# COMPUTER VISION
# Assignment #5

By

Mostafa Mahmoud

65 Mohamed Ali Mokhtar

Mohamed Osama

51 Hassan

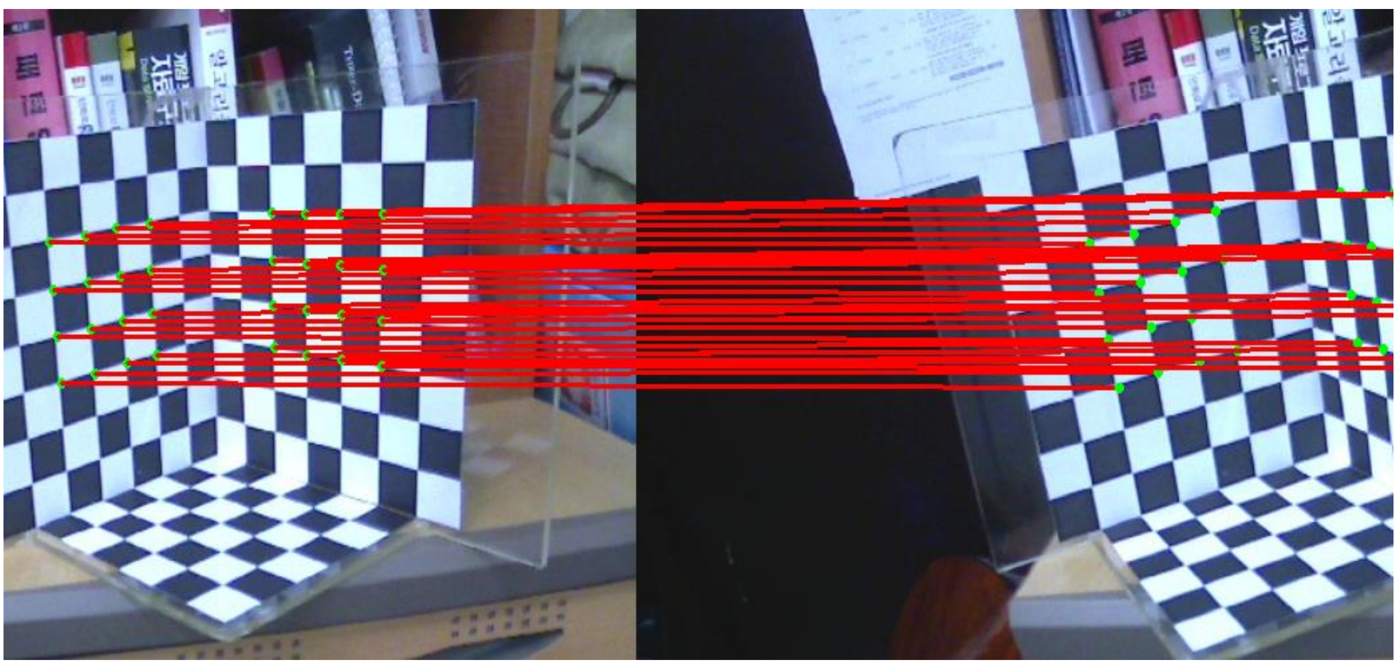
Azmy

# 1. Detecting Image Correspondence

In this section we used the given code with the problem statement, so the given code in this part is only reading 5 files, described as follow:

1- Left camera image: which is the image taken from the camera
2- Right camera image
3- Left camera matrix
4- Right camera matrix
5- Matches

Then we have to draw the corresponding matched points between the two images by drawing a line connecting every two matched points in the two images like in the following figure.



# 2. Fundamental Matrix Estimation

As we have given parameters about the camera calibration and the matched points between the two images obtained from the two cameras, so we have three sample test images with their own matched points data in the two images and the camera matrices.

The camera matrices have a definite shape and definite components and illustrated as follow.

Since $\mathcal{P} = \mathcal{K}[\mathcal{R}|t\ ]$ and $\mathcal{K} = \begin{bmatrix} m_x f & mxfs & m_x \beta_x \\ 0 & m_y f & m_y \beta_y \\ 0 & 0 & 1 \end{bmatrix}$

And $\mathcal{R}$, $t$ are the rotation matrix and the translation matrix.

So next we will show how to find the Fundamental Matrix using the 8-Points algorithm.

## 2.1 Eight-Point Algorithm

The Eight-Point Algorithm is an algorithm used to estimate the essential matrix or the fundamental matrix related to a stereo camera pair from a set of corresponding image points.

So we have $x = (u, v, 1)$ , $x' = (u', v', 1)$ the two point on the images.

