cs512 Homework 4: Report

Arun Rajagopalan
Graduate student, A20360689
Department of Computer Science
Illinois Institute of Technology

December 1, 2015

Abstract:

The problem or the task is to derive the fundamental matrix that defines a given stereo image pair and to compute and display epipoles of the images and the epipolar lines corresponding to the user selected query point, using our own functions, alongside using other OpenCV library functions as necessary. The aim is to understand the basic concepts of epipolar geometry and implement them in software. Many of the crucial characteristics for the processes and algorithms involved and their effects on image processing are also to be studied, documented and analyzed.

1 Problem statement:

To find the fundamental matrix of the taken image pair, compute its epipoles and draw epipolar lines as necessary. If the file names (with path, if the images are in some other folder) are specified as arguments while starting the program from command line, the files should be read and used. For calculating fundamental matrix and epipolar lines, users should be able to select points on the images using the mouse. Also by using keyboard keys, users should be able to perform several operations related to epipolar geometry on the image pair. Solving this problem, helps to understand the concepts of fundamental matrices, epipoles and epipolar lines and familiarizes us with the software implementation details of the same.

2 Proposed solution:

The solution starts with a check on the number of parameters being passed to the program while starting. Based on parameter count, either images are read from disk or an exception is thrown in case of issues in the arguments. The solution listens to the keyboard always and if a valid input key is pressed (say 'p' or 'F'), appropriate image processing operation will be done. When the user presses 'p' or 'P', the program also starts listening to mouse events or clicks and continues to do so, until the session exit key 'X' is pressed. Thus by clicking on images, user inputs the corresponding points in the left and right images to the program for calculating fundamental matrix. While providing for fundamental matrix and epipole calculation, the program would also allow users to click a point on an image and get the corresponding epipolar line drawn in the other image. The program would never terminate unless the user intends and instructs the program to do so. Deeper implementation details follow in the next paragraph

3 Implementation details:

To implement the solution, python 2.7.3 and OpenCV 3.0.0 was used. Main packages used include 'cv2' (for core image processing methods), 'numpy' (for image storing and array

operation methods) and 'sys' (to get the arguments from command line). Source code (hw4.py) is placed in the 'code' folder under the parent directory 'arun_rajagopalan_as_4'. Images used to test the code are placed in 'data' folder under the same parent directory.

At first, as discussed above, the program checks for the number of arguments passed on startup. If it is three, images on the given paths are read and displayed. In case of invalid path or invalid number of arguments, an error message is thrown and program exits gracefully. A screen shot of the actual program behavior on startup is shown in Illustration 1.

