2014

Efficient Dense 3D Reconstruction Using Image Pairs

Padmapriya Ravi
Louisiana State University and Agricultural and Mechanical College

EFFICIENT DENSE 3D RECONSTRUCTION USING IMAGE PAIRS

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

in

The School of Electrical Engineering and Computer Science

by
Padmapriya Ravi
B.E., Anna University, 2012
May 2015

Acknowledgments

I would like to express my sincere gratitude to my advisor Dr.Xin Li for his support and guidance throughout my thesis. Without his valuable suggestions and constructive direction, this thesis would not have been successful. I also thank Dr.Suresh Rai and Dr.Hongchao Zhang for consenting to be on my thesis committee and supporting me through it. My special thanks to Sarah Marchiafa for giving me Graduate Assistantship throughout my masters program. Her support is very important for my degree.

I thank my parents for being a constant source of encouragement at all times during my study at LSU. This thesis would not have been possible without their love and support. I would like to extend a special note of thanks to Anton Joe for helping me throughout the thesis and masters program. Finally, I thank all my friends who made my stay at LSU enjoyable and memorable.

Table of Contents

Abstract

The 3D reconstruction of a scene from 2D images is an important topic in the field of Computer Vision due to the high demand in various applications such as gaming, animations, face recognition, parts inspections, etc. The accuracy of a 3D reconstruction is highly dependent on the accuracy of the correspondence matching between the images. For the purpose of high accuracy of 3D reconstruction system using just two images of the scene, it is important to find accurate correspondence between the image pairs.

In this thesis, we implement an accurate 3D reconstruction system from two images of the scene at different orientation using a normal digital camera. We use epipolar geometry to improvise the performance of the initial coarse correspondence matches between the images. Finally we calculate the reprojection error of the 3D reconstruction system before and after refining the correspondence matches using the epipolar geometry and compare the performance between them.

Even though many feature-based correspondence matching techniques provide robust matching required for 3D reconstruction, it gives only coarse correspondence matching between the images. This is not sufficient to reconstruct the detailed 3D structure of the objects. Therefore we use our improvised image matching to calculate the camera parameters and implement dense image matching using thin-plate spline interpolation, which interpolates the surface based on the initial control points obtained from coarse correspondence matches. Since the thin-plate spline interpolates highly dense points from a very few control points, the correspondence mapping between the images are not accurate. We propose a new method to improve the performance of the dense image matching using epipolar geometry and intensity based thin-plate spline interpolation. We apply the proposed method for 3D reconstruction using two images. Finally, we develop systematic evaluation for our dense 3D reconstruction system and discuss the results.

Chapter 1
Introduction

In recent decades, the problem of reconstructing a 3D scene from 2D images has been one of the important topics of research in Computer Vision area. This is due to the high demand for the 3D content for various applications like gaming, animations, human computer interaction, part inspection, face recognition and much more[1]. There are many scenarios where the 2D data is insufficient for an application which in turn imposes the need to reconstruct a 3D model from 2D images. One such example is in preserving and representing old architectural buildings and monuments for the future generations. Taking a picture of such buildings or statues will not give us enough information to appreciate and understand the details and the beauty, whereas a 3D model can give a realistic and detailed picture[16]. The machine vision industry has started using 3D reconstruction technique for identifying defects in the machine parts. Earlier, a model of the target machine part used to be built in CAD to calculate the error. This is expensive and time consuming. By reconstructing a 3D model, the parts can be inspected in real time with greater accuracy[17]. Another example where the 3D data out performs the 2D data is in face recognition[1]. 2D face recognition is not invariant to illumination changes in general. Also, it does not handle the effect of pose variation well. While in the 3D face recognition, even though changes in the illumination changes the texture, it does not change the shape of the image. The variation in the pose can also be recognized well in 3D face model.

The method for 3D reconstruction can be broadly classified into two types. Namely, *active* and *passive* methods[1]. In active method, special types of light sources are used to find the depth of the scene. For example, the depth information can be captured by *Microsoft Kinect*, which in turn can be used to reconstruct a 3D scene in real-time. In passive method, no such special light sources are used. In passive method, two types of reconstructions are possible. One is by using calibrated rigs, where two or more cameras are placed in a fixed position around the target object. In this method, the motion between the cameras are already known. Therefore reconstructing the target object is not very difficult[1]. The other one is by using Structure from Motion technique where a single camera is used to take the picture of the same scene from different orientations. Here, the position or motion of the camera is not known. The former method is expensive and also requires elaborate calibration setup. While in the latter method, a single normal digital camera can be used to reconstruct the 3D scene[1]. In this thesis, we use Structure from Motion

method to implement a 3D reconstruction system by using just two images of the scene.The main disadvantage of using passive method for 3D reconstruction is that it does not have the direct depth information of the scene, therefore it requires a minimum of two images of the scene to calculate the depth information. On the other side, passive method does not require the use of special cameras with special light properties. It can be implemented by using normal digital cameras. There are two big challenges in passive 3D reconstruction method. The first one is to find very accurate correspondence points between the two images as the accuracy of 3D points directly depends on the accuracy of initial correspondence points. In this thesis we use the epipolar constraint for the initial correspondence obtained using SURF to improve the accuracy. The second challenge is to find dense 3D points in order to represent the scene with more details. Most of the robust algorithms available gives only coarse correspondence between two images which is not enough to calculate the dense 3D points. In this thesis we propose a new method to get accurate dense correspondence between the images by using thin plate spline interpolation with intensity and epipolar constraint.

1.1 Understanding image formation

Before understanding how 3D scenes are reconstructed from image pairs it is important to understand how two dimensional images are formed from three dimensional scenes because taking a picture of a scene in camera is exactly the reverse process of reconstructing a 3D scene from 2D images[1]. Consider a simple pinhole camera to understand how images are being formed. When we click a picture in camera, rays of light from the scene passes through the

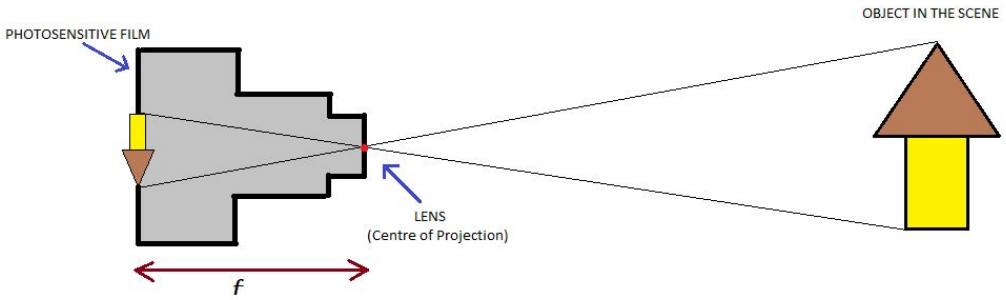

Figure 1.1: Pinhole camera model

lens in the camera and falls on the photosensitive film present inside the camera (figure 1.1). The image formed on photosensitive film is called as photo-negative image. The image that we

see in photograph or computer is called a photo-positive image. It is obtained by projecting the scene in a hypothetical plane situated in front of the camera lens. The distance between the lens and the plane in front of the camera is exactly same as the distance between lens and the photosensitive film[1]. The center of the lens is called as the *center of projection*. The hypothetical plane in front of the camera is called as *image plane*. The distance between the

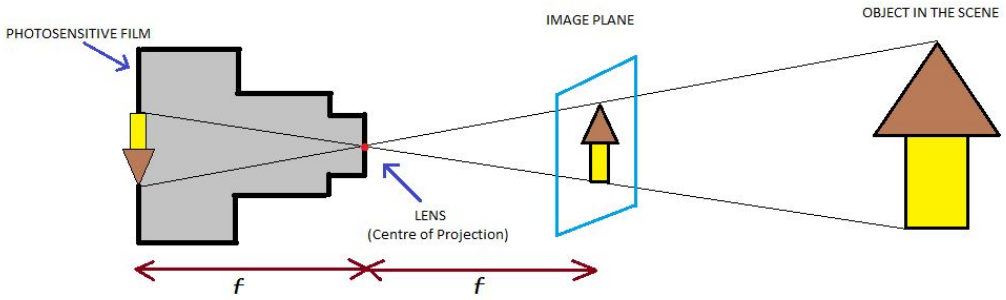

Figure 1.2: Image formation

lens and the image plane or photosensitive film is called as *focal length* (f) of the camera. These are depicted in figure 1.2. It is known that point,lines,curves and surfaces together, collectively form a scene in three dimensional Euclidean space[1]. In order to describe the above process

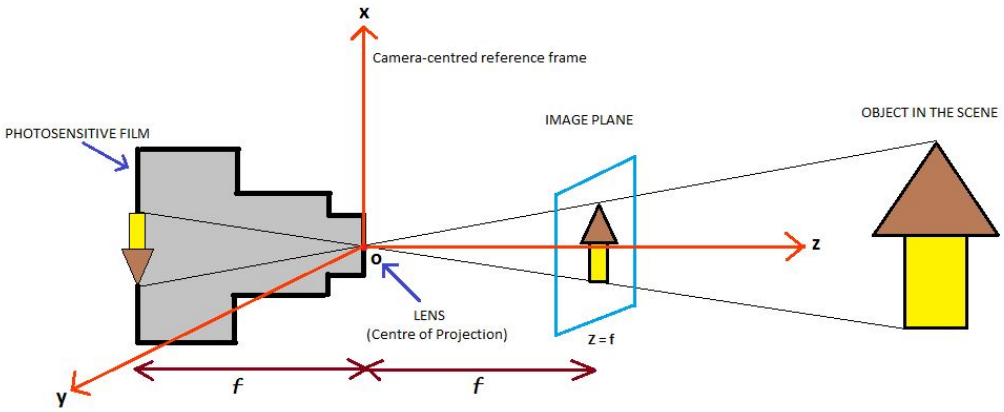

Figure 1.3: Camera centered reference frame

in the form of mathematical formulas, we should define a reference frame in Euclidean space.