Then let us say the $\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{23} & f_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$

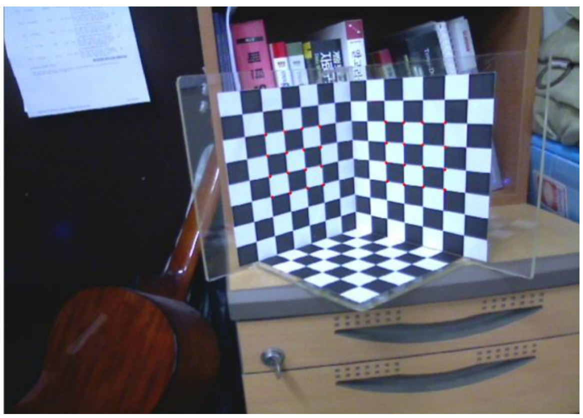
$$\underbrace{\begin{bmatrix} u'u & u'v & u' & v'u & v'v & v' & u & v & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u'_n u_n & u'_n v_n & u'_n & v'_n u_n & v'_n v_n & v'_n & u_n & v_n & 1 \end{bmatrix}}_{A} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Then trying to minimize $||Ax||^2$ which is equivalent to minimizing $\sum_{i=1}^{N}(X_i'^T F X_i)^2$

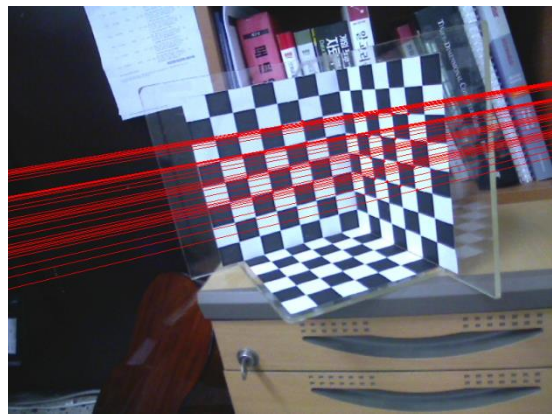
So "OpenCV" provides us the function "cv2.findFundamentalMat()" that takes a flag to make it find to us the fundamental matrix using the 8-Points algorithm from the corresponding points in two images.

## 2.2 Visualization

After fitting the fundamental matrix to the matching points, we can now visualize the results by transforming points from the first image to get epipolar lines in the second image.



(a) Select point from First Image



(b) Corresponding epipolar lines in the second image

# 3. Depth Estimation From Triangulation

## 3.1 Linear Triangulation

The linear triangulation method is the most common method for triangulation.

We write in homogeneous coordinates x = w (u; v; 1), where (u; v) are the observed point coordinates in (x, y) directions and w is an unknown scale factor.
So in our code to solve such an equation we first build a Matrix A in shape of 3x4, and for homogenous equation system, but for Linear-LS method we can change the equation into the form:

$$AX = B \; as \; X = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T \text{,z}$$

As we have input projection matrices of the two cameras so now we construct the Matrix A, by projection of point p by Matrix P and we have the formula of Matrix A in the attached Code.

But we will give short notes how did we got that formula.

Let Matrix A is 4x3 Matrix , and Matrix B is 3x1

From our knowledge that $\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} P[0,0] & P[0,1] & P[0,2] & P[0,3] \\ P[1,0] & P[1,1] & P[1,2] & P[1,3] \\ P[2,0] & P[2,1] & P[2,2] & P[2,3] \end{bmatrix} \begin{bmatrix} X \\ Y \\ W \end{bmatrix}$

After applying matrix multiplication,

$$\begin{aligned} wx &= P[0,0]x' + P[0,1]y' + P[0,2]z' + P[0,3] \\ wy &= P[1,0]x' + P[1,1]y' + P[1,2]z' + P[1,3] \\ w &= P[2,0]x' + P[2,1]y' + P[2,2]z' + P[2,3] \end{aligned}$$

After getting an equations of (x',y',z') as the those parameters are the coordinates of the point in the 3d

For first row:

