UNIVERSITY OF TEXAS AT ARLINGTON
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

6367 **COMPUTER VISION** **SPRING 2020**

ASSIGNMENT 5 (75 POINTS)
ASSIGNED: 4/2/2020 DUE: 4/21/2020

This assignment constitutes 10% of the course grade. You must work on it individually and are required to submit a PDF report along with the MATLAB scripts described below. Your programs must not use computer vision functions provided by MATLAB unless otherwise specified.

Problem 1 (15 + 15 = 30 points)

(a) The coordinates of a set of points in the world frame are given in the file 'pts_world.mat'. Two pictures of these points were taken from a pair of stereo cameras. The corresponding image coordinates for the left and right cameras are given in the files 'pts_viewL.mat' and 'pts_viewR.mat' respectively.

Write a MATLAB function that computes the intrinsic parameters for each camera and the transformations from the world frame to the camera frame. The function is of the form `[Kl,Tl,Kr,Tr] = compute_stereo_calib(P,pl,pr)` where `Kl` and `Kr` are the 3×3 matrices for the intrinsic parameters of the left and right cameras, `Tl` and `Tr` are the homogeneous transformations from the world frame to the left and right camera frames, and `P` is the given set of points in the world frame.

Use the calibration technique described in the lecture notes which uses linear least squares to estimate the projection matrix of each camera, and then extracts the intrinsic and extrinsic parameters using the structure of the camera matrix. You can assume that the skew is zero for both cameras. Report all parameters in the report.

(b) Using only the corresponding points in the left and right images, perform 3D triangulation to compute an estimate for the location of the corresponding points in the left camera frame. Write a MATLAB function `P_hat = get_world_points(pl,pr)` where `pl` and `pr` are the sets of corresponding points in the left and right camera frames, and `P_hat` is the estimated locations of the world points in the left camera frame. Using the extrinsic parameters `[Kl,Tl,Kr,Tr]` computed in part (a), estimate the average, min, max and standard deviation of the error (defined as distance between the computed and ground truth points). Use this function within a script 'problem_1.m' that generates the required error statistics.

Submission Instructions: *For part (a) submit the MATLAB function* `compute_stereo_calib`. *For part (b) submit the MATLAB function* `get_world_points` *and the MATLAB script 'problem_1.m' along with any other files necessary to run your script. In addition, submit the output matrices for part (a) and all the error values computed for part (b) in the report.*

Problem 2 (25 points)

Figure 1 shows two images of a scene taken from the left camera (viewL.png) and right camera (viewR.png) of a stereo system. The images have been rectified and are free from radial distortion. Compare the results of following algorithms for computing correspondences and generating a disparity map:

- Sum of squared differences (SSD)
- Cross-correlation (CC)
- Normalized cross-correlation (NCC)

(a) Left camera image.

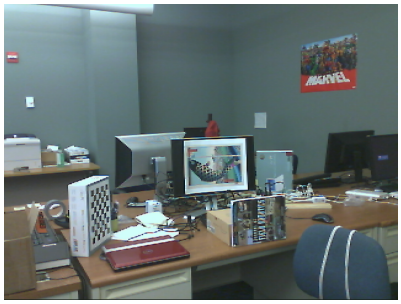
(b) Right camera image.

Figure 1: Images of a scene obtained using a stereo rig.

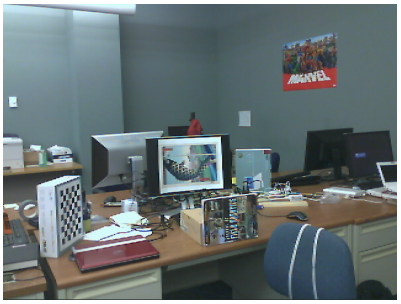
Note that some of these values must be minimized whereas others must be maximized. You will also need to experiment with window sizes and decide on a method to disambiguate in case of ties. Report these decisions as part of your answer. You can use MATLAB functions to convert color images to gray scale. The ground truth for the disparity (right - left) is provided in the file 'disparity.mat'. Write a MATLAB function `compute_corrs` that reads in the images and computes the disparity map using each of the three methods. For each method determine the mean, min, max and standard deviation of the error values using the ground truth. Also report the running times of each method implemented. Using the above criteria, report which method is the best.

