

## CSE 252B: Computer Vision II, Winter 2016 – Assignment 4

Instructor: Ben Ochoa

Due: Tuesday, March 1, 2016, 11:59 PM

### Instructions

- Review the academic integrity and collaboration policies on the course website.
- This assignment only contains programming problems that must be completed using MATLAB.
- You must prepare a report describing the problems, and your solutions and results. The report must contain enough information for a reader to understand the problems and replicate your work without also having the assignment (i.e., this document).
- Your report will be a pdf file named `CSE_252B_hw4_lastname_studentid.pdf`, where `lastname` is your last name and `studentid` is your student ID number. The report must be prepared using  $\LaTeX$ .
- All of your MATLAB source code must be included in an appendix of your report. You may find the `listings` package useful for this.
- You must create a zip file named `CSE_252B_hw4_lastname_studentid.zip`, where `lastname` is your last name and `studentid` is your student ID number. This zip file will contain the pdf file and a directory named `code` that contains all of your MATLAB source code.
- Submit your completed assignment by email to `bochoa@ucsd.edu` and `nkinkade@eng.ucsd.edu`. The subject of the email message must be CSE 252B Assignment 4. Attach the zip file to the message.
- It is highly recommended that you begin working on this assignment early to ensure that you have sufficient time to correctly implement the algorithms and prepare a report.

### Problems

#### 1. Programming: Automatic estimation of the planar projective transformation (110 points)

##### (a) Feature detection (20 points)

Download input data from the course website. The file `price_center20.JPG` contains image 1 and the file `price_center21.JPG` contains image 2. In your report, include a figure containing the pair of input images.

For each input image, calculate an image where each pixel value is the minor eigenvalue of the gradient matrix

$$\mathbf{N} = \begin{bmatrix} \sum_w I_x^2 & \sum_w I_x I_y \\ \sum_w I_x I_y & \sum_w I_y^2 \end{bmatrix}$$

where  $w$  is the window about the pixel, and  $I_x$  and  $I_y$  are the gradient images in the  $x$  and  $y$  direction, respectively. Set resulting values that are below a specified threshold value to zero (hint: calculating the mean instead of the sum in  $\mathbf{N}$  allows for adjusting the size of the window without changing the threshold value). Apply an operation that suppresses (sets to 0) local (i.e., about a window) nonmaximum pixel values in the minor eigenvalue image. Vary these parameters such that around 600–650 features are detected in each image. For resulting nonzero pixel values, determine the subpixel feature coordinate using the Förstner corner point operator.

In your report, state the size of the feature detection window (i.e., the size of the window used to calculate the elements in the gradient matrix  $\mathbf{N}$ ), the minor eigenvalue threshold value, the size of the local nonmaximum suppression window, and the resulting number of features detected in each image. Additionally, include a figure containing the pair of images, where the detected features (after local nonmaximum suppression) in each of the images are indicated by a square about the feature, where the size of the square is the size of the detection window.

**(b) Feature matching (15 points)**

Determine the set of one-to-one putative feature correspondences by performing a brute-force search for the greatest correlation coefficient value (in the range  $[-1, 1]$ ) between the detected features in image 1 and the detected features in image 2. Only allow matches that are above a specified threshold value (note that calculating the correlation coefficient allows for adjusting the size of the matching window without changing the threshold value). Vary these parameters such that around 200 putative feature correspondences are established. Optional: constrain the search to coordinates in image 2 that are within a proximity of the detected feature coordinates in image 1.

In your report, state the correlation coefficient threshold value and the resulting number of putative feature correspondences (i.e., matched features). Additionally, include a figure containing the pair of images, where the matched features in each of the images are indicated by a square about the feature, where the size of the square is the size of the matching window, and a line segment is drawn from the feature to the coordinates of the corresponding feature in the other image (see Fig. 4.9(e) in the Hartley & Zisserman book as an example).

**(c) Outlier rejection (15 points)**

The resulting set of putative point correspondences should contain both inlier and outlier correspondences (i.e., false matches). Determine the set of inlier point correspondences using the M-estimator Sample Consensus (MSAC) algorithm,

where the maximum number of attempts to find a consensus set is determined adaptively. For each trial, you must use the 4-point algorithm (as described in lecture) to estimate the planar projective transformation from the 2D points in image 1 to the 2D points in image 2. Calculate the (squared) Sampson error as a first order approximation to the geometric error.

In your report, describe any assumptions, including the probability  $p$  that at least one of the random samples does not contain any outliers (used to determine the number of attempts to find a consensus set), and the probability  $\alpha$  that a given data point is an inlier and the variance  $\sigma^2$  of the measurement error (both used to determine the distance threshold; hint: this problem has codimension 2). State the resulting number of inliers and the number of attempts to find the consensus set.

Additionally, include a figure containing the pair of images, where the inlier features in each of the images are indicated by a square about the feature, where the size of the square is the size of the matching window, and a line segment is drawn from the feature to the coordinates of the corresponding feature in the other image (see Fig. 4.9(g) in the Hartley & Zisserman book as an example).

(d) **Linear estimation (15 points)**

Estimate the planar projective transformation  $\mathbf{H}_{\text{DLT}}$  from the resulting set of inlier correspondences using the direct linear transformation (DLT) algorithm (with data normalization). You must express  $\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$  as  $[\mathbf{x}'_i]^\perp \mathbf{H}\mathbf{x}_i = \mathbf{0}$  (not  $\mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = \mathbf{0}$ ), where  $[\mathbf{x}'_i]^\perp \mathbf{x}'_i = \mathbf{0}$ , when forming the solution. Include the numerical values of the resulting  $\mathbf{H}_{\text{DLT}}$ , scaled such that  $\|\mathbf{H}_{\text{DLT}}\|_{\text{Fro}} = 1$ , in your report with sufficient precision such that it can be evaluated (hint: use `format longg` in MATLAB prior to displaying your results).

(e) **Nonlinear estimation (45 points)**

Use  $\mathbf{H}_{\text{DLT}}$  and the Sampson corrected points (in image 1) as an initial estimate to an iterative estimation method, specifically the *sparse* Levenberg-Marquardt algorithm, to determine the Maximum Likelihood estimate of the planar projective transformation that minimizes the reprojection error. You must parameterize the planar projective transformation matrix and the homogeneous 2D “scene” points that are being adjusted using the parameterization of homogeneous vectors (see section A6.9.2 (page 624) of the textbook, and the corrections and errata).

In your report, show the initial cost (i.e., the cost at iteration 0) and the cost at the end of each successive iteration. Show the numerical values for the final estimate of the planar projective transformation matrix  $\mathbf{H}_{\text{LM}}$ , scaled such that  $\|\mathbf{H}_{\text{LM}}\|_{\text{Fro}} = 1$ , in your report with sufficient precision such that it can be evaluated.