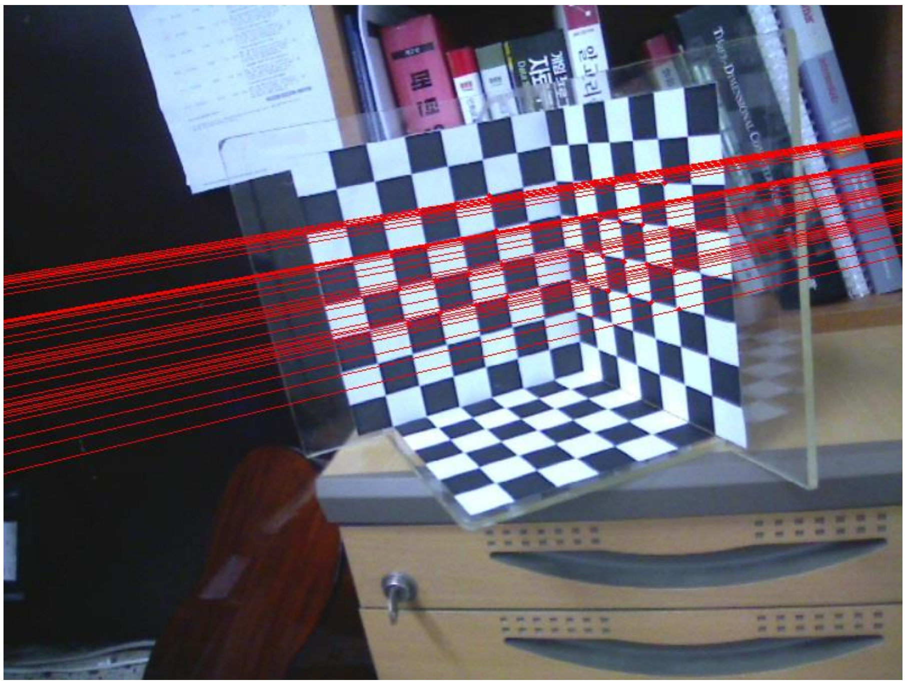
$$x'(P[2,0]x - P[0,0]) + y'(P[2,1]x - P[0,1]) + z'(P[2,2]x - P[0,2]) = -(P[2,3]x - P[0,3])$$

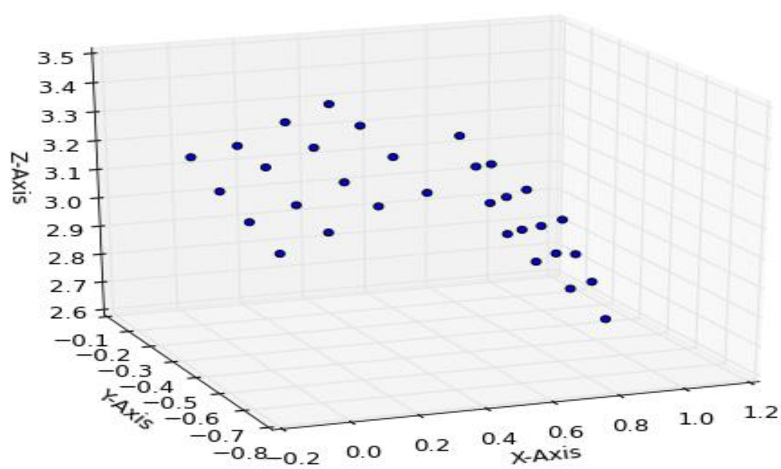
A[0,0] A[0,1] A[0,2] B[0]

And then same is for the remind A[0.1],B[1] till A[3,2],B[3] Then we use the method "cv2.solve(A, B, X, cv2.DECOMP_SVD)" to solve the equation using the flag of decomp_SVD.