To make the calculation easier it is fruitful to choose a reference frame that will depict how the camera has been setup[1]. The reference frame, shown in orange color in figure 1.3 is called as *camera centered reference frame*. The origin O, lies in the center of projection. The Z-axis is a line perpendicular to the image plane and passing through the origin. The X and Y axis are parallel to the image plane. It can be seen that the camera centered reference frame exactly defines the actual camera setup. As described before, the image plane is at a distance f from the the origin. Therefore the image plane is at $Z = f$ plane, where f is the focal length of the image. To develop a mathematical relation between a point in 3D scene and a point in 2D image,

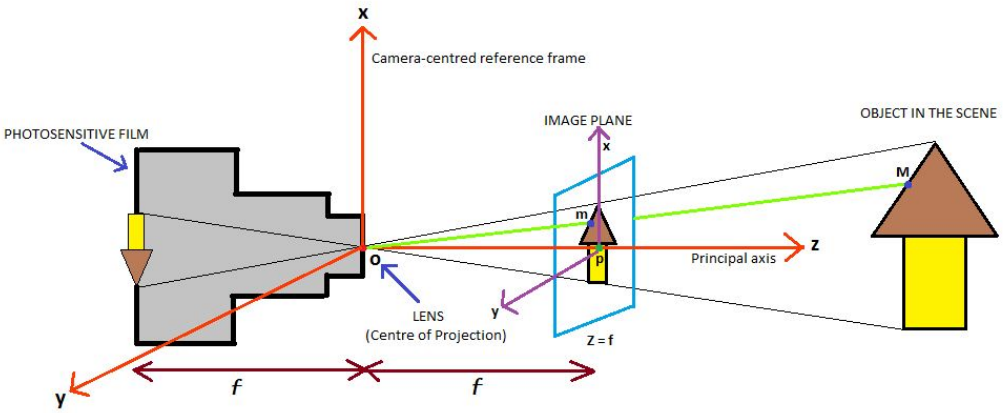

Figure 1.4: 3D-2D Image Coordinates

consider a point M in the actual scene and let m be the corresponding point in the 2D image[1]. The point m lies in the intersection of the line from M through the center of projection and the $Z = f$ plane as shown in (figure 1.4). If the coordinates of M is (X, Y, Z) then the coordinates of m will be $\rho(X, Y, Z)$ where ρ is some real number. m is nothing but an arbitrary point in the line through M and the origin of camera centered reference frame. Since we already know that the line from M through the center of projection intersects the image plane at $Z = f$, the point of intersection of the line with the plane for m should satisfy the condition $\rho Z = f$ or in other words $\rho = \frac{f}{Z}$. The coordinates of m can be written as (u, v, f)[1] where,

$$u = \rho X, \qquad v = \rho Y, \qquad f = \rho Z \tag{1.1}$$

Now by substituting $\rho = \frac{f}{Z}$ in (1.1) we get

$$u = f\frac{X}{Z} \qquad \text{and} \qquad v = f\frac{Y}{Z} \tag{1.2}$$

There are different ways of representing image coordinates with respect to the reference frame. In perspective projection, m is represented with respect to the *principal point* P[1]. It is the point where the principal Z axis of the camera meets the image plane. In Computer vision,

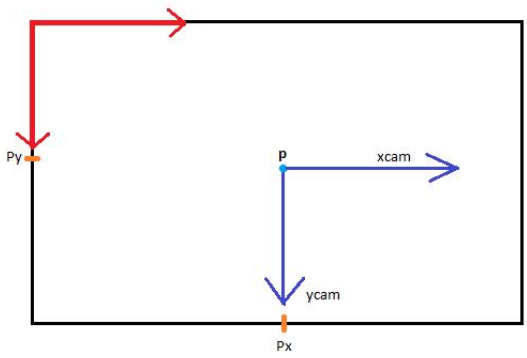

Figure 1.5: Representation of Image coordinates in Computer Vision

the image coordinates are represented with respect to the reference frame where the origin lies at the top left corner[1]. The X-axis is the horizontal axis which points towards right and Y-axis the vertical axis which points downwards as shown in(figure 1.5). There are two main reasons to choose this type of coordinates in Computer Vision[1]. Firstly, the X and Y-coordinates are placed in the same way in which a digital camera reads an image with CCD. Secondly, when we say that X is the horizontal axis pointing towards right and Y is the vertical axis pointing downwards, it automatically implies that the Z-axis is pointing away from the image perpendicular to X and Y axis. The Z-coordinate essentially corresponds to the depth of the scene which is the missing parameter for 3D reconstruction problem. Therefore, this type of representation is sensible for 3D reconstruction of a scene.

The image coordinates that we discussed so far is with respect to Principal point P. It has to be first converted in accordance with computer vision before modeling the image formation[1]. Let the coordinates of the principal points be (p_u, p_v). Then the projection of scene point M on

the image at point m will have the coordinates considering the offset of principal points[1].

$$\tilde{u} = f\frac{X}{Z} + p_u \qquad \text{and} \qquad \tilde{v} = f\frac{Y}{Z} + p_v \tag{1.3}$$

The image coordinates are represented in different units for different image coordinate systems[1]. The conventional image coordinate system with respect to the principal point are expressed in metric units. In digital camera the image points are represented in terms of pixels. The image

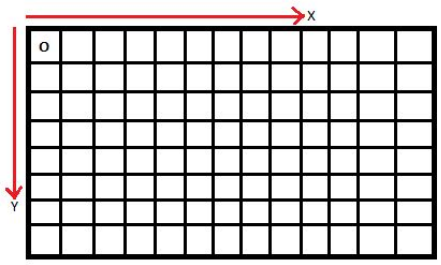

Figure 1.6: Pixel coordinates in digital camera

sensor in the digital camera is composed of array of pixels as shown in (figure 1.6). Each pixel position is represented by its row and column number in the array. These are referred to as Pixel coordinates. For example, the pixel coordinate of the first pixel in the topmost left corner is $(0, 0)$ and it increments towards right and downwards. In order to convert $(\tilde{u}, \tilde{v})$ to pixel coordinates, $\tilde{u}$ and $\tilde{v}$ has to be divided by the pixel length and the width respectively. Let m_u be the inverse of pixel length and m_v be the inverse of pixel width. The pixel coordinates (x, y) of the point m in the image is[1],

$$x = m_u(f\frac{X}{Z} + p_u) \qquad \text{and} \qquad y = m_v(f\frac{Y}{Z} + p_v) \tag{1.4}$$

Expanding (1.4)

$$x = (m_u f\frac{X}{Z} + m_u p_u) \qquad \text{and} \qquad y = (m_v f\frac{Y}{Z} + m_v p_v)$$

or

$$x = \alpha_x\frac{X}{Z} + p_x \qquad \text{and} \qquad y = \alpha_y\frac{Y}{Z} + p_y \tag{1.5}$$

where $\alpha_x = m_u f$, $\alpha_y = m_v f$, $p_x = p_u f$ and $p_y = p_v f$ are the focal lengths and principal points in x and y directions expressed in pixel coordinates. The point m in the image can be represented

in terms of homogeneous coordinates as $(x, y, 1)^T$ Now the equation(1.5) can be expressed in the matrix form as[1],

$$Zm = Z \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_x & 0 & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{1.6}$$

So far, we discussed about the mathematical modeling of image formation process with respect to the camera centered reference frame. Now, when we use more than one camera to represent the same scene then the scene is represented with respect to the non- camera centered reference frame known as world frame[1]. As seen in (figure 1.7), now two cameras

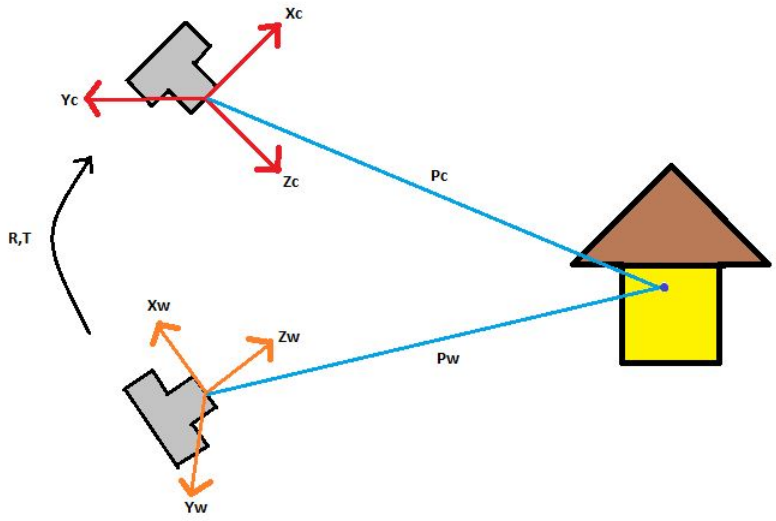

Figure 1.7: Pixel coordinates in digital camera

are used to represent the same scene. Each camera has its own reference frame (o_c, x_c, y_c) and (o_w, x_w, y_w) respectively. It is not possible to model the image formation with respect to one camera centered reference frame in this case. The position and orientation of the camera is described by indicating the origin, and the rotation and translation matrix describes how much the second camera should rotate to fit the first camera centered reference frame[1]. In other words, it describes the orientation of the camera centered reference frame with respect to the

world frame. The final expression for the camera model is[1],

$$Z\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_x & 0 & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{1.7}$$

where $(X, Y, Z, 1)^T$ are the homogeneous coordinates of the scene point in real world and $(x, y, 1)^T$ are the corresponding homogeneous coordinates on the image plane.

$K = \begin{pmatrix} \alpha_x & 0 & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{pmatrix}$ are the *intrinsic parameters* of the camera.

$P = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix}$ are the *extrinsic parameters* of the camera.

1.2 Understanding 3D reconstruction

The previous section discussed about the process of getting 2D image coordinates from a 3D scene. In this section the process of getting 3D scene coordinates from 2D images[1] will be discussed briefly. As we know, the images are formed from the rays of light reflected from the scene. To obtain the lost depth information back from the 2D images it is intuitive to project back the light rays from the image. In order to find the 3D coordinates (X, Y, Z) at point M for the corresponding point m in the image, a ray is projected from the camera center c through point m in the image as shown in figure 1.8. But it is clear from the Figure 1.8 that it is not sufficient to have one ray back project from the image to find its corresponding 3D coordinates since there is no way to find the coordinates at which the ray intersects the scene point M. Therefore, a minimum of two images of the same scene are required to reconstruct a 3D scene from 2D images[1]. Consider the figure 1.9. Here, the same scene is captured from two different cameras. Let m1 be the 2D point in the first image, m2 is the corresponding point of m1 in the second image. Now, when two rays are back projected from camera centers c1 and c2 through m1 and m2 respectively the rays intersect at point M to give the 3D points of the scene. There are two important approaches for 3D reconstruction. The first approach is to take two/multiple images of the same scene from two/multiple cameras at the same time using calibrated rigs. This is known as stereo 3D reconstruction. In this approach the motion between the cameras

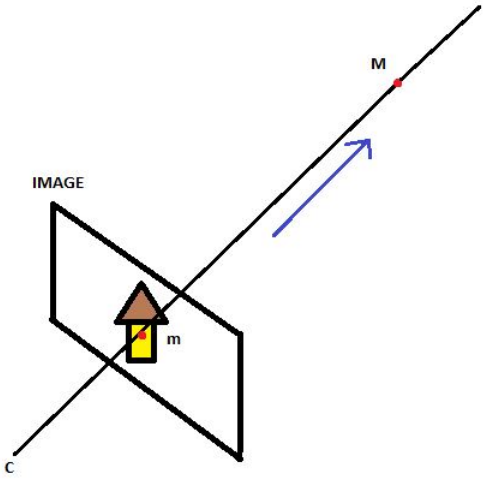

Figure 1.8: Light rays projected back from a single camera

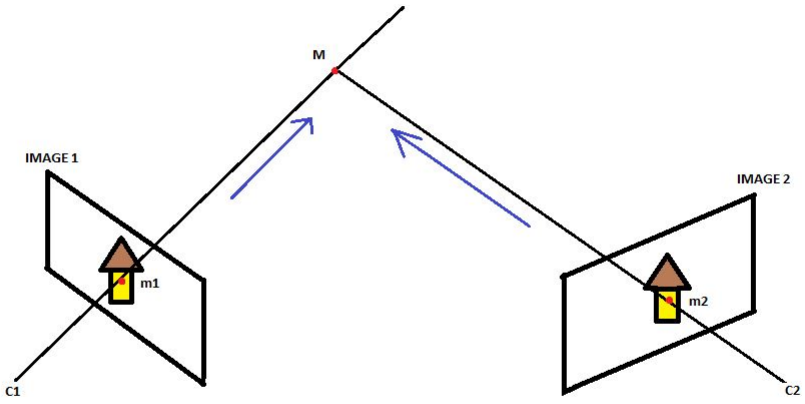

Figure 1.9: Light rays projected back from two cameras

are already known. The second approach is to use a single camera to take two/multiple images of the same scene from different views. This is known as Structure from Motion. In SfM we do not know the motion between the cameras. In this thesis we use Structure from Motion concept to reconstruct a 3D scene using just two images of the scene.

