Punjab University College of Information and Technology

**Computer Vision project:**

**Image based 3D reconstruction**

**Submitted to: Sir Idrees.**

**Dated: 27th June, 2016.**

**Submitted By:**

**Zaid Annas <BCSF12A527>**

**Muhammad Habib Ali <BCSF12A520>**

**Shahbaz Nawaz <BCSF12A548>**

**Ali Mamoon <BCSF12A531>**

**Introduction**

Our project was to construct 3D model of an object from images. There are several techniques used for this purpose. Some of them are listed below.

* Stereo Vision
* Ray tracing
* Structure from motion

Each technique has its own significance and difference method to construct a 3D model. Stereo vision and Ray tracing method require special hardware. Whereas Structure from Motion uses multiple images from a single camera to construct a 3D model. So we will be using this technique because it is less complex and do not require special hardware.

**Multi View 3D reconstruction**

Multi View 3D reconstruction uses multiple images of an object from different angles to generate a 3D model of that object. Due to the loss of one dimension in the projection process, the estimation of the true 3D geometry is difficult and a so called ill-posed problem, because usually infinitely many different 3D surfaces may produce the same set of images.

The goal of multiview 3D reconstruction is to infer geometrical structure of a scene captured by a collection of images. Usually the camera position and internal parameters are assumed to be known or they can be estimated from the set of images. By using multiple images, 3D information can be (partially) recovered by solving a pixel-wise correspondence problem. Since automatic correspondence estimation is usually ambiguous and incomplete, further knowledge (prior knowledge) about the object is necessary.