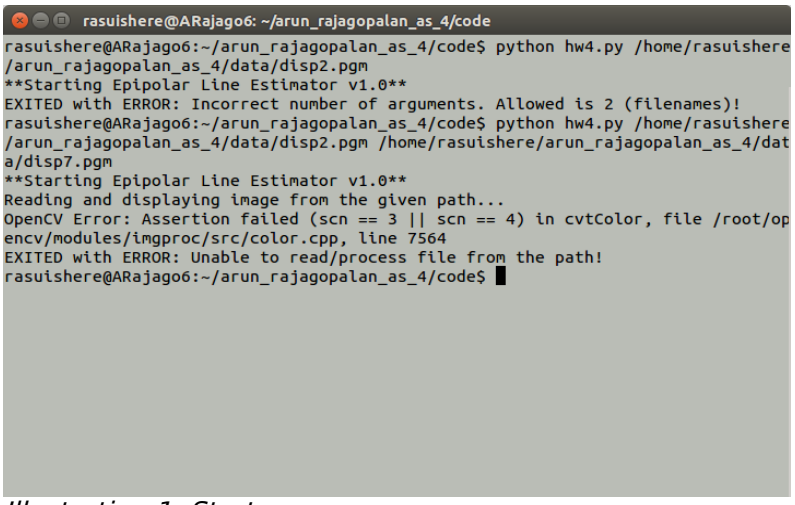

Illustration 1: Startup cases

After the images are obtained, the program starts listening for keyboard inputs. On valid keyboard entries, a global flag variable for operation, 'op_flag' is set equal to the respective input. As long as the flag is set to a respective key, say 'p', which is to enable mouse click recording, the program would continue tracking and saving the co-ordinates of clicked points in both left and right images alternatively, for future use, till the reception of another valid keyboard interrupt. **Pressing 'h' with the program window active, will show the valid keyboard entries with their functions. User can end the program only by pressing ESC key.** For every key press, terminal prints a corresponding message, so that it would be easy to keep track of what is happening currently and what has happened since the start of the program. Look of the GUI on startup with 'disp2.pgm' and 'disp6.pgm' available in the 'data' folder, the 'h' key output and the terminal state are as below in Illustrations 2, 3 and 4 respectively. Details about the other valid keyboard entries are as follows.