1.3 Pipeline for 3D reconstruction

3D reconstruction is carried out using Structure from Motion concept. Using a single digital camera, two pictures of the same scene are taken from two different positions.Once the images are taken,the first step is two find the correspondence between two images.Then the camera should be calibrated to find the internal camera matrix K which gives information about the focal length and the principal point of the camera as discussed in Section 1.1. The next step is to calculate the extrinsic parameters which gives information about the Rotation and translation between the two cameras as seen in Section 1.1. Once the intrinsic and extrinsic parameters are

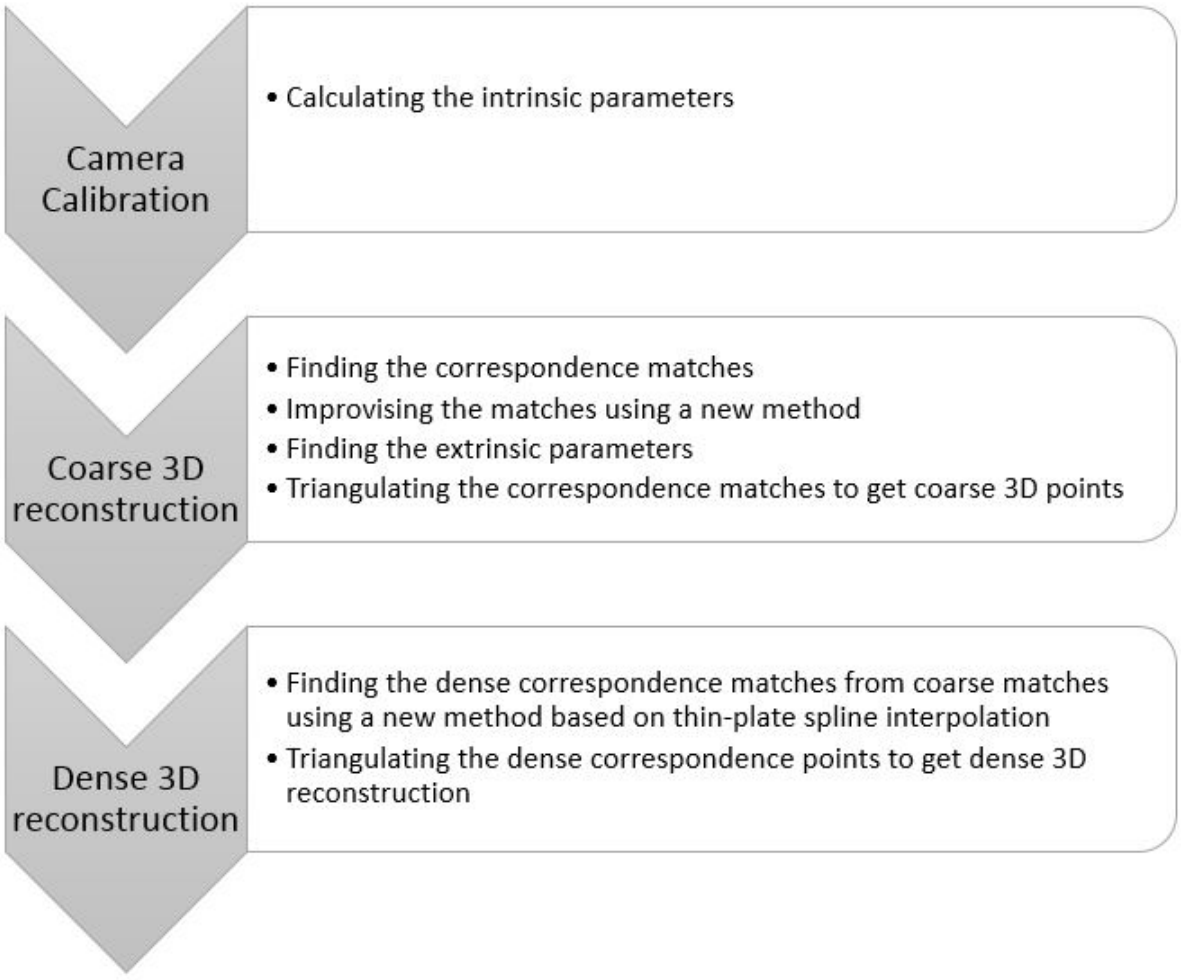

Figure 1.10: Pipeline for 3D reconstruction

calculated, triangulation has to be done to find the 3D points of the corresponding 2D points in the images. This will give sparse 3D reconstruction of the scene as the initial correspondence obtained between images are sparse. Then dense image matching is done to get a lot of correspondence between images. The final step is to do dense 3D reconstruction using the same

procedure above for dense image correspondences.The pipeline for 3D reconstruction is shown in figure 1.10.

1.4 Organization of the thesis

The remainder of this thesis has been organized as follows: Chapter 2 discusses the prior work in 3D reconstruction area. Chapter 3 provides an overview about camera calibration and epipolar geometry. Chapter 4 provides the implementation of 3D reconstruction system. Chapter 5 describes the proposed method for dense 3D reconstruction using Thin plate splines. Chapter 6 presents the experimental results and discusses about the coarse and dense 3D reconstruction.

Chapter 2
Literature Review

2.1 Calibrating the Camera

The problem of finding the camera calibration is highly important for metric 3D Reconstruction. It is used to find the intrinsic parameters of the camera which contains information about focal length and principal points. Different methods of camera calibration can be seen in the literature. In [2] and [3], camera calibration is done by using a special calibration object. A 3D object is placed in front of the camera whose geometry is already known. The calibration object is usually a single plane or two planes placed orthogonal to each other with some regular pattern marked on the plane. The 3D coordinates of some of the reference points on the plane are known in the coordinate system with respect to the calibration object. The reference points are chosen such that it is easy to obtain the coordinates of the projection of the points in the 2D image. Thus for each reference point, the 3D-2D relation is found. From this, the internal calibration matrix or the intrinsic parameters is found. Even though the method of using 3D calibration object in camera calibration gives very accurate results, it is very difficult to set up the calibration object for each 3D reconstruction system. [4] uses special off-the shelf TV cameras and lenses to calibrate the camera. The plane which undergoes a known translation is used to impose additional constraint in order to find the camera calibration matrix. One big downside of this method is that it requires very expensive calibration apparatus setup. Later, many techniques were introduced for camera calibration which does not require the use of any calibration object or special calibration setup [5], [6], [7]. In [5], the camera is made to move in a static scene. This uses the epipolar geometry between the two images which imposes two constraints on cameras' internal parameters. Thus two equations are formed for each corresponding points between two images.Therefore, the internal parameters can be found by solving a system of equations formed by many corresponding points between the two images. With the correspondence between three images both internal and external camera parameters can be found, if the same camera is used with constant internal parameters. This helps to reconstruct a 3D scene up to a similarity from 2D images [6],[7]. In [8], Caprile and Torre proposed a method which uses vanishing points for calibrating the camera. The internal camera parameters, that is, the focal length and the principal points are recovered by taking a single image of a cube and by using the properties of the vanishing points. The purpose of using cube is to find the set

of parallel lines at different directions, since the vanishing points lie on the intersection of the parallel lines. The extrinsic parameters can be found by using a pair of suitable fixed pattern from which the vanishing points can be easily obtained.The corresponding vanishing points are matched between two images and the rotation matrix is computed from them. The translation vector is then found by using simple triangulation. Even though this method does not require elaborate calibration setup, it still requires the use of cube to find the vanishing points. In 1994, Hartley proposed a self-calibration method that depends purely on rotation [9]. In this method, the intrinsic parameters are found by using the correspondence points between three images. The images are taken at the same point but at different orientations. In other words, the camera is fixed at a point and rotated to three different directions to capture three pictures of the same scene. Since there is no translation or movement in the camera, this method differs fundamentally from [5]. It is easier to find the correspondence points between images when there is just rotation.

2.2 Image matching

The problem of finding correspondence points between two images of the same scene is the key step in 3D reconstruction problem. Many research have been done in the development of image matching by using a set of local interests point. In [10], Moravec proposed a stereo matching technique using some corner detection. In this method, a local rectangular window is considered in the image and the average changes in the image intensity is determined by shifting the window by a small amount in different directions. If the window region has constant intensity, then there is a very small change in all the shifts. If the window reaches an edge, then there is a small change in the intensity along the direction of the edge but large change in intensity in the direction perpendicular to the edge. If the window reaches a corner or an isolated point then there is a large change in intensity along all the sides of the shifts. Thus the corner can be identified when the change in intensity produced by all the shifts are large. Mathematically, the change in intensity produced by the shifts is given by

$$E_{x,y} = \sum_{u,v} w_{u,v} |I_{x+u.y+v} - I_{u,v}|^2 \tag{2.1}$$

where w is the window,the value of w is 1 withing the specified region and 0 elsewhere. The shifts (x, y) are $(1, 0), (1, 1), (0, 1), (-1, 1)$ Thus for detecting the corner, it simply takes the local maxima in min[E] greater than some specific threshold. There are some drawbacks in this method. Firstly the result is anisotropic because the shifts used to determine the intensity

change is very discrete. Secondly, the result is noisy due to the rectangular and binary window. Thirdly the operator responds to the edges along with the corners since only the minimum of E is taken into account.

The drawbacks of Moravec's detector was overcome by Harris Detector in 1988. In [11], Harris and Stephens proposed a combined corner and edge detector. The change in intensity E for the small shift (x,y) is given by

$$E(x,y) = (x,y)M(x,y)^T \tag{2.2}$$

where M is a 2×2 symmetric matrix.

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \tag{2.3}$$

M is the quadratic terms in Taylor expansion. Let α and β be the eigen values of M. The corner response of Harris detector is given by

$$R = Det - kTr^2 \tag{2.4}$$

where,

Det is the determinant of the matrix M, given by

$$Det(M) = \alpha\beta = AB - C^2 \tag{2.5}$$

Tr is the trace of the matrix M, given by

$$Tr(M) = \alpha + \beta = A + B \tag{2.6}$$

If the value of R is positive, then it is corner region. The negative value of R denotes the edge region and if the R value is small, it is considered as flat region. This method is invariant to rotation and slight changes in illumination. Harris corner detector is one of the most popular detector and it is broadly used for many image matching applications. In [12] Zhang et al proposed an image matching technique which mainly depends on the epipolar constraint. Thus any number of images can be taken for the same scene and it is not required to know the position of the cameras. But if the correspondence points are searched based on the epipolar geometry,

then the complexity will be high. Therefore standard interest point detectors like Harris corner detector is used to find the initial matches and then the epipolar geometry is used to eliminate the outliers. This method has been tested for different input scenarios and it works well if the scene has repeated input pattern.