There are 2 different types of techniques used is this method

1. Using calibrated camera images.
2. Using uncelebrated images.

**Original images**

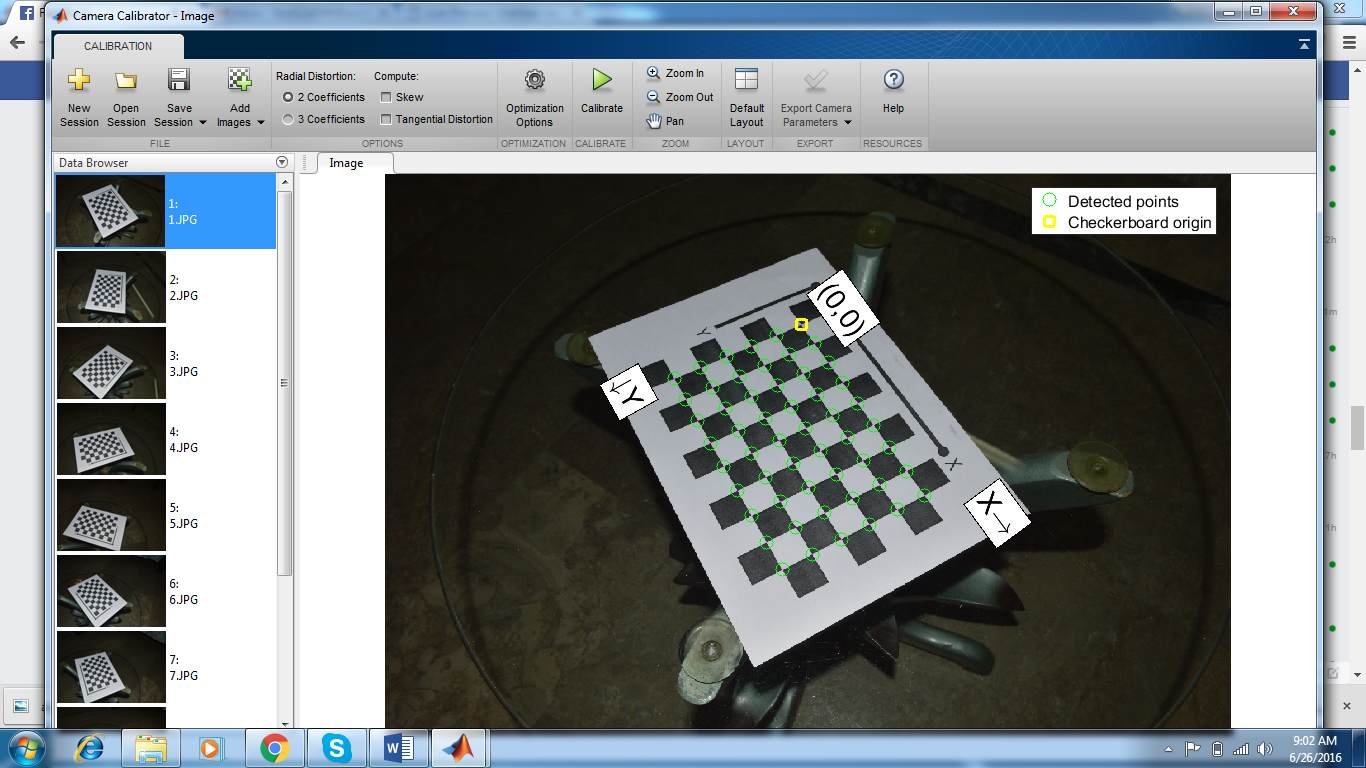
****

**Method Using Calibrated camera images:**

Following steps are performed to get 3D model from multiple images

**Calibrate Camera:**

Calibrate the camera using chess board pattern and taking point correspondences to estimate camera intrinsic parameters (we have used MATLAB camera calibrator app).



**Remove lens distortion:**

Evert camera lens distort image to some extent and parallel lines do not remain parallel and become distorted. Take the distortion parameters from earlier camera calibration matrix and using these parameters remove camera lens distortion.

