CSE 252B: Computer Vision II, Winter 2017 – Assignment 1

Instructor: Ben Ochoa

Due: Tuesday, January 24, 2017, 11:59 PM

Instructions

- Review the academic integrity and collaboration policies on the course website.
- This assignment must be completed individually.
- This assignment contains both math and programming problems.
- Programming aspects of this assignment must be completed using MATLAB.
- Unless specified below, you may not use MATLAB functions contained in the computer vision toolbox. Use the MATLAB `which` command to determine which toolbox a function is contained in. If you are unsure about using a specific function, then ask the instructor for clarification.
- 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_hw1_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 submit your report on Gradescope.
- Additionally, you must create a zip file named `CSE_252B_hw1_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 `bmankala@eng.ucsd.edu` and `atripath@eng.ucsd.edu`. The subject of the email message must be `CSE 252B Assignment 1`. 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. **Line-plane intersection (5 points)**

   The line in 3D defined by the join of the points $\mathbf{X}_1 = (X_1, Y_1, Z_1, T_1)^\top$ and $\mathbf{X}_2 = (X_2, Y_2, Z_2, T_2)^\top$ can be represented as a Plücker matrix $\mathtt{L} = \mathbf{X}_1\mathbf{X}_2^\top - \mathbf{X}_2\mathbf{X}_1^\top$ or pencil of points $\mathbf{X}(\lambda) = \lambda\mathbf{X}_1 + (1-\lambda)\mathbf{X}_2$ (i.e., $\mathbf{X}$ is a function of λ). The line intersects the plane $\boldsymbol{\pi} = (a, b, c, d)^\top$ at the point $\mathbf{X}_\mathtt{L} = \mathtt{L}\boldsymbol{\pi}$ or $\mathbf{X}(\lambda_{\boldsymbol{\pi}})$, where $\lambda_{\boldsymbol{\pi}}$ is determined such that $\mathbf{X}(\lambda_{\boldsymbol{\pi}})^\top\boldsymbol{\pi} = 0$ (i.e., $\mathbf{X}(\lambda_{\boldsymbol{\pi}})$ is the point on $\boldsymbol{\pi}$). Show that $\mathbf{X}_\mathtt{L}$ is equal to $\mathbf{X}(\lambda_{\boldsymbol{\pi}})$ up to scale.

2. **Line-quadric intersection (5 points)**

   In general, a line in 3D intersects a quadric $\mathtt{Q}$ at zero, one (if the line is tangent to the quadric), or two points. If the pencil of points $\mathbf{X}(\lambda) = \lambda\mathbf{X}_1 + (1-\lambda)\mathbf{X}_2$ represents a line in 3D, the (up to two) real roots of the quadratic polynomial $c_2\lambda_\mathtt{Q}^2 + c_1\lambda_\mathtt{Q} + c_0 = 0$ are used to solve for the intersection point(s) $\mathbf{X}(\lambda_\mathtt{Q})$. Show that $c_2 = \mathbf{X}_1^\top\mathtt{Q}\mathbf{X}_1 - 2\mathbf{X}_1^\top\mathtt{Q}\mathbf{X}_2 + \mathbf{X}_2^\top\mathtt{Q}\mathbf{X}_2$, $c_1 = 2(\mathbf{X}_1^\top\mathtt{Q}\mathbf{X}_2 - \mathbf{X}_2^\top\mathtt{Q}\mathbf{X}_2)$, and $c_0 = \mathbf{X}_2^\top\mathtt{Q}\mathbf{X}_2$.

3. **Programming: Automatic feature detection and matching (35 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

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

   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. Calculate the gradient images using the five-point central difference operator. Set resulting values that are below a specified threshold value to zero (hint: calculating the mean instead of the sum in $\mathtt{N}$ 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 $\mathtt{N}$), 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 correlation coefficient threshold value (note that calculating the correlation coefficient allows for adjusting the size of the matching window without changing the threshold value). Further, only allow matches that are above a specified distance ratio threshold value, where distance is measured to the next best match for a given feature. 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 size of the proximity window (if used), the correlation coefficient threshold value, the distance ratio 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).