In [13], Lowe proposed a method to find the distinctive image features that is scale and rotation invariant and also provides robust matching across addition of noise, change in the viewpoint of the three dimensional scene, variation in illumination and some range of affine distortion. For the method to be scale invariant, it is clear that same window cannot be used to detect keypoints with different scales. To address this problem, some scale-space filtering has to be done. Generally, Laplacian of Gaussian is used which detects blobs at different scales σ. Thus σ acts as the scaling parameter through which the keypoint (x,y) at σ scale can be found. In this paper, SIFT (Scale Invariant Feature Transform) uses difference of Gaussian instead of LoG, since the computational cost of LoG is high. Once DoG is found, the image which has local extrema over different scale and space is considered as the potential scale. For rotation invariance, the gradient magnitude and direction is calculated in the region surrounding the keypoint. Finally the keypoint descriptors are formed by taking a 16×16 neighborhood around the keypoints. They are represented in the form of vectors. Images are matched by calculating the nearest neighbors. SIFT is widely used in many applications since it is both scale and rotation invariant.

2.3 Coarse and dense correspondence matching

Finding a good initial coarse correspondence matches between the images is a crucial step to calculate the parameters required for 3D reconstruction. In [20], Li and Chen proposed a method to find a non rigid coarse correspondence for volumetric images. In this method, the initial image features are extracted and correlated using an improvised 3DSIFT algorithm known as I3DSIFT algorithm. The feature correlation obtained using I3DSIFT algorithm is less sensitive to image rotation and scaling [20]. Finally, the one-to-one mapping between corresponding features are found using Improved spectral matching algorithm [20].

Even though coarse correspondence matches can be used to find the parameters required for 3D reconstruction, it is essential to find the dense correspondence matches in order to reconstruct the scene with high details. There are various methods to find dense correspondence matches from existing coarse correspondence such as [21], [22], [23], and [24]. In [22], Li and Xu proposed a method for 4D Image registration in which dense correspondence matches are found from coarse matches through interpolation technique which solves two 4D B-spline functions. The mapping

between correspondence points is done by solving the two B-spline functions by minimizing an objective function. The objective function is penalized for intensity matching error, feature alignment error, spatial and temporal non-smoothness and inverse inconsistency [22]. In [23] Li and Xu proposed another consistent feature-aligned algorithm where they first developed forward and inverse matching models based on feature alignment constraints and then refined the correspondence matches iteratively by using inverse consistency. In [24] Li et al proposed a 4D spatiotemporal image registration method which has both spatial and temporal smoothness. In this method, the initial coarse feature correspondence is found using an improved 3D SIFT descriptor which is invariant to scale and rotation. Then the dense correspondence matches are found from the coarse correspondence matches using an intensity based deformable image registration algorithm [24]. In [25], Li and Iyengar reviewed various methods for computing the 3D geometric data mapping where the dense correspondence matches can be calculated directly from the 2D/3D data. In this paper, various mapping algorithms are discussed and compared based on their formulations of objective functions, different constraints and the strategies used for optimization [25].

Chapter 3
Camera Calibration and Epipolar Geometry

3.1 Camera calibration

Camera Calibration is the most important step to get metric 3D Reconstruction. Neither knowing the camera calibration matrices nor the position of the cameras, will result in reconstruction done using the correspondence point of the images alone to be just projectively equivalent[1]. The projective reconstruction preserves only collinearity and coplanarity. It neither preserves parallelism nor the metric information. Hence, calibrating the camera will give information about the focal length and Principal points which helps in reconstruction to preserve parallelism and metric information[1].

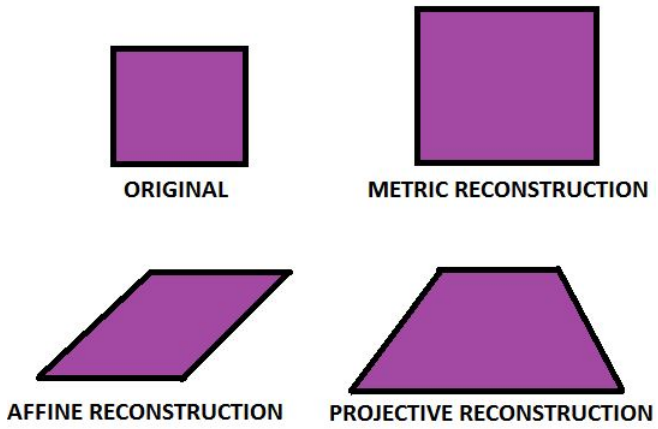

Figure 3.1: Types of Reconstruction

The different types of reconstruction of a square is shown in (figure 3.1). It can be seen that the square will look like a parallelogram if the calibration matrices are unknown. This is known as affine reconstruction where the metric information is not preserved. The projective reconstruction does not preserve even parallelism as explained above. The metric reconstruction preserves the shape as in the original image.

Initially Camera calibration was performed by observing 2 or more planes (calibration object) which were placed orthogonal to each other[15]. The planes had a fixed regular pattern whose geometry in 3-D space were known. Calibration was done by observing the corners of the pattern and equating the image coordinates with world coordinates as shown in (figure 3.2).

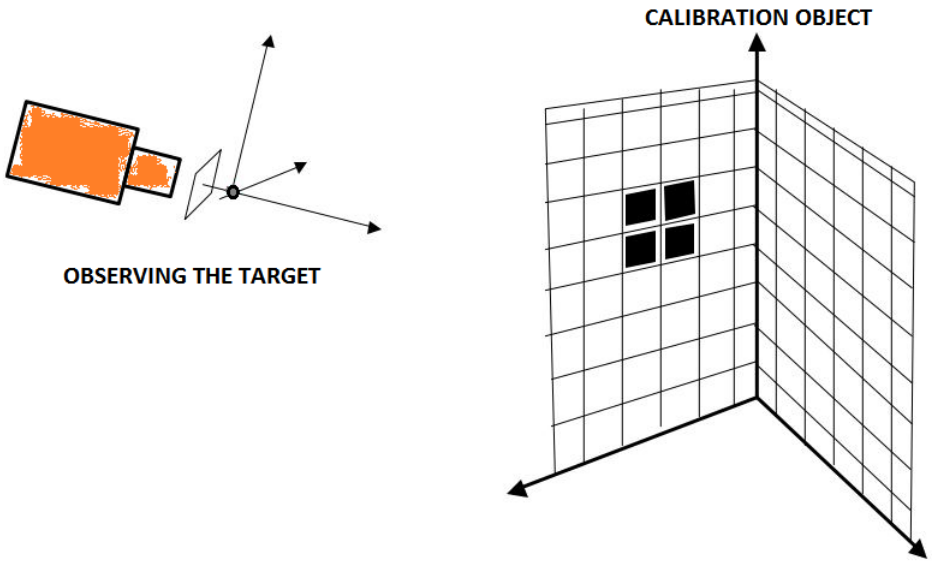

Figure 3.2: Camera Calibration using Calibration object

The disadvantage of this method is that it is very difficult to setup calibration object for each reconstruction and the process is expensive.

In [14] Zhang proposed "A flexible new technique for camera calibration". This technique does not require the use of any calibration object. In this method the camera is made to observe a checkerboard pattern. The checkerboard pattern with known dimensions is printed and placed on any planar surface. Pictures of the planar pattern are taken at different orientations by either moving the camera or the planar surface. It is not required for the motion of camera or the planar surface to be known.

The relationship between a 3D point M and its image projection m is given in equation 1.7. This can be written as[14]

$$Z\tilde{m} = K[R \quad t]\tilde{M} \tag{3.1}$$

where $\tilde{m} = (x, y, 1)^T$ is the 2D homogeneous coordinate $\tilde{M} = (X, Y, Z, 1)^T$ is the 3D homogeneous coordinate

The objective is to find the K matrix which is known as internal calibration matrix. Let us assume that the model plane is in $Z = 0$ plane. Therefore when we take a picture of this model

plane, the relationship between $\tilde{m}$ and $\tilde{M}$ is given by

$$Z\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K\begin{pmatrix} r_1 & r_2 & r_3 & t \end{pmatrix}\begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} \tag{3.2}$$

where $r_i$denotes the i^{th} column of the rotation matrix. Thus the above equation can be re-written as

$$Z\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K\begin{pmatrix} r_1 & r_2 & t \end{pmatrix}\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \tag{3.3}$$

The point M and its projection in the image m is related by a Homography H.

$$Z\tilde{m} = H\tilde{M} \tag{3.4}$$

Where $H = A[r_1 \quad r_2 \quad t]$. H is a 3×3 matrix which is defined up to a scale factor. From equation 3.4 it is clear that, there exists a homography H when the image of model plane is taken. Let $H = [h_1 \quad h_2 \quad h_3]$. From equation (3.4) we can write

$$[h_1 \quad h_2 \quad h_3] = \Lambda K[r_1 \quad r_2 \quad t] \tag{3.5}$$

where Λ is an arbitrary scalar. The two important properties of rotation matrix are[14], $r_1.r_2 = 0$ and $r_1^2 = r_2^2$. Using these properties in equation (3.5), we have

$$h_1^T K^{-T} K^{-1} h_2 = 0 \tag{3.6}$$

$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2 \tag{3.7}$$

Thus,when an image of model plane is taken with a camera, the homography will give rise to two constraints on intrinsic parameters. Let

$$D = K^{-T}K^{-1} \equiv \begin{pmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{pmatrix} \tag{3.8}$$

Since D is symmetric it can be defined by a 6D vector as

$$d = [D_{11}, D_{12}, D_{22}, D_{13}, D_{23}, D_{33}]^T \tag{3.9}$$

If the column vector of H is $h_i = [h_{i1}, h_{i2}, h_{i3}]^T$, then

$$h_i^T D h_j = u_{ij}^T d \tag{3.10}$$

with

$$u_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T \tag{3.11}$$

Therefore, the two constraints (3.6) and (3.7) on intrinsic parameters from a given homography can be rewritten as two homogeneous equations in d

$$\begin{pmatrix} u_{12}^T \\ (u_{11} - u_{22})^T \end{pmatrix} d = 0 \tag{3.12}$$

In general, equation 3.12 can be written as

$$Ud = 0 \tag{3.13}$$

If we take n images of the model plane then U is a $2n \times 6$ matrix. We need a minimum of two images to get the K matrix. The solution to equation 3.13 is the eigen vector of $U^T U$ with the smallest eigen value. Once we estimate d then all the unknowns in K can be estimated from equation 3.8. The above results can be refined through maximum likelihood estimation which is given in detail in [14]. This is the most convenient method as it only requires the camera to take an image of the model plane [14].