Formulas used to remove lens distortion:



Kn  = nth  radian distortion coefficient

rn is the distance from origin







**Feature detection:**

In this scenario the main feature of images to be reconstructed are their corners. Corners correspondences will be made because corners are those features which will remain same in multiple images. For this purpose, we will use feature detection techniques like Harris Corner detector or Minimum Eigen values.

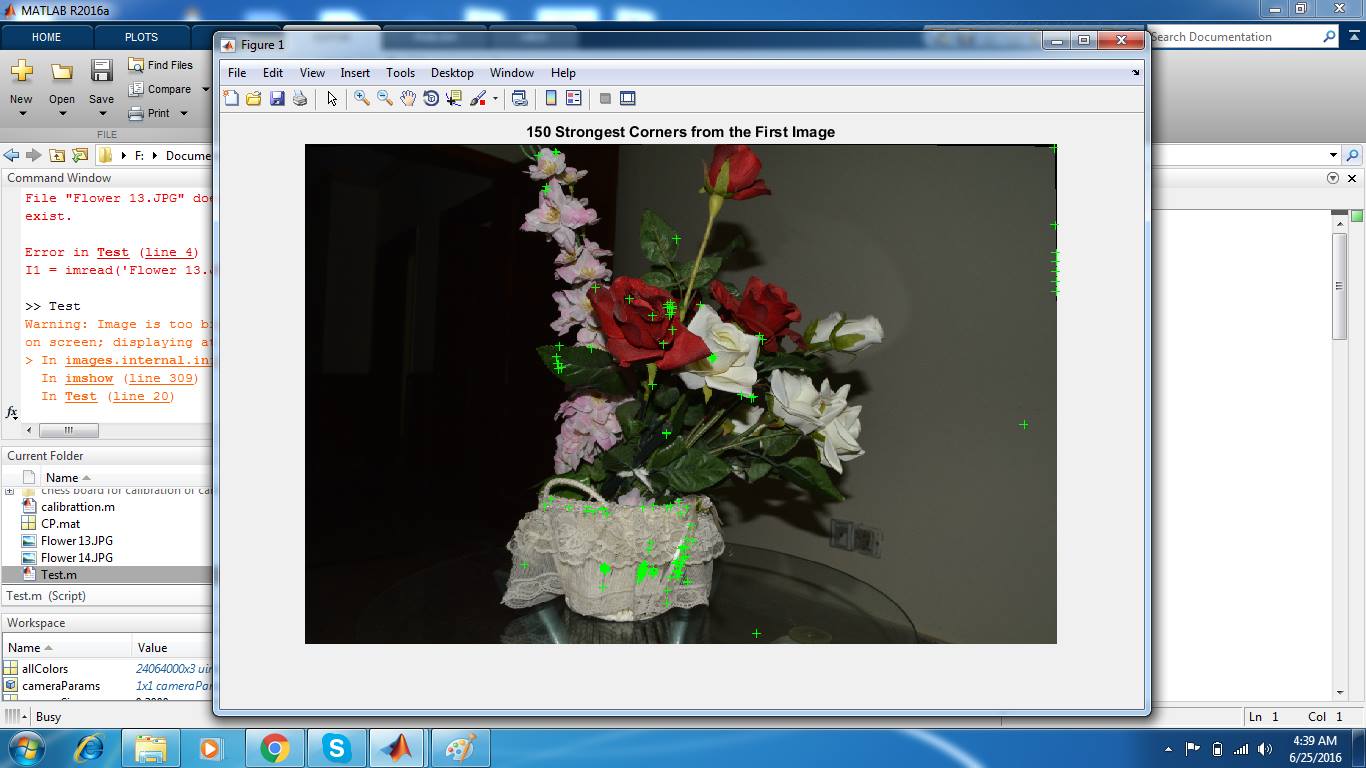
J. Shi and C. Tomasi made a small modification to Harris Corner Detector in their paper **Good Features to Track** which shows better results compared to Harris Corner Detector. The scoring function in Harris Corner Detector was given by:

R = \lambda_1 \lambda_2 - k(\lambda_1+\lambda_2)^2

Instead of this, Shi-Tomasi proposed:

R = min(\lambda_1, \lambda_2)

If it is a greater than a threshold value, it is considered as a corner. If we plot it in \lambda_1 - \lambda_2 space as we did in Harris Corner Detector, we get an image as below:



*Properties of features:*

* *Partially invariant* to affine intensity change
* Corner location is covariant w.r.t. translation
* Corner location is covariant w.r.t. rotation
* Corner location is not covariant to scaling!

**Feature tracking:**

Take strongest 150 features (corners) from first image and track those features on all other images.

There are different methods for feature tracking. Some take into account the covariance of features with respect to geometric transformations and photometric transformations.

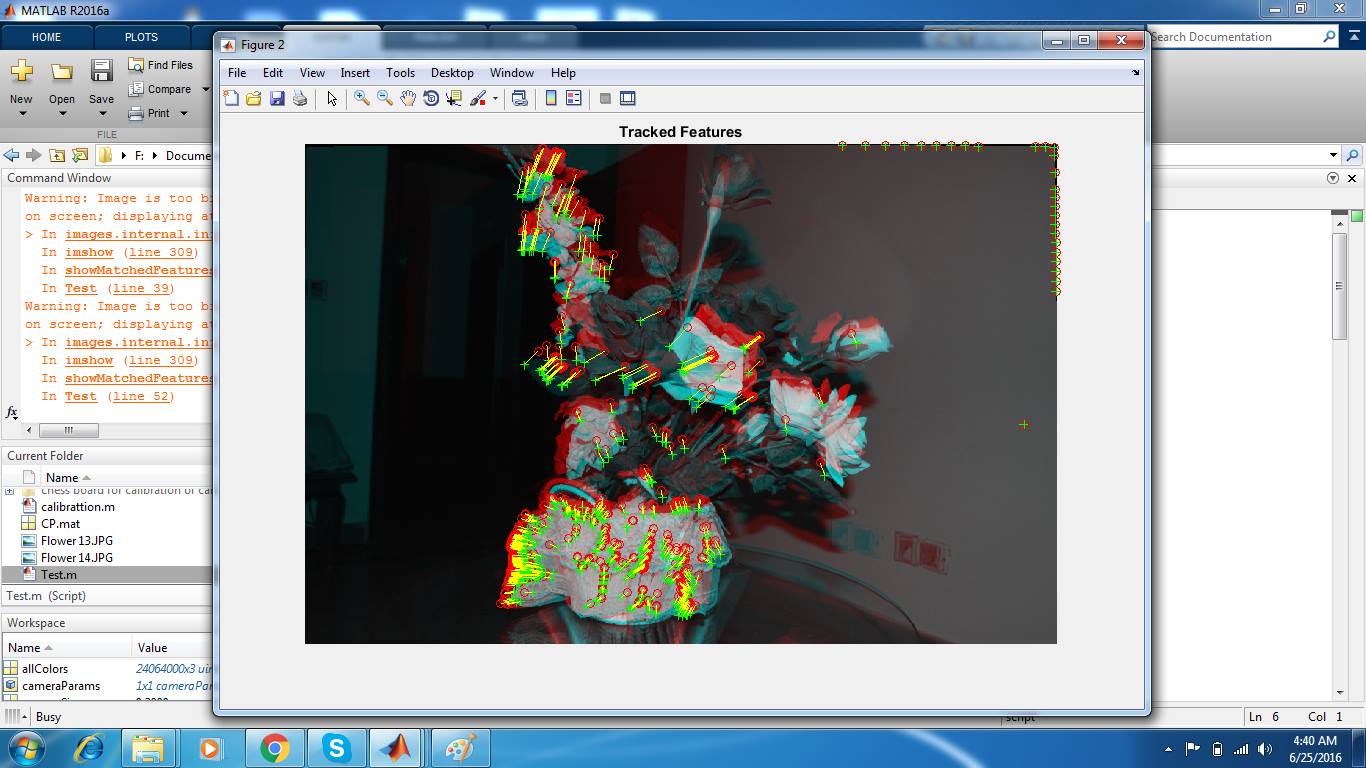
1. Zero Mean correlation (not invariant to photometric transformation)
2. Sum of Squared differences (not invariant to photometric transformation)
3. Normalized cross-correlation.

These methods do not handle geometric transformations. So we need other techniques for template matching.

1. MultiScale Oriented PatcheS descriptor. (rotation invariant)
2. Scale Invariant Feature Transform (invariant to affine transformation)

We used Kanade–Lucas–Tomasi feature tracker because of the following reasons :

1. Images are not that much different from each other
2. Prior Spatial information is available.
3. Fewer potential matches.