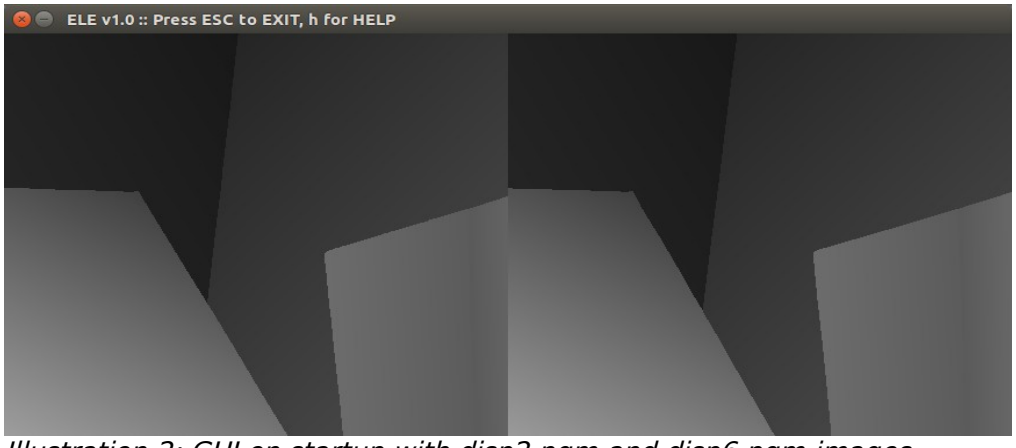

Illustration 2: GUI on startup with disp2.pgm and disp6.pgm images

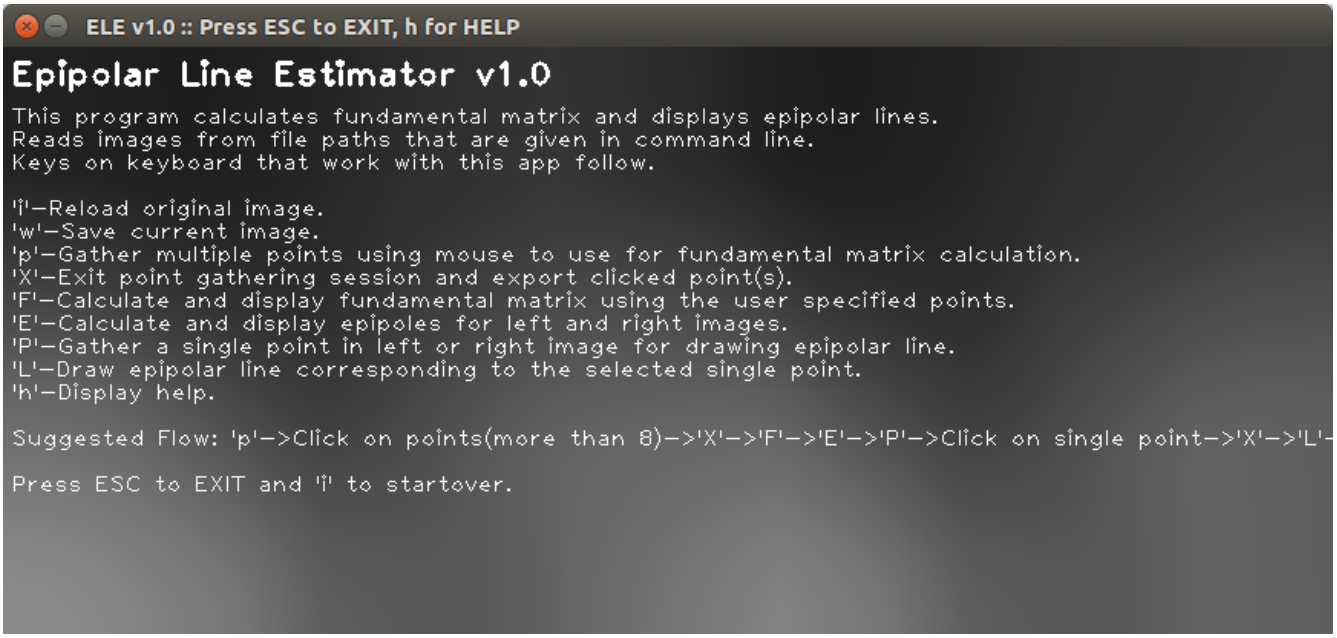

Illustration 3: Output of 'h' key

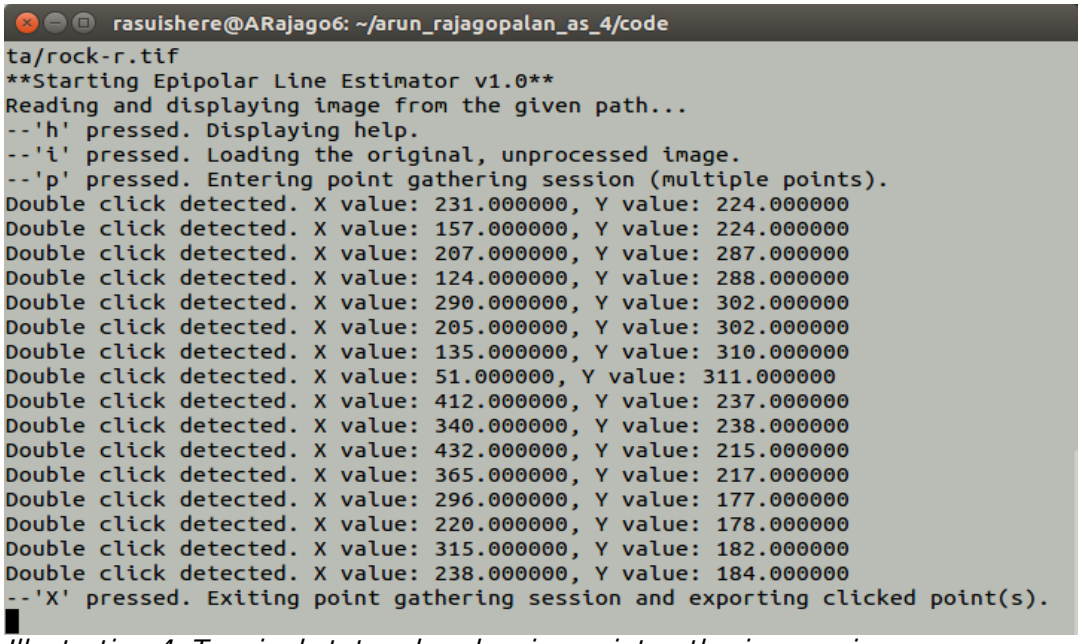

Illustration 4: Terminal state, also showing point gathering session

Mouse usage and the program control flow:

One of the main requirements of the homework was to allow user to specify points for fundamental matrix and epipolar line calculations. **When the user presses 'p'(lowercase), the program enters multiple point gathering mode and keeps on saving points until X is pressed.** 'X' key is used to disable the mouse listening function and export the saved points to global arrays. There are some important constraints to note here. **One is that at least 8 pairs of points must be chosen for the successful calculation of fundamental matrix.**

Second is that initially the click must be made on left image, then on the corresponding point in right image and then on some other point in the left and so on. This is because the two input images are concatenated and hence the program subtracts a constant value from every second click to get right co-ordinates. If the order is changed, the program would end up subtracting value from wrong points and we would be left with incorrect array of co-ordinates. **After the corresponding points are exported, user can press 'F' and then 'E' to calculate and display fundamental matrix and epipoles.**

'P'(uppercase) should be used only after the usage of 'F' and 'E' keys. When 'P' is pressed, the program can save only one point (the latest one) for which the epipolar line would be drawn in the other image. **After selecting a point, 'X' should be pressed to export the selected point.** At the end of this single point selection, **user can then press 'L' to draw the corresponding epipolar line.**

Illustration 5: GUI during 'p' and 'P' keys

Fundamental matrix calculation:

For this purpose, we use the 8-point algorithm and hence prior to starting the calculation, this routine checks whether the input point arrays have atleast 8 points. If not, the program halts. If the arrays have more than 8 points, then these points are normalized.