3.2 Epipolar geometry

Concept of epipolar geometry is essential to derive the extrinsic parameters for 3D reconstruction[16]. When we take two images of the same scene from two different views for 3D reconstruction, there prevails an intrinsic projective geometry between them which is known as the Epipolar geometry. In other words, epipolar geometry tells us how the two views are related to each other. The epipolar geometry does not depend on the scene structure. It depends on the cameras' internal parameters and their relative pose[16]. Suppose a point X in 3D space is being imaged by two

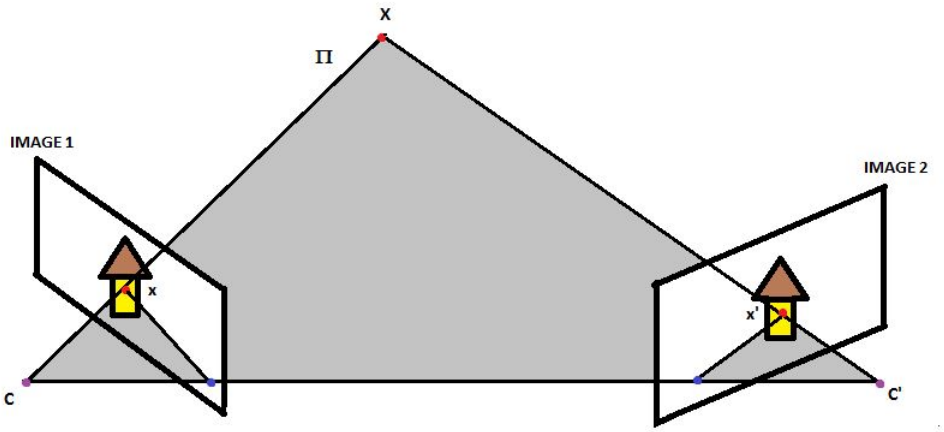

Figure 3.3: Epipolar geometry- The camera centers c and c', the correspondence points x and x', and the 3D- point X lie in a common plane π

different views at x and x' as shown in figure 3.3, then the epipolar geometry gives the relation between x and x'. As seen in figure 3.3 the image points x and x' , the point X in 3D space and the camera centers c and c' lie in same plane. In other words x, x', X, c and c' are coplanar. It can be clearly seen that the rays back projected from x and x' intersect the point in 3D space at X and also the rays x and x' lie on the same plane. Let this plane be denoted as π.The line joining the camera centers c and c' is known as *baseline*. As shown in figure 3.4, the point of intersection of the baseline with the image plane is known as *epipole*. Any plane which contains the baseline is called as *epipolar plane*. One such epipolar plane is highlighted in green color in figure 3.4. An *epipolar line* is the line where the epipolar plane intersects with the image plane. All the epipolar line intersects at epipole. The green lines on both the images are the epipolar lines in figure 3.4. Given point x in the first image, to find how point x' is constrained with respect to x, the first step is to find the plane π since we know that x, x' and X are coplanar. The plane π can be determined from the baseline and the ray extending from x. Once the plane π is known, one can find the intersection of the π and the image plane. Since we know that the point x' also lies in plane π it is clear that x' lies on the epipolar line l'. The ray back projected from point x in the first image is the line l' in the second image. Thus, it is not required to search in the entire image for correspondence points. The search can be restricted to the epipolar line l'[16].

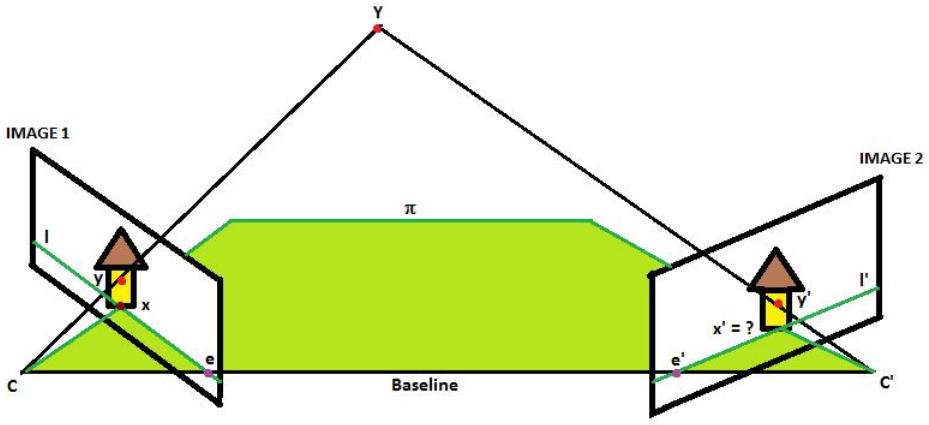

Figure 3.4: Epipolar geometry - Relationship between point x in the first image and the corresponding point x' in the second image

3.2.1 Fundamental matrix

The fundamental matrix represents the epipolar geometry in algebraic form[16]. We know from the previous section that point x in the first image maps to the epipolar line l' in the second image. There is a mapping from a point to a line. This mapping is used to derive the fundamental matrix. The map from point x through the camera center c to the line l' can be written as

$$x \mapsto l' \tag{3.14}$$

This mapping can be achieved in two steps[16]. First the point x in the first image is mapped to the corresponding point x' in the second image which lies on the epipolar line l'. Then the epipolar line can be easily determined by joining x' and the epipole e'. Figure 3.5 shows one of the methods to get point correspondence between two images. Here the corresponding points are found by transferring point via a plane. As shown in (figure 3.5) consider a plane π that does not pass through any of the camera centers. When a ray is back projected from point x in the first image, it will hit the plane at point X. The point X is then projected to x' in the second image. This process is known as Point transfer via Plane[16]. Since point X in 3D and point x in the first image lie on the same ray, the point x' projected from X will lie on the epipolar line l' corresponding to x. Thus all the points x_i in the first image and the corresponding points x'_i in the second image whose 3D points X_i lie on the same plane are projectively equivalent. When x_i and x'_i are projectively equivalent to the set of points X_i which lies on the same plane, then

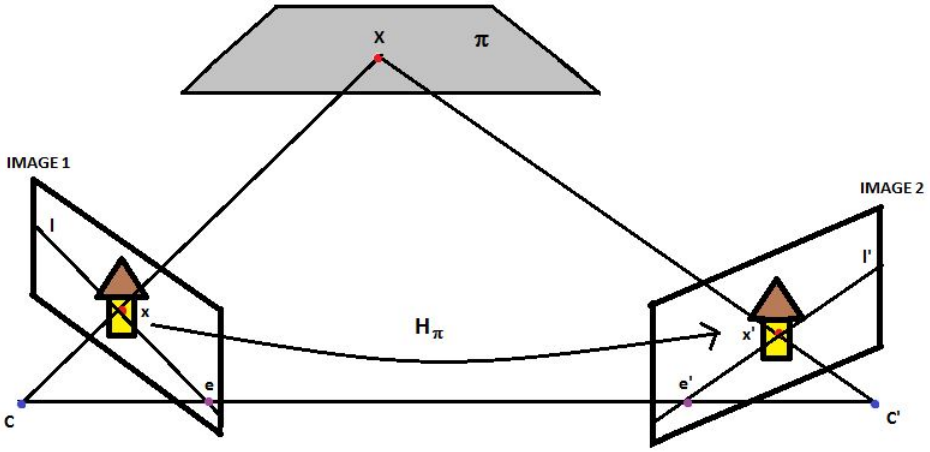

Figure 3.5: Epipolar geometry - Point transfer via plane

there exists an Homography mapping between x_i and x_i'.

As discussed before the epipolar line l' can be drawn by joining the points x' and e'. This can be written mathematically as[16]

$$l' = e' \times x' = [e']_\times X' \tag{3.15}$$

Where $[e']_\times X'$ is the matrix representation of the vector e'. Since we know that an homography mapping exists between x_i and x_i' x' can be written as

$$x' = H_\pi x \tag{3.16}$$

Substituting (3.16) in (3.15)

$$l' = [e']_\times h_\pi x = Fx \tag{3.17}$$

Where

$$F = [e']_\times H_\pi \tag{3.18}$$

is the Fundamental Matrix[16].

The Fundamental Matrix is a rank 2 matrix with 7 degrees of freedom[16]. One of the most important property of the Fundamental matrix is that if x and x' are the corresponding points

then

$$x'^T F x = 0 \tag{3.19}$$

This can be proved to be true because if x and x' are the corresponding points then it is obvious that the point x' lies on the epipolar line l'. This can be written mathematically as

$$x'^T l' = 0 \tag{3.20}$$

Substituting (3.17) in (3.20) we get $x'^T F x = 0$.

The fundamental Matrix holds the transpose property[16]. That is , if F is the fundamental matrix for a pair of cameras (1,2) then F^T is the fundamental matrix for the same pair of cameras in opposite order(2,1). Similarly if $l' = Fx$ is the epipolar line for the corresponding point x, then $l = F^T x$ is the epipolar line for the corresponding point x'. These are some of the important properties of Fundamental matrix[16].

Chapter 4
Implementation of 3D Reconstruction System

This chapter explains in detail about the implementation of our 3D Reconstruction system. Reconstruction of a scene is done by using just two images of the same scene. A normal digital camera is used to take the images. The two images are taken from two different views in such a way that it best describes the scene. The focal length of the camera is maintained same as the focal length used for calibrating the camera.

4.1 Calibration of camera

The first step in reconstruction is camera calibration. As discussed in the previous chapter, camera calibration is essential to get metric 3D reconstruction. Zhang's algorithm[14] is used for calibrating the camera. The details of Zhang's algorithm is given in Chapter 3. This is implemented in C++ using OpenCV library[17]. OpenCV has inbuilt functions for camera calibration. To calibrate the camera at a fixed focal length, a black and white checkerboard image is printed on a paper and placed on a planar surface as shown in figure 4.1. This is known as model plane. Snapshots of this model plane is taken in different views. Each snapshot accounts for one equation 3.12 as explained in Chapter 3. In theory two snapshots are required to find all the parameters in K matrix. But in practice more than two images are required as the input images have a good amount of noise. A minimum of eight snapshots of the model plane in different positions are required for good results.The number of inner corners in the checker board pattern and the size of the square is given as input. Once the input images are given, the program checks for the input pattern in the images as shown in (figure 4.2). In this case it checks for the corners of square in the given image. The program will throw an error if it does not find enough good images to detect the pattern. Then all the unknown parameters in the K matrix are determined by solving equation 3.13 which are formed by the input images.

4.2 Feature detection and extraction

The accuracy of 3D reconstruction totally depends on how well the correspondence points are found between the images. If the point correspondence between the images are not accurate, then this will lead to wrong Fundamental Matrix which will finally give erroneous reconstruction. Therefore this is a very crucial step in 3D reconstruction. There are a wide variety of detectors and descriptors being proposed in literature. Choosing a right detector and descriptor plays

