CSE166 – Image Processing – Homework #6

Instructor: Prof. Serge Belongie
`http://www-cse.ucsd.edu/~sjb/classes/fa02/cse166`
Due (in class) 3:00pm Wed. Nov. 20, 2002.

Written exercises

1. Restate the definition (e.g. from your old linear algebra textbook) of *positive semidefinite*. Prove that the second moment matrix (as defined in class) is positive semidefinite.

2. The *homogeneous coordinates* for a point with cartesian coordinates (x, y) are obtained by adding a third coordinate of 1 to the cartesian coordinates, i.e. (x, y) becomes $(x, y, 1)$. Show how one can solve for the affine parameters all at once (instead of separating out the translation part as we did in class) using homogeneous coordinates and the following parameter matrix (in place of A and $\mathbf{t}$):

$$B = \begin{bmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{bmatrix}$$

Matlab exercises

1. Estimating Affine Transforms.

   This exercise makes use of the pointset data contained in `coords.txt`, which is available on the course web page. The data file consists of two lists of six 2D coordinates, arranged in corresponding order.

   (a) Plot the two pointsets on the same set of axes. Use the `'x-'` pointmarker/linetype for the first set and `'o-'` for the second.

   (b) Solve for the least-squares affine transform (consisting of a 2×2 matrix A and a 2×1 translation vector $\mathbf{t}$) that maps the first pointset onto the second. Display the values you obtain for A and $\mathbf{t}$.

   (c) Use the estimated affine transform to align the two pointsets, and make a plot to show the alignment. Compute and display the sum of the squared Euclidean distances between the corresponding point pairs before and after alignment.

   Things to turn in:

   - Code listing, plots, and program output for each step.

2. Windowed-Image Second Moment Matrix.

   (a) Compute the eigenvalues $(\lambda_{max}, \lambda_{min})$ of the windowed-image second moment matrix for the checkerboard image in the figure for GW Problem 10.18(right). Use a window size of 3×3. On top of the original checkerboard image, plot the coordinates (use the `'.'` pointmarker) of all pixels for which $\lambda_{min} > \tau$, with τ set to 80% of the maximum value of λ_{min} over the whole image. The resulting coordinates should fall on or near the corners of the squares in the image.

   (b) Repeat the above steps for the fingerprint image in GW Figure 10.29(a); this time set τ to 20% of the maximum value of λ_{min} over the whole image. The resulting coordinates should fall on or near the minutia points of the fingerprint, but due to noise, there will also be many spurious responses. As a final step, compute ϕ, the angle of the principal eigenvector for each pixel, and display it as an image.

   Things to turn in:

   - Code listing and plots for each step.