CS 6320, 3D Computer Vision
Spring 2012, Prof. Guido Gerig

Assignment 3: Stereo and 3D Reconstruction from Disparity

Out:	Monday Mar-19-2012
Due:	Monday Apr-02-2012, 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 7, 8.1 and 11.3
Slides to 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, deadline Monday Feb. 13 midnight).

Problem 1: Epipolar Geometry with 3 Cameras

We discussed epipolar geometry betwen two cameras, which limits a search for correspondence to an epipolar line. The Java demo (http://www.ai.sri.com/ luong/research/Meta3DViewer/EpipolarGeo.html) also illustrates the epipolar geometry for 3 camera views. Given the fact that for two cameras, the constraint for finding a corresponding point in the right camera for each point in the left camera is a line (i.e. still one degree of freedom), discuiss the situation with 3 cameras. Could it be that with 3 cameras, point to point correspondences are uniquely defined? Why or why not?

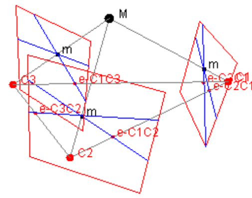

Figure 1: Epipolar geometry with 3 camera views.

Problem 2: Epiploar geometry and disparity forward translating camera

In our course lecture on image rectification (CS6320-CV-F2012-Rectification.pdf), we discussed the epipolar geometry and fundamental matrix of a forward translating camera.

- Given the definition of the epipolar plane as the plane defined by the two optical centers and a world point, explain why the geometry for a forward moving camera results in a set of radially oriented intersecting planes as shown in the following figure.
- Given geometric considerations of a forward moving camera, develop a framework to calculate depth of world points from disparity as observed in consecutive frames. Make use of the definition of disparity as presented for the case of a stereo camera setup.

Figure 2: Epipolar geometry with 3 camera views.

Problem 3: Point correspondence

- State possible contraints that can be used to reduce the ambiguities of finding matching point pairs (CS6320-CV-F2012-correspondence.pdf).
- Discuss occlusion problems and ordering constraints, and use illustrations to explain where these help and where these fail.
- Summarize the stereo matching method via finding the minimum cost path via dynamic programming. How does this method deal with occlusions and eventual violation of ordering constraints (e.g. images of trees in a forest)?

I. Practical Problems

Problem 4: 3D from stereo image pairs

This objective of this assignment is to take a stereo pair of images, to calculate a dense z-map (distance map), and to use a sparse set of corresponding points to calculate 3D world coordinates.

4a: Dense distance map from dense disparity

Write an algorithm to search for corresponding points along same scan lines (rectified image pairs). given optimal correspondence based on the match criteria (here we will used normalized cross correlation), we will calculate disparity and associated depth Z for each pixel. The result is a Z-buffer, or distance image.
Given the high computational expense, you are strongly encouraged to choose small size images for this experiment. Please note that images obtained by your camera can be easily subsampled using standard image display programs.
We will use a rectified image pair to simplify correspondence search along horizontal lines. You can create two images by simply shifting a camera horizontally along its image plane (e.g. placing a camera against a wall and shifting it horizontally along this wall). Remember to get a good estimate of the baseline of this shift.

- Write a Matlab program to scan two images in a TV-scan like fashion, i.e. a line-by-line, row-by-row algorithm where at each location, the nxn neighborhood of each pixel is available for calculations. We need an algorithm for simultaneous calculations on two images with different scan locations.
- For each nxn window in the left image, calculate normalized cross-correlation (see course slides) for each window in the right image. Remember that for each window in the left image, you need to search the whole horizontal line for the best matching window. Choose the best corresponding location as the location with the highest correlation. It is suggested to write a version for 3x3 and for 5x5 windows. Please note that the image border cannot be processed, i.e. you need to leave a frame of 1 or 2 pixel width, respectively.
- Calculate disparity for each corresponding pair of pixels. The disparity in pixel units needs to be converted into mm-units using the known pixel size from camera calibration.
- Using the disparity, calculate the depth Z at each pixel location.
- Display the stereo pair and the resulting depth image side by side and discuss your result.

4b: 3D Object geometry from triangulation

Using a similar setup as in the previous question, take a stereo picture of an object with simple geometry, e.g. a cube (see Fig. 3. Such objects are defined by a small set of landmarks (here 3D corners) and can be reconstructed by displaying a set of lines joining these key points.

- Define a small set of corresponding key locations by either manual definition of landmarks or a correlation-based image processing method (with selection of only the major key points).

- Calculate horizontal and vertical disparity in pixel units and mm-units, and from those the 3D point coordinates (X,Y,Z).
- Use Matlab to display the 3D points and edge lines for the reconstructed object (only those visible in your images). Choose display viewpoints different from the camera views to verify the quality of 3D reconstruction.
- Discuss your results.

Figure 3: Two pictures of a simple geometric object (here vertical stereo) and its 3D reconstruction with a wireframe illustration.

4c: Bonus Problem: Stereo Including Image Rectification

Optional bonus question, for those who have capacity to do more.
Use a stereo setup of your cameras where cameras are rotated against each other. The apply image rectification to project your images into a rectfied normative space where correspondences appear on horizontal lines. The proceed as described above for dense stereo and 3D reconstruction.