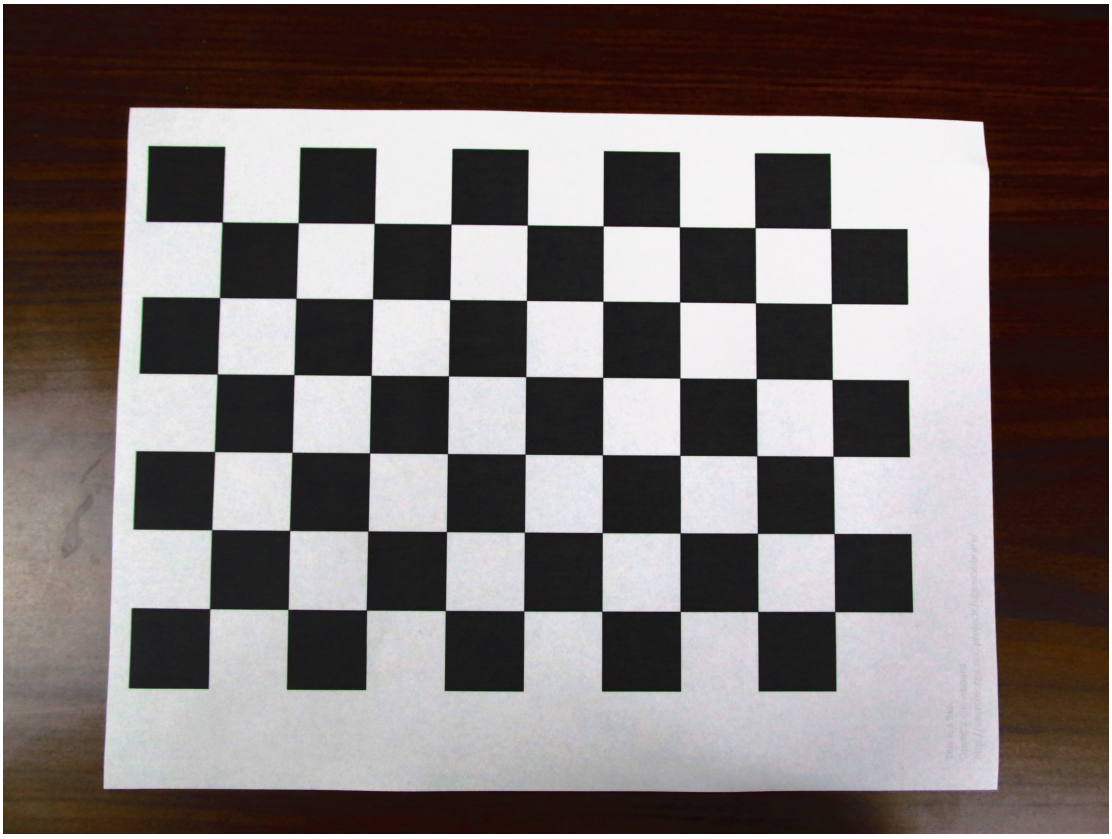

Figure 4.1: Model plane for Camera Calibration

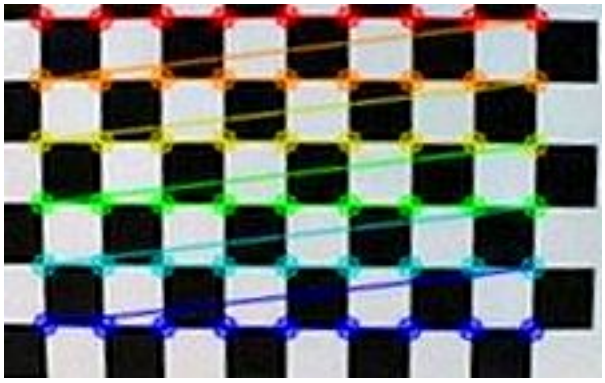

Figure 4.2: Input Pattern in the checkerboard image

a vital role in reconstructing a scene from images of the same scene. In our implementation, SURF(Speeded Up Robust Features) has been used for feature detection and extraction[18]. The reason for choosing SURF is that it is computationally fast, robust and scale and rotation invariant[18]. The first step is Feature Detection,that is finding interest points in both the images. For example finding the corners, edges, T-junction,etc. Once the interest points have been found in both the images, the next step is to find the feature descriptors. Feature extractors

or descriptors means representing the interests points in low dimensions. It is highly essential to reduce the dimensionality while representing the interest points otherwise the computation will take a very long time for matching the interest points. Also the input data can be redundant, hence it is better to reduce the dimensions of the features extracted that will define the features best for matching it with other features in different images. For example some of the feature descriptors can be the intensity of that point, orientation of the area surrounding the point, gradients of the area surrounding this point etc. SURF uses reduced descriptors compared to other algorithms[18], which is the main reason for it to be computationally fast. SIFT(Scale-Invariant Feature transform)[13] keypoint and Descriptor is also scale and rotation invariant but the difference between SURF and SIFT is that SIFT approximates Laplacian of Gaussian with Difference of Gaussian in order to find the scale-space whereas SURF approximates Laplacian of Gaussian by using box filter. The advantage of using box filter is that the convolution can be easily calculated by using integral images. To determine the orientation SURF uses wavelet responses which can also be determined easily using integral images. SURF is faster than SIFT but in terms of performance both of them are comparable to each other. The details of SURF keypoints and descriptors can be found in [18]. SURF is implemented using OpenCV library. OpenCV has functions for finding SURF keypoints and descriptors. In our implementation, the two images of the same scene at different views are given as input. SURF gives only coarse correspondence between two images as it looks for interests points like corners,edges,blobs, T-junction etc. This will not give correspondence points for every pixel in the image but the keypoints found are highly robust. Fewer, highly robust keypoints are sufficient to calculate the cameras' extrinsic parameters. Thus SURF is used to find initial coarse correspondence for 3D reconstruction.

4.3 Feature matching

Once the keypoints and descriptors are found in each image, the features are matched between the two input images. In our implementation, *brute force matcher* has been used for Feature matching[17]. For every feature in the first image, the descriptors of that feature is taken and it is compared with the descriptors of all the other features in the second image using the distance of the vectors. The matcher returns the feature points in both the images that has least distance between them. The distance of the feature vector is calculated using $l^2 - norm$.

For example, if

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ x_n \end{bmatrix} \tag{4.1}$$

is the feature descriptor of the first image and

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ . \\ . \\ y_n \end{bmatrix} \tag{4.2}$$

is the feature descriptor of the second image. Then the $l^2 - norm$ of the first and second feature descriptors will be

$$|X| = \sqrt{\sum_{k=1}^{n} |x_k|^2} \tag{4.3}$$

and

$$|Y| = \sqrt{\sum_{k=1}^{n} |y_k|^2} \tag{4.4}$$

respectively.

The distance between the two feature descriptors is given by

$$d = |X| - |Y| \tag{4.5}$$

For each feature in the first image, d is calculated for all the features in the second image. The features which returns the least d are considered to be the best match. BF matcher is implemented using OpenCV. The function BFmatcher() returns the best match between two images. Brute force matcher can also be used to return k best matches instead of one match. BFmatcher.knnmatch() returns k best matches in OpenCV. In our implementation two best matches are returned for each feature to perform some tests. Figure 4.3 shows the feature

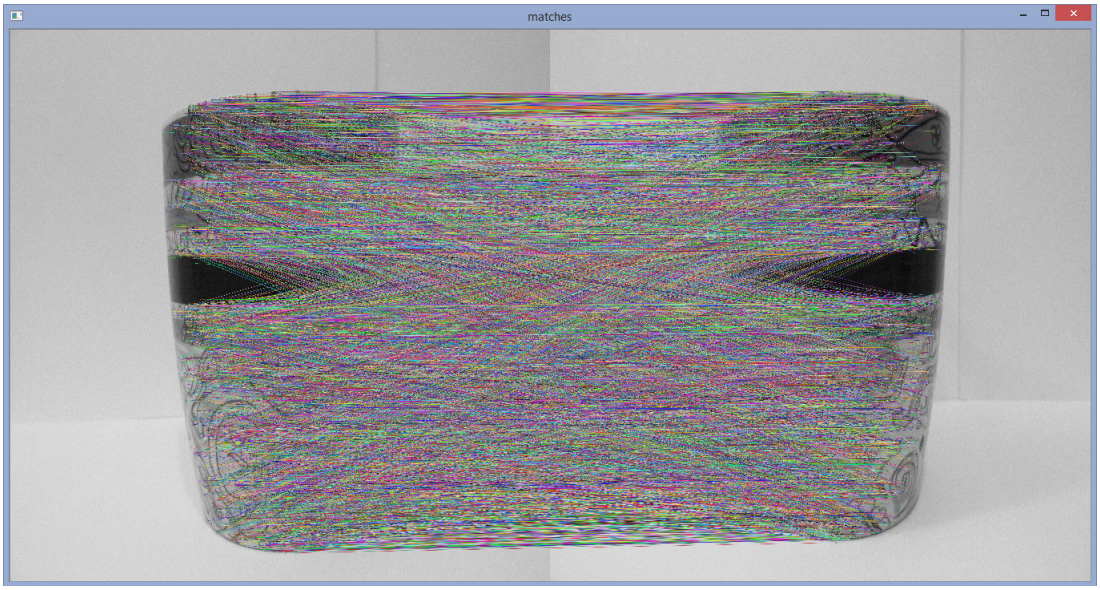

Figure 4.3: Feature Matching using Brute Force matcher

matching between two images. Feature matching is done by Brute Force matcher which returns two best matches for each feature. SURF keypoints and descriptors are used for feature detection and extraction.

4.3.1 Ratio test

As seen in (figure 4.3) Brute Force matcher gives some wrong matches. To eliminate the wrong matches, ratio test is performed. Let d_1 and d_2 be the two best distances which gives the least difference between two feature descriptors, returned by Brute Force knn matcher. The ratio of d_1 and d_2 is calculated.

$$\frac{d_1}{d_2} \leq \tau \tag{4.6}$$

If the ratio is lesser than or equal to the given threshold τ as in (equation 4.6), then the matches are considered to be good. All the other matches can be eliminated. From the(figure4.4) it is clear that ratio test improves the accuracy of the matching between the input images.

4.4 Computation of the fundamental matrix

The epipolar geometry and the Fundamental matrix was explained in detail in Chapter 3. From (equation 3.19) we know that $x'^T F x = 0$. This property is very essential to find the Fundamental matrix from the correspondence points of the images.There are so many methods to find fundamental matrix from the point correspondence between the images. In our implementation we use Automatic computation of F using RANSAC and normalized 8-point algorithm[16].
ALGORITHM

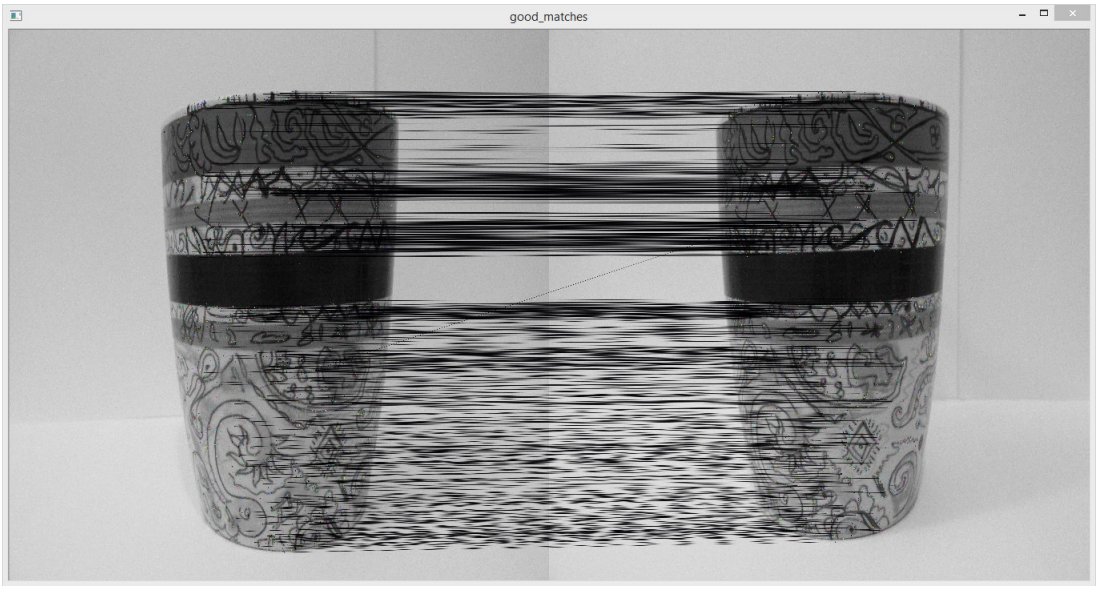

Figure 4.4: Brute Force Matching after ratio test

(i) The initial coarse correspondence found using SURF and Brute Force matcher is given as input for the computation of F.

(ii) A random sample of minimum 8 correspondences are chosen and F is computed using normalized 8-point algorithm.

(iii) The distance $d_\perp$ is calculated for each correspondence pairs. Calculation of $d_\perp$ is explained in section 4.5.

