

CS 6320, 3D Computer Vision

Spring 2012, Prof. Guido Gerig

Assignment 2: Feature Detection and Feature-based Alignment

Out: Wed Feb-15-2012
Due: Wed Feb-29-2012, midnight (theoretical and practical parts)
TA: Kathlea Quebbeman
Office hours (TA and Instructor): Mo/Tue 3pm to 5pm

Required Readings: Computer Vision, Richard Szelisky, Chapters 4 and 6
Slides to these chapters as provided on course web-page
(http://www.sci.utah.edu/~gerig/CS6320-S2012/CS6320_3D_Computer_Vision.html)

Grading

Theoretical problems: These serve as your own study of the material using the textbook and all materials provided on our course homepage. Detailed solutions will be provided.

Practical problem: Grading will primarily concern your solution strategy and solution of the camera calibration, and the report that describes your project, your development of the methodology, results, and critical assessments of the results.

Late submissions: Late submissions result in 10% deduction for each day. The assignment will no longer be accepted 3 days after the deadline.

Honor Code, Misconduct: Please read the School of Computing misconduct information and acknowledge that you understood by signing the form (see class web-page). Bring the signed form to class, latest on February 13. The assignment will not be graded without this document with your personal signature.

I. Theoretical Problems

Write a report on your solutions for the theoretical problems. This report can be handed in on paper during the class lecture on Monday February 13 or can be added to the electronic pdf/Word report of the theoretical part (submitted to the CADE handin system, see details on course web page).

Problem 1: Pinhole Camera

How does an image change if the focal length is varied?

Problem 2: Perspective Projection

Use the perspective projection equations to explain why, in a picture of a face taken frontally and from a very small distance, the nose appears much larger than the rest of the face. Can this effect be reduced by acting on the focal length?

Problem 3: Simple Disparity

Assume two cameras in the standard stereo geometry arrangement. Let the focal separation (baseline) be $b=5\text{cm}$, let the focal length be $f=16\text{mm}$, let the horizontal and the vertical pixel width be uniformly $l=17\ \mu\text{m}$. Assume two scene points P1 and P2 for which correctly determined corresponding image point pairs are given,

$$(x_{1,left}, y_{1,left}) = (80,30), \text{ and } (x_{1,right}, y_{1,right}) = (80,30) \\ (x_{2,left}, y_{2,left}) = (81,30), \text{ and } (x_{2,right}, y_{2,right}) = (28,30).$$

What are the Euclidean distances between these points P1 and P2 and the focal point of the left camera? What are the differences δX , δY , δZ of these two points in the three-dimensional space of the scene objects? Outline the relative depth values (i.e. the distances to the focal point of the left camera) for varying integer disparity values in the interval $[20,80]$ for the specified parameters.

I. Practical Problem

Feature Detection

This objective of this assignment is to learn about feature detection, to implement a basic feature detection algorithm, and to discuss its performance with practical tests which include image stitching.

Problem 4: Harris Corner Detector

Hint: Matlab imaging novices can use the Matlab program for edge detection provided for the Matlab-Imaging Tutorials as a framework (see course Web-page downloads at the bottom of the page). For a description of the Harris corner detector, see slides on corner detection and section 4.11, pages 185-189 or the Szelisky course book.

1. Implement the Harris Corner Detector following the instructions given in the class slides. Basically, you need the partial derivatives of the image in x and y direction $(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$ commonly called (I_x, I_y) . Following the slides, you then build the squares and mixture terms for the matrix terms, and calculate the corner response function. In this first implementation, don't use any neighborhood weighting but calculate the function at every pixel. Please note that the results can be positive and negative, so that you have to calculate the unsigned value for your image output.
2. Implement a local maximum detection method for 3x3 pixel neighborhoods. The rule for a local maximum is to check if the center pixel is larger than any of its 8 neighbors. The output will be the points that pass this maximum tests with its corner response values, whereas all other points are set to 0.
3. Apply the maximum detection to your Harris corner output image.
4. Harris proposed to reduce noise by averaging the individual values $I_x^2, I_y^2, I_x I_y$ across a pixel neighborhood of 3x3 or 5x5, e.g.. After generating these smoothed arrays of these matrix elements, calculate the corner response function at each pixel and display the result as an image.

5. In practical implementations where we would like to generate features at different scales (i.e. at different blurring levels of the input images), we first blur the original images at different scales and then apply the pointwise corner detection on this smoothed images.
 - (a) **NON IMAGE PROCESSING STUDENTS:** Implement a 3x3 pixel averaging filter for image smoothing. Apply this filter repeatedly to your input image (1, 2, 4, 16times) to create a set of images with different smoothing.
 - (b) **IMAGE PROCESSING STUDENTS:** Apply a Gaussian filter as you did in the image processing course (remember that the Gaussian filter is separable, i.e. you can first apply a 1-D Gaussian in horizontal and then the same 1-D Gaussian in vertical direction). Choose different Gaussian width (e.g. $\sigma = 1, 2, 4, 8$) to create images with different smoothing levels.
6. Apply the Harris corner detection (without neighborhood weighting) to each image of your set of smoothed images. Then apply the maximum detection to each of these images.
7. Display the original image, the filtered images, the results of the Harris corner detection, and the results after maximum detection. Please note that dark images with a few white dots do not print well, better would be to invert the image brightness for display purposes (0=white, corner response dark).
8. Discuss your results and eventual positive and negative properties of this corner filtering.

Problem 5: Manually Assisted Image Stitching

Chapter 6.3 of our Szelisky course book describes “Panography”, i.e. an image stitching of images that are obtained by parallel translation only (see Figure 6.3).

1. Use your camera to create 3 images by shifting your camera position in a parallel fashion (e.g., you could hold your camera to a wall and move it along the wall or choose other creative ways to ensure that the image planes of all images are part of the same plane). Note that you can also move your camera up and down as shown in Fig. 6.3, but do NOT rotate the camera. Ensure that the resulting images show considerable overlap, i.e. in Fig. 6.3 all three images have a joint region of overlap.
2. Choose your interactive point selection method as used in assignment 1 to mark a few corresponding keypoints between neighboring pairs of images.
3. Calculate the global translation components as explained in equation 6.12 for each image.
4. Use the translation results to map all three images into a joint panorama image. Best is to define the middle image as the reference, define an extended border around this image, and map the other images into this reference frame. For overlapping pixels that get multiple values, a simple way is to calculate the average pixel intensity.
5. **Additional Experiment:** Apply your Harris corner detection (with maxima detection) at a smoothing scale which you heuristically choose based on the results in the previous section to all the three images. Use your manual mouse-based keypoint selection on the corner response images to evaluate if it would assist you in an easier finding of corresponding image locations, which is a first step towards an automatic scheme to automatically find corresponding keypoints.
6. Description of your procedures and discussion of results.

Optional Bonus Problem

See exercise 6.2 item 8 Szelisky Book: Extend your parametric motion model to include rotations and scale, i.e., the similarity transform given in Table 6.1. Images could therefore be rotated in the 2D image plane, and could be taken at different scales. Discuss how you could handle the case of translations and rotations only (no scale).