**Camera Pose Estimation:**

1. **Estimate Fundamental Matrix:**

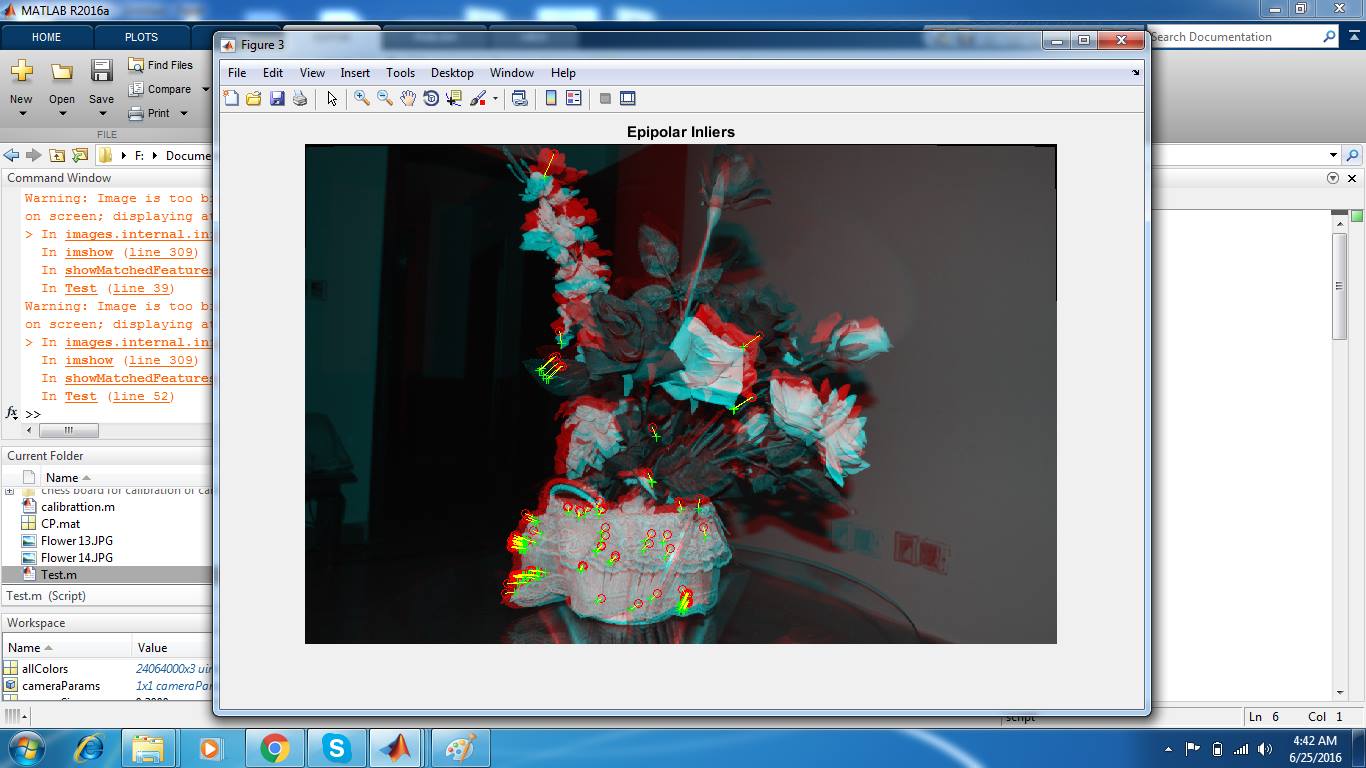
The fundamental matrix **F** is a 3×3 matrix which relates corresponding points in stereo images. In epipolar geometry, with homogeneous image coordinates, x and x′, of corresponding points in a stereo image pair, Fx describes a line (an epipolar line) on which the corresponding point x′ on the other image must lie. That means, for all pairs of corresponding points holds

**X’TFx = 0**

F matrix can be calculated using *Eight-Point Algorithm.* OR *Seven Point Algorithm.*

1. **Find Epipolar inLiers :**

Find the features that lie on Epipolar lines between two consecutive images.



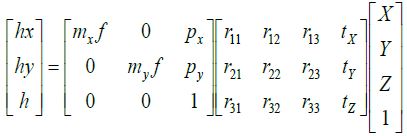
Compute the rotation and translation between the camera poses corresponding to the two images using precalculated camera intrinsic parameters, fundamental Metrix, and corresponding inlier points of two consecutive images.



