CSE 152 Assignment 3
Spring 2006
Due Tuesday, May 30, in class

Instructions :

- Follow all naming conventions suggested in the questions.
- Please comment all your Matlab code adequately.
- Try to minimize use of loops in Matlab!
- Each question is worth 25 points.

1 Epipolar Geometry

In this problem, we will verify the claim in class that for any point in one image of a stereo pair, the corresponding point in the other image lies on the epipolar line. The final output for this problem should be a Matlab function, contained in a file *testepipolar.m*. Here is what the function should look like:

```
function [F] = testepipolar(xl, xr, l, r)

  % Compute the fundamental matrix
  F = computeF(xl, xr);

  % Compute the epipolar lines in the left image
  leftlines = ??
  % Compute the epipolar lines in the right image
  rightlines = ??

  % Draw epipolar lines on the left image
  figure; imshow(l,[]); axis image; hold on;
  drawLine(leftlines(:,1), l, 'yellow');
  drawLine(leftlines(:,3), l, 'cyan');
  drawLine(leftlines(:,5), l, 'green');
  drawLine(leftlines(:,7), l, 'magenta');
  drawLine(leftlines(:,9), l, 'red');
  hold off;

  % Draw epipolar lines on the right image
  figure; imshow(r,[]); axis image; hold on;
```

```
drawLine(rightlines(:,1), r, 'yellow');
drawLine(rightlines(:,3), r, 'cyan');
drawLine(rightlines(:,5), r, 'green');
drawLine(rightlines(:,7), r, 'magenta');
drawLine(rightlines(:,9), r, 'red');
```

The overall code for this problem must be contained in a file called *script.m* which might look like:

```
% Read in the input images
l = imread(left);
r = imread(right);

% Convert to double
l = double(l);
r = double(r);

% Pick the corresponding corners
xl = clickPoints(l, 10, 1, 1);
xr = clickPoints(r, 10, 1, 1);

F = testepipolar(xl, xr, l, r);
```

We will now have a look at the individual components of the above code.

(a) Use the provided function in *clickPoints.m* to click 10 points from the image *left* and the corresponding 10 points in the image *right*. For the blocks data, *left* is *blocks1.gif* while for the desk data, *left* is *desk1.gif*. The other image of the stereo pair is *right*. Choose the 10 points as shown in Figure 1 for the blocks and as shown in Figure 2 for the desk data. Be very careful to click the corners as accurately as you can.

(b) Write a function, call it *computeF*, to compute the fundamental matrix for an image pair. The function takes as input two $2 \times n$ arrays, each of which contains the points clicked in part **(a)**. n is the number of points clicked in each image. Output is the 3×3 fundamental matrix.

(c) Using the fundamental matrix, compute the epipolar line in each image that corresponds to points in the other image. Store the homogeneous representations of the epipolar lines in the left image (which correspond to corners in the right image) in the 3×10 matrix *leftlines* and those for the right image in *rightlines*.

(d) Use the provided function in *drawLine.m* to draw the epipolar line in each image for the points 1, 3, 5, 7 and 9.

(e) Compute the epipoles in the left image and the right image. Call them *el* and *er*, respectively.

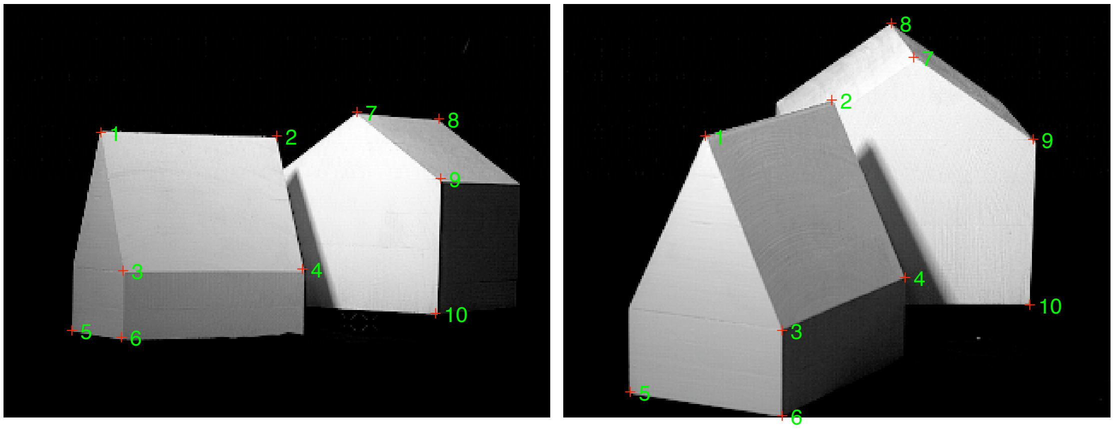

Figure 1: The points to be clicked for the blocks data.