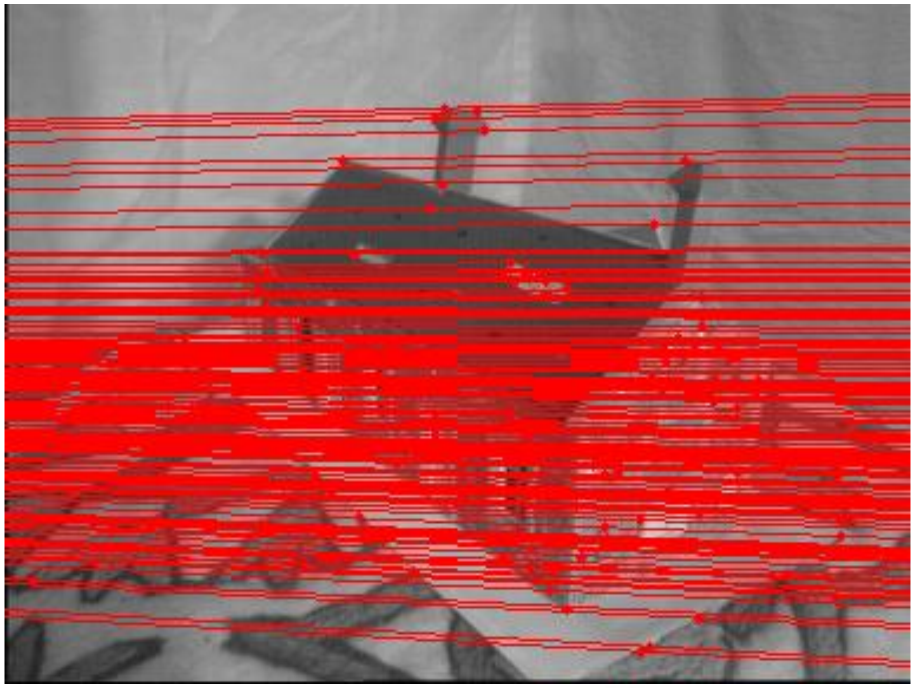
# 4. Sample Runs and outputs

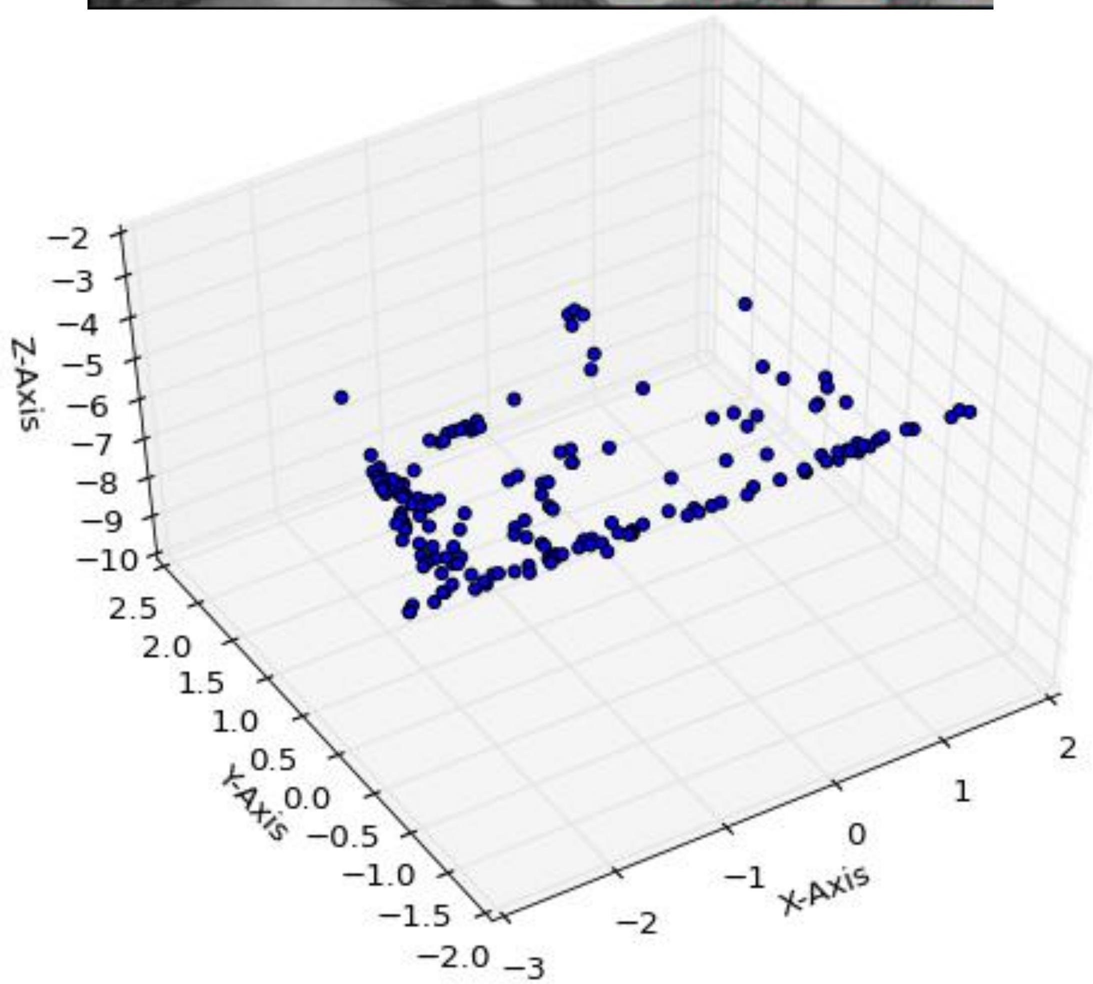
Here the Output after running our code on the input checkers photos.





Here the Output after running our code on the input House photos.





Here the Output after running our code on the input Library photos.