(iv) If the distance $d_\perp$ is lesser than t pixels, then those correspondence pairs are considered as inliers.

(V) The above process is repeated from step (ii) for N samples, where N is determined adaptively which will be explained in detail below.

4.4.1 Normalized 8-point algorithm

The objective is to find F from $x'^T F x = 0$[16].

Where $x = (x, y, 1)$, $x' = (x', y', 1)$ and $F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f32 & f_{33} \end{bmatrix}$. Expanding the above equation, we get

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0 \tag{4.7}$$

For n set of points ,the above equation can be written as a set of linear equations of the form

$$A\hat{F} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \hat{F} = 0 \tag{4.8}$$

Where $\hat{F}$ is the 9×1 vector form of F. F is obtained by solving the equation 4.8, using Singular Value Decomposition. F corresponds to the smallest singular value of A. Before solving the equation, x and x' are normalized as it will improve the stability of the results and conditioning of the problem. Normalization is done by translating and scaling the points so that all the three points in x and x' do not have large difference between them.

ALGORITHM[16]

(i) The image coordinates are transformed to $\tilde{x}_i = Tx_i$ and $\tilde{x}'_i = T'x'_i$, where T and T' are the normalizing transformation matrix.

(ii) F is determined by taking Singular Value decomposition for the equation 4.8.

(iii) F is replaced by $\tilde{F}$ such that it satisfies the singularity constraint $det\tilde{F} = 0$. For details refer to [16].

(iv)F is finally denormalized by setting $F = T'\tilde{F}T$.

4.5 Distance measurement $d_\perp$

For every estimate of F, $d_\perp$ is calculated for all the correspondence pairs. $d_\perp$ is calculated either by using reprojection error or by using Sampson approximation to reprojection error[16]. The distance measures tells us whether the given pair of corresponding points satisfies the epipolar geometry or not.

4.6 Adaptive estimation of number of samples N

Initially N is chosen high to ensure that at least one of the points are free from outliers with the probability p[16]. Let w be the probability that the given points are inliers. Then the probability that the given points are outliers will be $\epsilon = 1 - w$. At least N selections are needed, each consisting of s points. This can be translated mathematically as

$$(1 - w^s)^N = 1 - p \tag{4.9}$$

Thus[16]

$$N = log(1-p)/log(1-(1-\epsilon)^s) \tag{4.10}$$

Initially ϵ is set to 1 so that $N = \infty$ with the probability $p = 0.99$. Some sample correspondence points are chosen and the inliers are counted using $d_\perp$. Once the inliers are counted ϵ is updated by using the formula $\epsilon = (1-w)/(Total \quad number \quad of \quad points)$. The new N is calculated from equation 4.10. This is continued till N samples have been performed[16].

The details of computation of F using RANSAC can be seen in [16]

4.7 Refining fundamental matrix

The accuracy of initial correspondence points plays an important role in determining a good fundamental matrix. If the system of equations (4.8) are solved with erroneous correspondence points, then the Fundamental matrix obtained will not be accurate since it depends only on the correspondence points. Even though ratio test eliminates some wrong matches based on the distance between first two matching for each feature, there might some erroneous matches which satisfies the ratio test because of the wrong matches for each feature being present very close to each other. In our method we have added another test based on epipolar geometry. As we discussed in the previous section, The computation of F using RANSAC eliminates the outliers based on the distance measure. This distance measures tells us whether the correspondence points satisfies the epipolar geometry. Once all the matches that do not satisfy the epipolar geometry are eliminated then the Fundamental matrix is recalculated using the inliers which was obtained from the initial calculation of Fundamental Matrix as input. The algorithm to calculate the final fundamental matrix is as follows

ALGORITHM[16]

(i) The coarse correspondence obtained from SURF is given as input.

(ii) The Initial Fundamental Matrix is calculated as described in section 4.4

(iii)The inliers got from the Fundamental Matrix is taken as new input.

(iv) The Final fundamental matrix is calculated using the same method as above but with inliers as the input correspondence points.

4.8 Computation of essential matrix

The Essential matrix can be considered as the special case of Fundamental Matrix[16]. The fundamental matrix has information about both intrinsic and extrinsic camera parameters. In order to get the extrinsic parameters from the Fundamental matrix, the assumption about calibration matrix K has to be removed from the fundamental matrix F. The Fundamental matrix

without intrinsic parameters is known as Essential matrix. Unlike Fundamental Matrix, the Essential matrix has five degrees of freedom and additional constraints[16]. Once the K matrix is found as described in section 4.1, the inverse of K can be applied to the input points x and x' to remove the effect of intrinsic pare meters. Thus the input coordinates without assumption about intrinsic parameters can be written as

$$\hat{x} = K^{-1}x \tag{4.11}$$

$$\hat{x}' = K^{-1}x' \tag{4.12}$$

The coordinates defined in equations 4.11 and 4.12 are known as *normalized image coordinates* Similarly the *normalized camera matrices* can be defined as

$$P' = K^{-1}K[R|t] \tag{4.13}$$

Thus, the fundamental matrix corresponding to the pair of normalized camera matrices $P = [I|0]$ and $P' = [R|t]$ are known as *Essential Matrix*. Like Fundamental matrix, Essential matrix also has the property

$$\hat{x}'^T E\hat{x} = 0 \tag{4.14}$$

Substituting equation 4.11 and 4.12 in equation 4.14, we get

$$x'^T K'^{-T} E K^{-1} x = 0 \tag{4.15}$$

Comparing equation 4.14 with the relation $x'^T Fx = 0$ for fundamental matrix, we get

$$E = K'^T F K \tag{4.16}$$

The important constraint in Essential matrix is that two of the singular values in Essential matrix should be equal and the third value should be zero[16]. Essential matrix can be easily found once we know Fundamental Matrix and Camera Calibration matrix K.

4.9 Retrieving camera matrices from essential matrix

Let us assume that the first camera matrix is

$$P = [I|0] \tag{4.17}$$

In order to compute the second camera matrix P', the essential matrix has to be factorized to SR[16]. This is due to the additional constraint introduced by Essential matrix as discussed in the previous section. E is decomposed as

$$E = [t]_{\times} R = SR \tag{4.18}$$

For decomposition, let us use two matrices

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.19}$$

where W is the orthogonal matrix.

$$Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.20}$$

where Z is the skew-symmetric matrix.

Let the Singular Value Decomposition of E be $Udiag(1, 1, 0)V^T$. From equation 4.19 and 4.20 the factorization can be defined as

$$S = UZU^T \tag{4.21}$$

and

$$R = UWV^T \quad \text{or} \quad UW^TV^T \tag{4.22}$$

If the signs are ignored then there are two possibilities for R. The translation part t for the camera matrix P' is determined from the factorization (4.21), up to a scale from $S = [t]_{\times}$. Since $St = 0$

$$U(0, 0, 1)^T = u_3 \tag{4.23}$$

which is the third column of U. It is not possible to determine the sign of E, hence t. Therefore for each essential matrix E there are four possible choices for camera matrix P', two for rotation

and two for translation as follows

$$P' = [UWV^T| + u_3] \quad \text{or} \quad [UWV^T| - u_3] \quad \text{or} \quad [UW^TV^T| + u_3] \quad \text{or} \quad [UW^TV^T| - u_3] \quad (4.24)$$

The geometric interpretation of four possible solutions for reconstruction from Essential Matrix E is given in the following figures. Between figure 4.5, 4.7 and 4.6, 4.8 the baseline is reversed. Between figure 4.5, 4.6 and 4.7, 4.8 the camera B is rotated 180 degree about the baseline. It can be seen that, of all the figures, only in figure 4.5 the reconstructed point lies in front of both the cameras. Therefore this is the right solution for P' among all the four possibilities. The camera matrix P' can be found only after reconstructing the 2D points. From the geometric interpretation it is clear that, for the reconstructed points to lie in front of both the cameras the Z coordinate should be positive. In our implementation P' is calculated for all the four possibilities. The number of reconstructed points with positive Z is counted for all the four solutions. The P' matrix which corresponds to the maximum number of Z with positive value

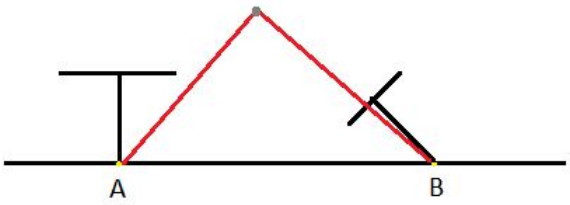

Figure 4.5: First possible reconstruction

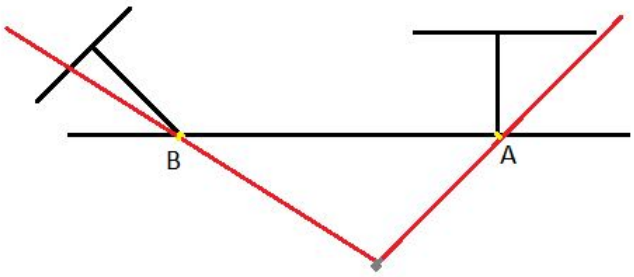

Figure 4.6: Second possible reconstruction

is considered as the second camera matrix. Once the camera matrices P and P' are found, the

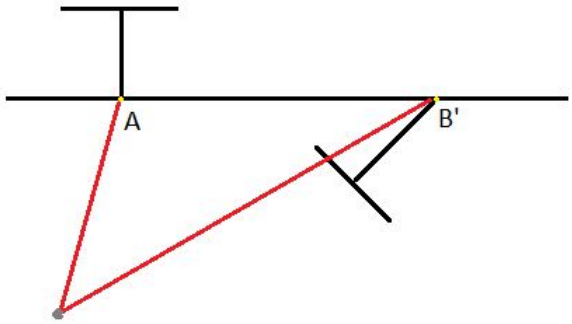

Figure 4.7: Third possible reconstruction

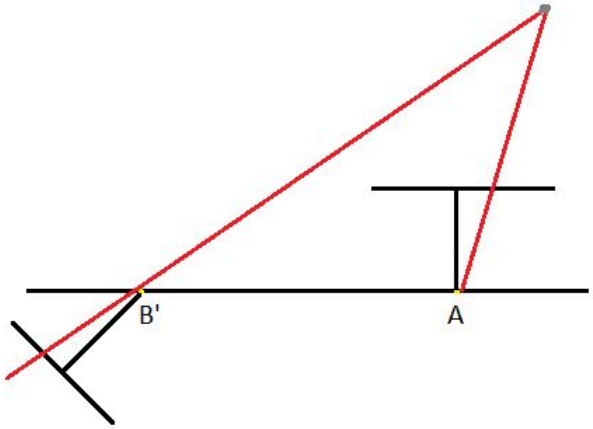

Figure 4.8: Fourth possible reconstruction

next is to triangulate the two points x and x' in the first and second image respectively, to get the 3D point X.