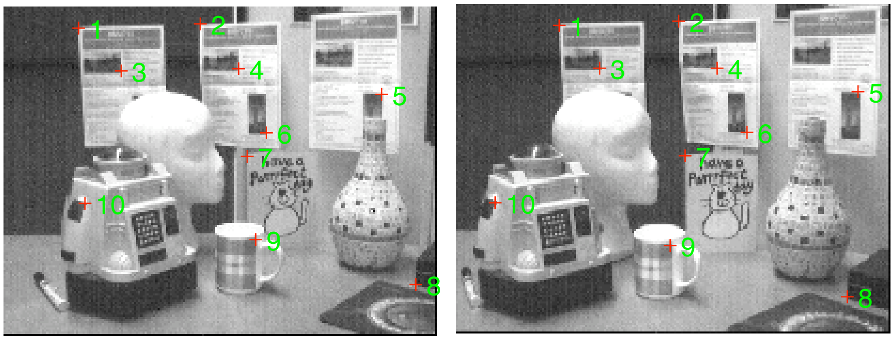

Figure 2: The points to be clicked for the desk data.

Turn-in :

1. The Matlab files *script.m*, *testepipolar.m* and *computeF.m*.
2. *.mat* files called *datablocks.mat* and *datadesk.mat* which contain the saved variables `l`, `r`, `xl`, `xr`, `F`, `leftlines`, `rightlines`, `el` and `er` that correspond to each data set.
3. *.fig* files that contain saved Matlab images for the clicked points. Call them *lblock-click.fig* and *rblock-click.fig* for the blocks data and *ldesk-click.fig* and *rdesk-click.fig* for the desk data.
4. *.fig* files that contain saved Matlab images for the epipolar lines. Call them *lblock-lines.fig* and *rblock-lines.fig* for the blocks data and *ldesk-lines.fig* and *rdesk-lines.fig* for the desk data.
5. A write-up with details of your implementation. It must contain the following:
   (a) A description, in words, of the method you use to compute the fundamental matrix.
   (b) The procedure to compute the epipolar lines in the left and right images.

(c) The procedure to compute the epipoles in the left and right images.

(d) The values of the fundamental matrix for the blocks data and the desk data (use greater precision than the default in Matlab, if necessary, to report these).

(e) The location of the epipoles on the 2-D image plane (not the homogeneous points) for the blocks data and the desk data.

(f) Your comments (with reason) on the relationship between the relative orientation of the epipolar lines and the magnitude of the epipole location on the 2-D image plane.

2 Intensity-based Dense Stereo

In this problem, we will use the method of matching intensity windows discussed in class to compute disparity between a pair of stereo images. The left and right images to be used are *tsukuba-l.png* and *tsukuba-r.png*, respectively. Note that the images are rectified, so that the point in the right image corresponding to one in the left image lies at the same horizontal level (that is, in the same row).

We will use two different metrics discussed in class: Sum of Squared Differences and Normalized Correlation. For each, compute disparity maps using three different window sizes 3×3, 5×5 and 9×9. To get the disparity value at each point in the left image, search over a range of disparities and compare the windows using the two different metrics. Restrict the possible disparity values to be integers and choose reasonable values for the minimum (possibly negative) and maximum disparity from an examination of the images and your results.

Note : A naive implementation in Matlab would require four nested for-loops, which can be very slow. Try to avoid looping wherever possible, take inspiration from filtering and Assignment 2. But if you cannot get this to execute in Matlab within a reasonable amount of time, write the program in C. In that case, it is up to you to figure out reading and writing of images in C.

Turn-in :

1. Matlab *.fig* files that contain disparity maps for the three window sizes, using both metrics. Name your figures as *disparity_w_method.fig*, where *w* is the window size and must be 3, 5 or 9 and *method* is either *ssd* or *nc*, which stand for Sum of Squared Differences and Normalized Correlation, respectively. There must be 6 disparity maps in all.

2. Your implementation code (Matlab or C), call it *stereo.m* or *stereo.c*. It must take as input (in this order) - the left image, the right image, a window size and a flag to choose the metric (set the flag to 1 for sum of squared differences and 2 for normalized correlation). Output must be the disparity map. State the time your code takes to execute for each combination of metric and window size.

3. Optionally, any comments you might have can be included in the write-up.