**Camera Matrix:**

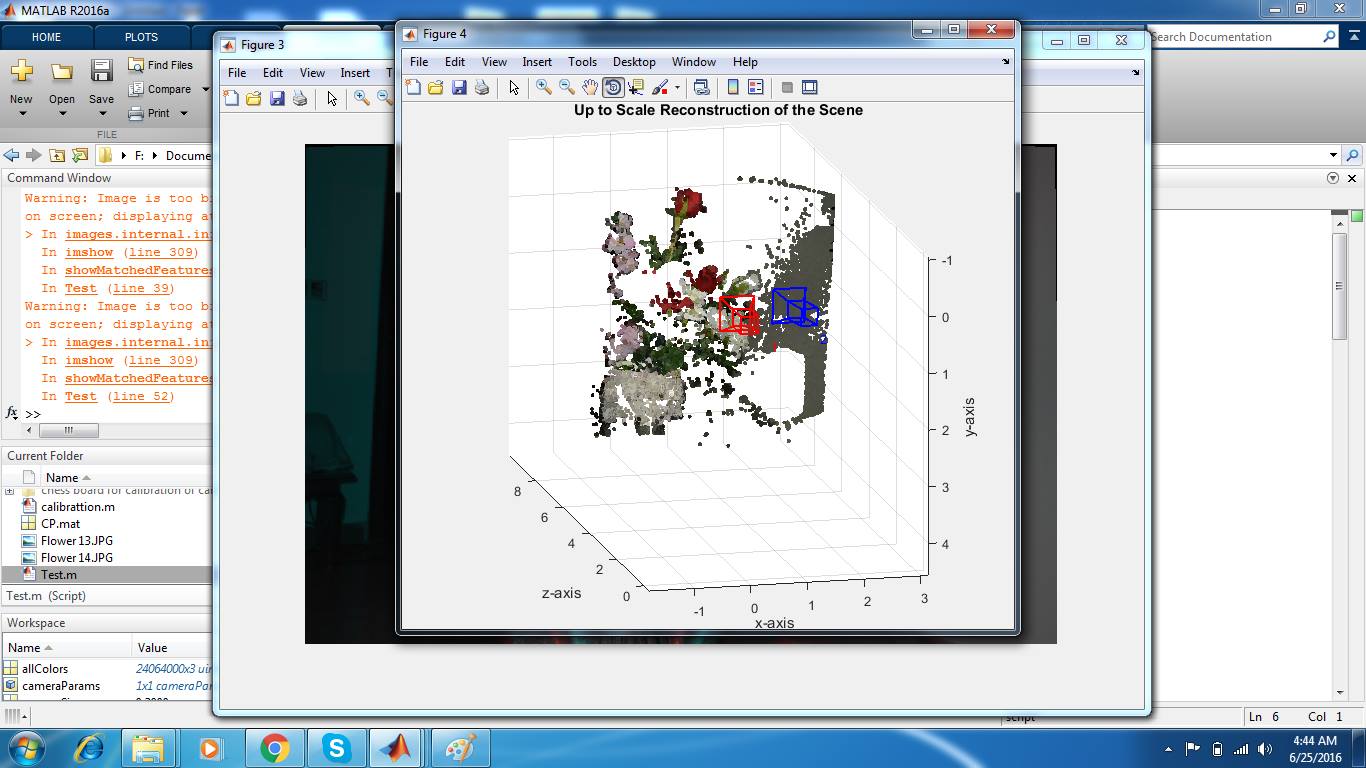
Compute the camera matrices **K**n of all images with respect to their deviation from original camera pose.





**Triangulate 3D points:**

Backward project matching 2D points from all images, to create a dense 3D point cloud. This will give you a projective reconstruction of original 3D model.



**Difference in Uncalibrated Images:**

There are two main differences when using images from a uncalibrated camera. (Or different camera even).

1. Camera Pose Estimation.
2. Camera Matrix Calculation.

Because we have no prior information about the camera so these tasks become somewhat difficult to do accurately. But we can estimate to some extent.

**Camera Pose**

Take matching features from images and estimate fundamental matrix (3x3 transformation matrix) using least square error method.

X2iT \* F \* X1i = 0

For the matches X1i <> X2i

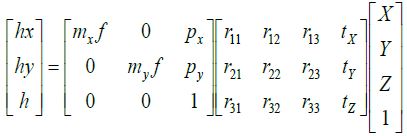
Determine epipolar inliers based on matched features and fundamental matrix.

**Camera Matrix:**

The camera matrix Pf is required to backward project 2D image points to their corresponding locations in 3D world.

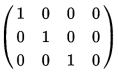
Where f is the frame number or picture number.

Camera matrix is composed of camera intrinsic parameters and extrinsic parameters. In this case intrinsic parameters are unknown



But we do have extrinsic parameters (3D rotational matrix and translation vector).

To calculate Camera Matrix, we first initialize intrinsic matrix as identity

.

First we project 2d points using these intrinsic parameters. Moving onto next image, because of rotational and translational parameters camera matrix changes. Now projected points have some different locations from the first one.

Take the mean of the first and next 3d points and using these new normalized 3d points we estimate camera matrix for the next set of images.