Normalization of image points is done to ensure consistent and correct derivation of fundamental matrices and epipoles. It is done by subtracting the mean of all co-ordinates from respective co-ordinate of every point and dividing them by the variance of all co-ordinates in the array. This operation can be written as a transformation matrix Mp which when multiplied with an original homogeneous point would give out the normalized homogeneous point.

$$Mp = \begin{bmatrix} 1/\sigma x & 0 & 0 \\ 0 & 1/\sigma y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\mu x \\ 0 & 1 & -\mu y \\ 0 & 0 & 1 \end{bmatrix}$$

Using the normalized point arrays, the A matrix is calculated as follows. Note that this A matrix has a dimension of N X 9 where N is the number of corresponding pairs. For every point in the input arrays, we do the following.

```
A_mat[i][0] = nr_array[i][0]*nl_array[i][0]
A_mat[i][1] = nr_array[i][0]*nl_array[i][1]
A_mat[i][2] = nr_array[i][0]
A_mat[i][3] = nr_array[i][1]*nl_array[i][0]
A_mat[i][4] = nr_array[i][1]*nl_array[i][1]
A_mat[i][5] = nr_array[i][1]
A_mat[i][6] = nl_array[i][0]
A_mat[i][7] = nl_array[i][1]
A_mat[i][8] = 1
```

Here A_mat is the A matrix, nr_array is the set of normalized right image points and nl_array is the set of corresponding left image points. From the singular value decomposition (SVD) of this A matrix, we get u, d and transposed v matrices. We take the last column of v as the initial fundamental matrix.

With this initial fundamental matrix, we apply SVD again to get u, d and v transpose again. Here, we make the smallest value of d as zero and reconstruct using u, new d and v transpose to get the fundamental matrix of normalized points(F').

To get the fundamental matrix of original points(F), we multiply the fundamental matrix of normalized points with the transformation matrix of left image(Mp') which is then multiplied with the transpose of transformation matrix of right image(Mp).

$$F = Mp^{T} * F' * Mp'$$

Calculation of epipoles:

First step is to perform the SVD of the fundamental matrix and the columns corresponding to the zero singular values of u and v give the right and left epipoles respectively.

Drawing the epipolar lines:

With the fundamental matrix, given a point in left image, its right epipolar line co-efficients can be found as below.

```
Right epipolar_coefficients = np.dot(fundamental_matrix,
                              (np.asarray(given_point[:3])).reshape(3,1))
```

Similarly, given a point in right image, its left epipolar line co-efficients can be found by the following.

```
Left epipolar_coefficients = np.dot(fundamental_matrix,
                             (np.asarray(given_point[:3])).reshape(3,1))
```

With the co-efficients the slope of the line can be found (arctan(b/a)) and if the line is more horizontal we use the below Y equation to define the line or if the line is more vertical, we use the below X equation to define the line.

$$Y= -c-ax/b$$
$$X=-c-bx/a$$

where the co-efficients are of the form (a,b,c) and the points are of the form (x,y,1)

Thus the fundamental matrix and epipoles are found and the epipolar lines are drawn.

