CS 6320, 3D Computer Vision
Spring 2013, Prof. Guido Gerig

Assignment 2: Stereo and 3D Reconstruction from Disparity

Out:	Mon Feb-11-2013
Due:	Mon Feb-25-2013, midnight (theoretical and practical parts, submission to CANVAS)
TA:	Anshul Joshi, MEB 3115
Office hours	TA: Mo/We 3pm to 5pm, G. Gerig Mo/We 3pm to 5pm

Required Readings: Computer Vision, Forsyth and Ponce, Stereopsis: New book Chapter 7, old book Chapter
Slides to chapters as provided on course web-page
(`http://www.sci.utah.edu/~gerig/CS6320-S2013/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.

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?

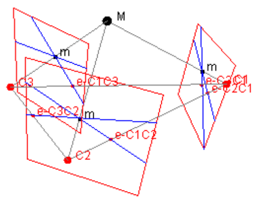

Figure 1: Epipolar geometry with 3 camera views.

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.

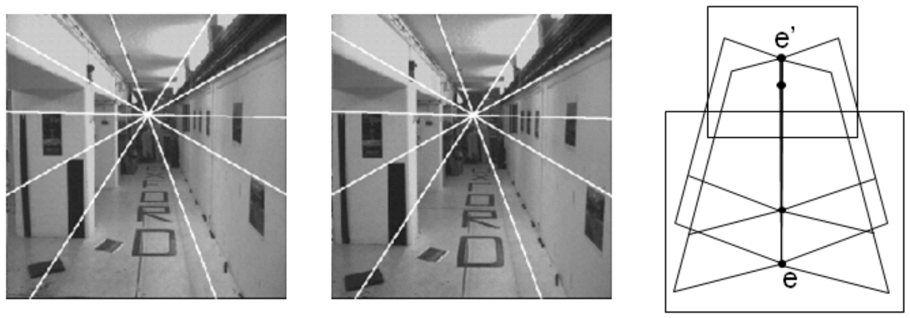

Figure 2: Epipolar geometry with 3 camera views.

Problem 3: Point correspondence

(Problem 7.8 new Textbook, slides stereo).

Show that the correlation function reaches its maximum value of 1 when the image brightness of the two windows are related by the affine transform $I^{'} = \lambda I + \mu$ for some constants λ and μ with $\lambda > 0$.

I. Practical Problems

Pleae note that you have to solve problem 4a but that you can then choose 4B or 4c.

Problem 4: 3D from stereo image pairs

This objective of this assignment is to take a stereo pair of images, and explore the epiploar gemeotry and the ability to recover depth from a pair of images. For this assignment, you will need two stereo pairs of your choice, one with arbitrary translation and rotation of the two camera views, and one where you simulate a translated camera by shifting one camera horizontally and by recording the distance of the baseline.

4a: Epipolar geometry from F-matrix

We discussed the essential and fundamental matrices and its properties to establish a relationship between corresponding points in the left and right cameras, and relationship of a points and their associated epipolar lines.
Following the examples discussed in our slides (CS6320-CV-F2013-chap11-multiple-views-part-II-animated.pptx) anc chapter 7.1 in our textbook, you will explore the following steps:

- Shoot a stereo-pair of a scene of your choice with your camera, where you take a left and a right picture where you translate and rotate the camera.
- Specify a set of corresponding pixel landmark pairs in the left and right camera views.
- Calculate the F-matrix using instructions as discussed in the slides.
- Choose a point in the left image and calculate the epipolar line in the right image. Display the point and the line as overlays in your images.
- Do the same for a point in the right image and epipolar line in the elft image.
- Calculate the position of the epipole of the left camera (see last two slides for instructions). Discuss if the calculated position seems reasonable.

4b: 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 precise estimate of the baseline length 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.
- A nice way to display depth images is the calculation and display of the slope, i.e. the image gradient. This can be obtained by calculating intensity differences of neighboring pixels (horizonally or verticall) and then display those as an image. Please note that differencing results in negative and positive values, so that double-type image matrices need to be shifted into the positive range of integer images to be displayed properly.

4c: 3D Object geometry via triangulation

Using a similar setup as in the previous question (cameras with parallel image planes), 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.

4d: Bonus Problem: Do both 4b and 4c

Optional bonus question, for those who have capacity to do more, by solving 4b and 4c.

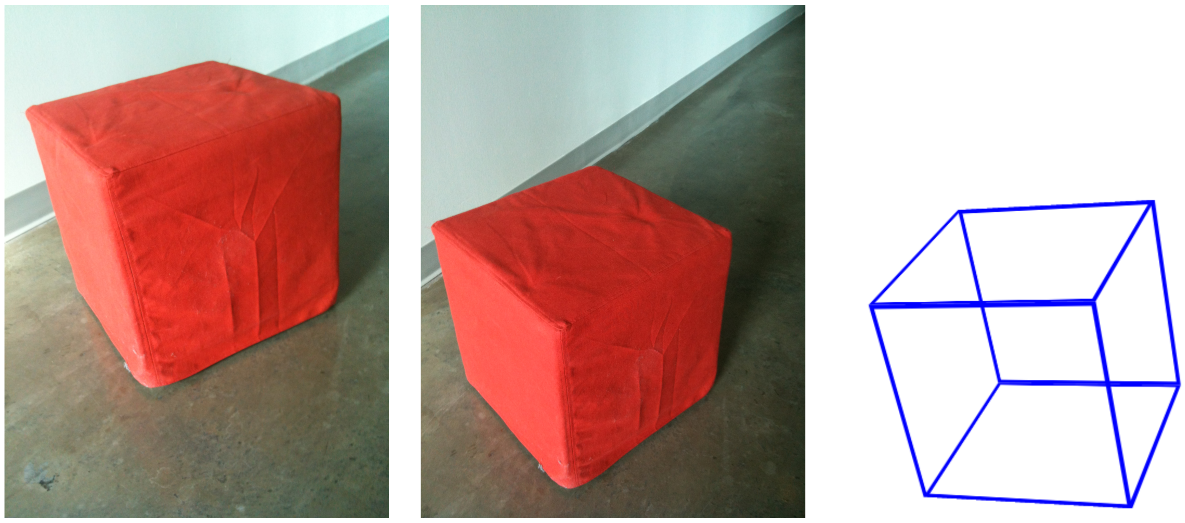

Figure 3: Two pictures of a simple geometric object (here vertical stereo) and its 3D reconstruction with a wireframe illustration.