CS 485 - HW4
Don Ma

1. Corner Detection With the Harris Corner Detector, using a smoothing function width of 7pixels and a standard deviation of 1.5
   (a) Comparing Harris Corner Detection of a rotated image Theoretically the rotated image and the original image should find the same corners. In practice this is almost completely true, but there are some artifacts that caused a few differences.
   (b) Comparing the Harris Corner Detection on scaled images, results in diffrent outcome. Each reduction of resolution introduces and removes feature points. This is because at a large resolution, the algorithm can find details which it can declare as a unique point; this is lost at lower resolutions. Similarly, at small resolutions, features that were previously too large to be considered are now at a managable size.
   (c) the Gradient images are shown on the webpage
2. Harris Point Correspondence
   The image for SSD matching is shown on the webpage
3. SIFT Correspondence
   (a) SIFT works by looking at the image at different sizes and resolutions. Then we must detect the edges, this is usually done with Laplacian of Gaussian. With these we look for the local maxima, but there are a lot of these, so we remove the points that have small gradients (these will be points that lie near corners). To make the image rotation invarient, we also look at the dominant gradient orientation. We then make descriptors for each of these features, We look at nearby pixels, and create a orientation histogram of the smaller 4×4 subregions. Comparing this information can give us very confident matches.
   (b) The SIFT algorithm seems to perform better in all cases. The SSD algorithm is too vague, and will only work with sterioscopic images. However in our implementation, the SIFT algorithm out performs the SSD comparison even with the sterioscopic images; finding more accurate and numerous links. With the SIFT method, from slightly shifted perspectives, we can confidently create a transformation and compute the differences in perspective.
4. Edge The tresholds in the Edge Detection adjust the amount of detail that appears as edges. This could range from extremely noisy edges (up to an edge for each pixel), or very uninformative edges (down to no edges).
5. Optical Flow
   (a) Starting with the objective squared sum function $E(u) = \sum_w (\nabla I^T u + I_t)^2$ We look for the minimization by setting the derivitive to 0

$$\nabla E(u) = 2 \times \sum_w \nabla I(\nabla I^T u + I_t)$$

$$\nabla E(u) = 2 \times \sum_w (\begin{bmatrix} \frac{\partial I}{\partial x}^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \frac{\partial I}{\partial y}^2 \end{bmatrix} u + \begin{bmatrix} \frac{\partial I}{\partial x}\frac{\partial I}{\partial t} \\ \frac{\partial I}{\partial y}\frac{\partial I}{\partial t} \end{bmatrix})$$

$$0 =$$

   (b) the matrix is singluar when the determinant is zero. In our matrix G, this would be when

$$\frac{\partial I}{\partial x}^2 \frac{\partial I}{\partial y}^2 - \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}\frac{\partial I}{\partial x}\frac{\partial I}{\partial y} = 0$$

   G should be singular when $\frac{\partial I}{\partial x} = \frac{\partial I}{\partial y}$