4 Results and discussion

The output screens of the program for the input images 'rock-l.tif' and 'rock-r.tif', which are present in the 'data' folder are as follows. The results are as expected, but for the epipolar lines, the result could be better.

```
--'X' pressed. Exiting point gathering session and exporting clicked point(s).
--'F' pressed. Calculating fundamental matrix using the user specified points.
The selected points in the left image are as below.
[[ 231.  224.]
 [ 207.  287.]
 [ 290.  302.]
 [ 135.  310.]
 [ 412.  237.]
 [ 432.  215.]
 [ 296.  177.]
 [ 315.  182.]]
The selected points in the right image are as below.
[[ 157.  224.]
 [ 124.  288.]
 [ 205.  302.]
 [  51.  311.]
 [ 340.  238.]
 [ 365.  217.]
 [ 220.  178.]
 [ 238.  184.]]
The estimated fundamental matrix for the given points is as below.
[[  7.25774584e-10  -3.09524312e-08   8.70973745e-06]
 [  3.35900535e-08  -2.10088458e-09  -9.52772718e-06]
 [ -1.00140665e-05   7.33738768e-06   9.04556317e-04]]
--'E' pressed. Calculating epipoles for left and right images.
The epipole of the left image is as below.
[[ 301.68927961]
 [ 288.46509044]]
The epipole of the right image is as below.
[[ 217.1369361 ]
 [ 293.43430668]]
```

Illustration 6: Outputs of 'F' and 'E' keys

```
--'P' pressed. Entering point gathering session (single point).
Double click detected. X value: 228.000000, Y value: 223.000000
--'X' pressed. Exiting point gathering session and exporting clicked point(s).
--'L' pressed. Epipolar line corresponding to the selected point will be drawn.
The selected point in the left image is as below.
[ 228.  223.]
The co-efficient of right epipolar line for the selected left point is as below.
[[  1.97282190e-06]
 [ -2.33769225e-06]
 [  2.57586600e-04]]
```

Illustration 7: Outputs of 'P' and 'L' keys

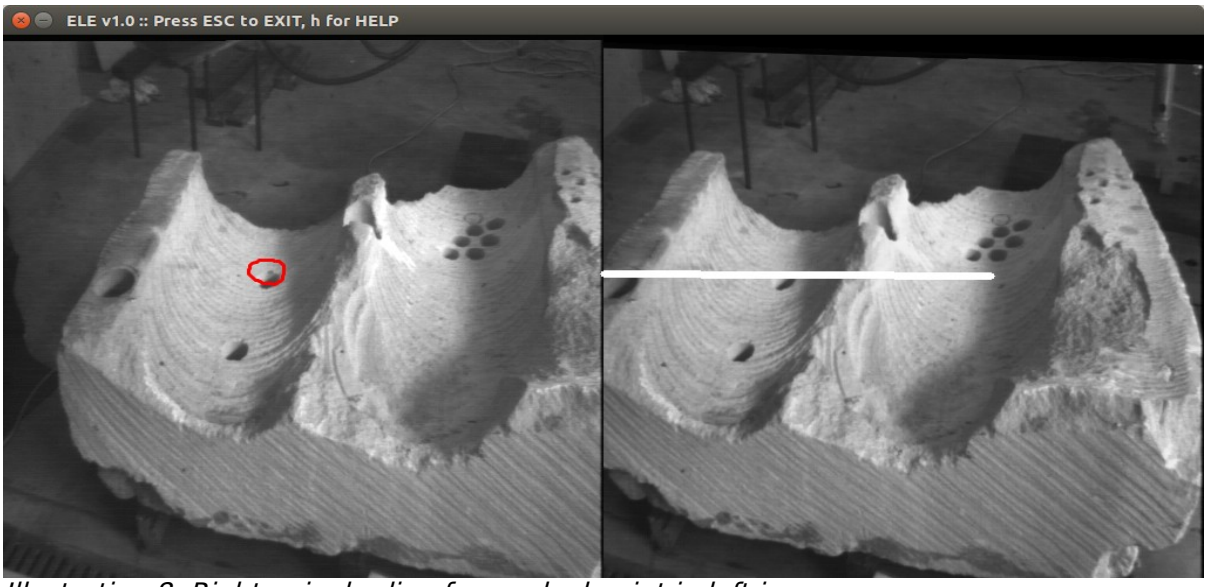

Illustration 8: Right epipolar line for marked point in left image

Illustration 9: Left epipolar line for marked point in right

5 References:

- https://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_tutorials.html
- http://docs.opencv.org/
- https://docs.python.org/3/tutorial/
- http://stackoverflow.com/questions/tagged/python-2.7
- http://docs.opencv.org/master/da/de9/tutorial_py_epipolar_geometry.html#gsc.tab=0