original resolution with a similarity transformation. Submit your source code, and the output giving the transformation at the end of the first and second "resolutions".