Problem set 2 (revised)

Assigned: 09/26/02
Due: 10/08/02 at 3pm

Problem 1 Weak Calibration (Matlab)

a. Given the two images `left1.bmp` and `right1.bmp`, write code using *imshow* and *ginput* to click on pairs of corresponding points. Select at least 15 pairs. Print these coordinates in your solution.

b. Implement the 8-point algorithm for weak calibration and estimate the fundamental matrix relating the image pair. Use only 8 of the pairs selected in part a. Show your estimated F matrix. Your function should have the following syntax:

```
function F = calibrate8(u1, v1, u2, v2)
```

where (u1, v1) are feature coordinates in the left image and (u2, v2) are the corresponding coordinates in the right image. You should submit your file `calibrate8-[last_name].m` and print a copy of your code. Include the calculated F matrix with your solution, normalized so F(3,3) = 1000.

c. Create plots similar to Figure 10.4 in the text for your image pair: e.g., for each feature point p in the left image plot the line $p^T F$ in the right image, as well as the corresponding point location p'. Plot the lines Fp' in the left image as well as the points p. Use all the points from part a.

d. Implement a point to line distance function in Matlab, and find the distance between the estimated epipolar line and corresponding point for all selected points (from part a). Calculate the average distance.

e. Implement an improved calibration function using the least-squares solution, numerical preconditioning, and rank-2 constraints proposed by Hartley (1995). Use matlab's SVD function to implement the rank-2 constraint. Estimate an improved F matrix using all of the points from part a. Your function should have the following syntax:

```
function F = calibrateHartley(u1, v1, u2, v2)
```

where (u1, v1) are feature coordinates in the left image and (u2, v2) are the corresponding coordinates in the right image. You should submit your file `calibrateHartley -[last_name].m` and print a copy of your code. Include the calculated F matrix with your solution, normalized so F(3,3) = 1000.

f. Repeat the plot from c) and error estimate from d) using the improved estimate of the F matrix.

Problem 2 Stereo Correlation (Matlab)

a. Implement a 1-d algorithm to search for corresponding points using normalized correlation. Your function should have the following syntax:

```
function C = stereoCorr(Ileft, Iright, ur, vr,
                        windowSize, nbDisp, offsetDisp)
```

where

- `Ileft` and `Iright` are the input images,
- `ur` and `vr` are the coordinates of the point in the right image,
- `windowSize` is the size of the correlation window,
- `nbDisp` is the number of disparities to try,
- `offsetDisp` is the value of the first disparity (by default 0),
- `C` is a vector filled with the normalized correlation score of each disparity.

Test your algorithm by searching for the point (u,v) in `left2.bmp` corresponding to the point (62,124) in `right2.bmp`. Images `left2.bmp` and `right2.bmp` are already rectified, so you can simply search on a horizontal scanline. Search over 64 disparities and set the offset to 0.
Plot the normalized correlation scores for these window sizes: 5, 7, 9 and 11. What is the maximum score for each window size? Which window size is optimal? Why?
You should submit your file `stereoCorr-[last_name].m` and print a copy of your code.

b. Search now using the maximum score found in (a) as the starting image coordinate for image `left2.bmp`, and see what point your algorithm matches in `right2.bmp`. Is it (62,124)? This procedure is called a "left-right check".
Hint: use the same function but swap left and right images and set the offset to -64.

c. Describe the common conditions when you might expect a left-right check to fail.

d. Modify your algorithm to use SSD matching instead of normalized correlation. Your function should have the following syntax:

```
function C = stereoSSD(Ileft, Iright, ur, vr,
                       windowSize, nbDisp, offsetDisp)
```

Test your new algorithm by searching for the point (u,v) in `left2.bmp` corresponding to the point (62,124) in `right2.bmp`. Plot the SSD error for different window sizes (5, 7, 9 and 11). Compare with the normalized correlation. Are their extrema at the same location?
You should submit your file `stereoSSD-[last_name].m` and print a copy of your code.

Problem 3

Exercise 11.6 from *Computer Vision: A Modern Approach* (Forsyth and Ponce)

Problem 4

Suppose we represent two lights by the three-dimensional vectors that represent each light's cone photopigment responses, **a** and **b**. The vector difference between the two representations of the two lights is **d** = **a** – **b**. Finally, consider two lights **m** and **n** that also differ by this same vector, **d** = **m** – **n**.

a. Suppose that we double the intensity of **a** and **b**. What happens to the vector representing each of the lights? What happens to the vector representing the difference between the scaled lights?

b. Suppose that we express the coordinates of these lights in another color space obtained by applying a linear transformation, *T*. What will be the vector difference between **a** and **b** in the new color space? What will be the vector difference between **m** and **n** in the new color space?

c. Do you think the two lights represented by **a** and **b** will be as discriminable as the lights **m** and **n**? Why or why not? Do you know of any experimental data to support your claim? What relevant data might be collected?

Problem 5

Exercise 6.5 from *Computer Vision: A Modern Approach* (Forsyth and Ponce)

Problem 6 Metamers

a. Suppose you establish a metameric match. Then you put on a pair of sunglasses. Will the metameric match be preserved? Describe why or why not.
b. As we age, the wavelength transmissivity of our cornea and lens changes. What effect will this have on the color-matching functions?