Submission Instructions: *Submit the MATLAB function* `[disp] = compute_corrs(img_left, img_right,method)` *that takes as input the left and right images and returns a disparity map the same size as* `img_left`, *for the given* `method = 'SSD', 'CC','NCC'`. *Submit a MATLAB script 'problem_2.m' along with any other files necessary to run your code. The script must generate the disparity maps for each method in separate windows. Include these images along with the error statistics and running times in the report. Also include the decisions such as window size, threshold, etc., that you needed to make. You must implement your own functions to compute the SSD, CC, and NCC. Do not use built-in MATLAB functions such as* `normxcorr2`.

Problem 3 (Extra Credit) (15 + 10 = 25 points)



(a) Left camera image.



(b) Right camera image.

Figure 2: Images of a scene obtained using a stereo rig.

Images 'officeL.png' and 'officeR.png', shown in Figure 2, were taken from the stereo pair used for the previous problem. In this problem, you are required to first rectify the two images, then compute the

correspondence between them, and finally produce a depth map and a 3D point cloud for the scene.

(a) Using the camera parameters obtained in the Extra Credit, write a MATLAB function `[rectL,rectR] = rectify_images(imgL,imgR,Pl,Pr)` that takes in the left and right images (`imgL` and `imgR`) along with the 3×4 camera projection matrices `Pl` and `Pr`, and produces the rectified images `rectL` and `rectR`.

(b) Use your function `compute_corrs` in Problem 2 to determine the correspondences between the pair of rectified images using the better of the three methods (SSD, CC, NCC). Using these correspondences along with the camera parameters, determine the 3D locations for all points in the image. (You may choose to ignore those points which do not occur in the other image.) Display the results using a 2D depth map and a 3D point cloud in MATLAB. For this part, submit a script 'problem_4.m' that reads in the image and camera parameters, rectifies them, and generates the point clouds.

Submission Instructions: *In addition to all the MATLAB function and script files mentioned above, embed the rectified left and right images generated in part (a) in the report along with the 2D depth map and 3D point cloud generated in part (b).*

Problem 4 (5 + 5 + 10 = 20 points)

The file 'linefit.mat' contains three arrays:

- `xs`: The x-coordinates of 100 points on a line
- `n_y1`: The measurements of the y-coordinates, these measurements have zero-mean Gaussian noise
- `n_y2`: A second set of measurements that is similar to the `n_y1`, but contains outliers produced due to device anomalies

(a) Fit a line through the first set of measurements using a least squares approach. Do not use the `pinv` function of MATLAB. Instead, compute the pseudoinverse yourself (you can use `inv`). Report the line parameters.

(b) Repeat part (a) with the second set of measurements.

(c) Develop a RANSAC-based algorithm to get rid of the outliers. Explain how you chose the number of points and decided on points that are "close" to the estimated line. Report the line parameters.

Submission Instructions: *In the report, print the line parameters obtained for each part. Include a plot which contains all three lines (use different colors and styles). Clearly mark each line with the method type. Submit a MATLAB script 'problem_5.m' that performs each of the above mentioned operations, along with any other files necessary to run your code.*

Extra Credit (20 points)

The images 'checkerboard[1-12][l—r].png' in 'checkerboard.zip' were taken using a stereo system. Each square on the checkerboard is of dimensions (25.4mm $\times$ 25.4mm). Use the MATLAB camera calibration toolbox (http://www.vision.caltech.edu/bouguetj/calib_doc/) to extract the intrinsic parameters for each camera and the extrinsic parameters giving the transformation for points in the right camera frame to those in the left camera frame. Submit the obtained camera parameters in the report.