4.10 Triangulation

Triangulation is the final step in 3D reconstruction. Once the camera matrices re found, the relation between 2D points and 3D points can be written as $x = PX$ and $x' = P'X$[16]. Due to some errors in obtaining x and x', there will not be a point X that can exactly satisfy the above two equations. Thus the method for triangulation should be invariant under any transformations. The process of triangulation can be generally represented as

$$X = \tau(x, x', P, P') \tag{4.25}$$

The triangulation can be considered to be invariant under a transformation H if

$$\tau(x, x', P, P') = H^{-1}\tau(x, x', PH^{-1}, P'H^{-1}) \tag{4.26}$$

In our implementation, we have used linear inhomogeneous method to triangulate the 2D points[16].The relation $x = PX$ involves only homogeneous vectors. That is x and PX has the same direction but different magnitude which differs by some non zero scale factor. This relation can be expressed in terms of vector cross product as[16]

$$x \times (PX) = 0 \tag{4.27}$$

Expanding equation 4.27

$$x(p^{3T}X) - (p^{1T}X) = 0 \tag{4.28}$$

$$y(p^{3T}X) - (p^{2T}X) = 0 \tag{4.29}$$

$$x(p^{2T}X) - y(p^{1T}X) = 0 \tag{4.30}$$

where p^{iT} represents the i^{th} row of the matrix P. Equation 4.28 and 4.29 are linearly independent, while equation 4.30 is linearly dependent on equation 4.28. Therefore each point can be represented by just two linear equations 4.28 and 4.29. $x = PX$ and $x' = P'X$ can be written in the form

$$AX = 0 \tag{4.31}$$

where,

$$A = \begin{bmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \\ x'p'^{3T} - p'^{1T} \\ y'p'^{3T} - p'^{2T} \end{bmatrix} \tag{4.32}$$

Equation 4.31 can either be solved by finding the singular vector corresponding to the smallest singular value of A or by solving a set of inhomogeneous equation[16]. By setting $X = (X, Y, Z, 1)^T$, equation 4.31 can be converted to inhomogeneous equations

$$AX = B \tag{4.33}$$

where

$$A = \begin{bmatrix} xp^{31} - p^{11} & xp^{32} - p^{12} & xp^{33} - p^{13} \\ yp^{31} - p^{21} & yp^{32} - p^{22} & yp^{33} - p^{23} \\ xp'^{31} - p'^{11} & xp'^{32} - p'^{12} & xp'^{33} - p'^{13} \\ yp'^{31} - p'^{21} & yp'^{32} - p'^{22} & yp'^{33} - p'^{23} \end{bmatrix} \tag{4.34}$$

$$B = - \begin{bmatrix} xp^{34} - p^{14} \\ yp^{34} - p24 \\ xp'^{34} - p'^{14} \\ yp'^{34} - p'^{24} \end{bmatrix} \tag{4.35}$$

Equation 4.33 can be solved by using singular value decomposition. Thus the 3D points (X,Y,Z) are calculated for each pair of 2D points (x,y) and (x',y')[16].

Chapter 5
Proposed Method for Dense 3D Reconstruction

Once the initial coarse 3D points are found, the next step is to find the dense 3D reconstruction from the two input images. The coarse reconstruction does not provide enough 3D points to describe the scene or an object in detail. Therefore it is necessary to do dense 3D reconstruction of the scene. The first step here is to find the dense correspondence between the input images. This is the most challenging step in 3D reconstruction problem as the accuracy of the 3D points directly depends on the accuracy of the correspondence points. After finding the corresponding points, it is not required to calculate the Fundamental and Camera matrix again. They will remain the same for the given input images. Finally, the dense correspondence points can be triangulated using the F and P matrix calculated in the previous chapter from the initial correspondences to get the dense 3D points. In this thesis, we proposed a new method to find the dense correspondence between the two images using intensity and epipolar geometry based Thin-Plate Spline interpolation.

5.1 Thin-plate splines

In [19] Bookstein proposed a new method for interpolating surface with few initial scattered data over the surface known as *thin-plate spline*. This technique provides a smooth interpolation between a set of points known as control points. The surface is interpolated based on a condition that the surface passes through all the control points with least bending. Thus these control points acts as a position constraints which will lead to less bending. The surface is defined by[19]

$$z(x, y) = -U(r) = -r^2 log r^2 \tag{5.1}$$

where r is the distance $\sqrt{x^2 + y^2}$. The negative sign indicates that the surface is slightly dented in this pose. Positive sign indicates that the surface is slightly convex in this pose.

In this thesis, thin-plate spline is used for finding the dense correspondence between the images. We already have the initial correspondence points between the images which was calculated from SURF and epipolar geometry. These initial correspondence points are considered as the control points. In order to find the mapping between the images. If (x, y) is the position of a pixel in the first image then after moving the camera the new position of the corresponding

pixel should be $(x + dx, y + dy)$ in the second image. Thus for each pixel in the first image, the displacement (dx, dy) has to be found to get the corresponding pixel position in the second image. The surface which is least bent is given by

$$f_y(x, y) = a_1 + a_{xy}x + a_{yy}y + \sum_{i=1}^{n} w_{iy}U(|P_{iy} - (y)|) \tag{5.2}$$

where the first three terms can be considered as the least square fitting term that defines the plane which best fits all the control points[19]. The last term provides the bending energy which are given by n control points. P_i is the set of control points in the first image. (x, y) is the point in the first image for which the corresponding point has to be found in the second image. w_i is the weight associated with the input control points. Before finding the mapping, the initial step is to find all the weights w_i for the given control points. Once the weights are calculated then all the values can be given to (5.2) to get the mapping separately in x and y direction.To calculate the weights, let us first define the following matrices

$$K = \begin{bmatrix} 0 & U(r_{12}) & ... & U(r_{1n}) \\ U(r_{21}) & 0 & ... & U(r_{2n}) \\ ... & ... & ... & ... \\ U(r_{n1}) & U(r_{n2}) & ... & 0 \end{bmatrix} \tag{5.3}$$

where K is a $n \times n$ matrix.

$$P = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ ... & ... & ... \\ 1 & x_n & y_n \end{bmatrix} \tag{5.4}$$

where P is a $3 \times n$ matrix.

$$L = \begin{bmatrix} K & | & P \\ P^T & | & O \end{bmatrix} \tag{5.5}$$

where L is a $(n + 3) \times (n + 3)$ matrix. O is a 3×3 matrix of Zeros. The initial correspondence points in the second image can be compiled together in the form of matrix as

$$V = \begin{bmatrix} x'_1 & x'_2 & ... & x'_n \\ y'_1 & y'_2 & ... & y'_n \end{bmatrix} \tag{5.6}$$

Some zeros are appended at the end of the V matrix to match the dimensions with L matrix. Let new matrix Y be

$$Y = (V|0 \quad 0 \quad 0)^T \tag{5.7}$$

Let the weights $W = (w_1, w_2, ..., w_n)$ and the coefficients a_1, a_x, a_y together be defined as $(W|a_1 \quad a_x \quad a_y)^T$.Thus the relation to calculate the weights and the coefficients is

$$L^{-1}Y = (W|a_1 \quad a_x \quad a_y)^T \tag{5.8}$$

The details of thin-plate spline interpolation can be seen in [19]. Equation (5.8) can be solved by LU Decomposition. Once the weights and coefficients are calculated, they can be substituted in (5.2) to get the displacement (d_x, d_y) for each point (x, y) in the the first image. In this way Thin-Plate Spline can be used to find the dense correspondence between two images.

5.2 Modified thin-plate spline interpolation

It is challenging to get highly dense reconstructed points from just two views of the image. Usually images of the same scene are taken from multiple views to get high correspondence points between the images which will in turn give dense reconstructed points. Thin plate spline interpolation gives good mapping for each pixel in one image to find the corresponding point in the other image.But, the interpolation of the surface is purely based on the initial control points. In this thesis, the initial control points obtained from the SURF features and descriptors are in the range of thousands, while the final dense correspondence points between the images are expected to be in the range of millions. Since the ratio of the final and initial correspondence points are in the range of thousands, the final matching points obtained from thin-plate spline are not very accurate. Therefore, in this thesis we propose a new method to improve the accuracy of dense image matching using two additional constraints.

The first constraint is based on the epipolar geometry between the two views. The Fundamental matrix has already been calculated for the two input images to find the coarse 3D reconstructed points. From equation 3.19, the condition for the two correspondence points to satisfy the epipolar constraint is $x'^T F x = 0$. This condition is used to check whether the matching points obtained from the thin-plate spline interpolation have satisfied the epipolar geometry. Practically $x'^T F x$ will not be equal to zero. Therefore this constraint is checked for a particular threshold range between t_u and t_l. If

$$t_l \leq x'^T F x \leq t_u \tag{5.9}$$

then the correspondence points are considered for 3D reconstruction. The upper and lower thresholds are chosen adaptively depending on how much error can be tolerated for each picture. The reason for choosing the threshold adaptively is that different images might have different range of thresholds to get dense correspondence points. If the threshold ranges are very close to zero then the reprojection error will be very low and it increases as the threshold range moves away from the ideal value zero. To find the lower and upper threshold, bins of width 0.025 are created between the range -1 and 1 with the initial value of zero in each bin. The value of $x'^T Fx$ is calculated for each correspondence points. If the value of $x'^T Fx$ falls within the range of any particular bin, then the corresponding bin is incremented by one. This gives the frequency distribution of the values of epipolar constraint between -1 and 1. The range of the thresholds can be chosen depending on the percentage of dense correspondence points required with respect to the input images. There is a trade off between the number of matching points and the reprojection error. If the constraints are kept closer to Zero, then the number of dense correspondence points obtained will be lesser but the reprojection error will be lower. On the other hand if the threshold ranges are chosen little away from zero, the number of dense correspondence points obtained will be higher but the reprojection error will also be increased. The users can change the percentage of correspondence points required for reconstruction depending on the percentage of error that can be tolerated for different applications.

The Second constraint is based on the intensity variation between the two images. The correspondence points which have same intensity values are considered for the 3D reconstruction. Practically, there might be a minor variation in the intensity values between two views due to the change in pose and the effect of illumination accordingly. Therefore, the difference in intensities less than a particular threshold is chosen as the second constraint to get the matching points for the reconstruction. Let I_1 be the intensity value of the pixel in the first image and I_2 be the intensity value of the image in the second image. Let ΔI be the difference in intensities of each pixels between the two images.

$$\Delta I = I_1 - I_2 \tag{5.10}$$

If the difference in intensities of each correspondence points is less than the average value of the intensity difference between the two images, then the corresponding points are considered for reconstruction. If

$$|\Delta I| < Average(\sum_{i=1}^{n} |\Delta I_i|) \tag{5.11}$$

where, n is the total number of correspondence points.

Adding these two constraints to the output of image matching points obtained from thin-plate spline interpolation have improved the accuracy of the dense 3D reconstruction.

Chapter 6
Results

In this thesis, Canon PowerShot SX170 IS is used to take all the images. The resolution of the image is 3456×4608. The camera is first calibrated to get the intrinsic parameters. The coarse 3D reconstruction is done for the given two input images. The extrinsic parameters and the correspondence points calculated from the coarse 3D reconstruction are used for dense 3D reconstruction.

6.1 Camera calibration

Camera calibration is done by using OpenCV [17]. Eight snapshots of the checkerboard pattern are taken in different positions as shown in figure 6.1. All the eight images are taken at a constant focal length.The number of inner corners of the square in the checkerboard and the size of the square in the checkerboard are given as input. In the checker board pattern used for this thesis, the number of inner corners are 9×6 and the size of the square is 50 pixels.The program gives the internal calibration matrix K.

$$K = \begin{bmatrix} 3.5595342957150860 \times 10^3 & 0 & 2.3035000000000000 \times 10^3 \\ 0 & 3.5595342957150860 \times 10^3 & 1.7275000000000000 \times 10^3 \\ 0 & 0 & 1 \end{bmatrix}$$

It is sufficient to calibrate the camera once for different inputs unless the focal length is changed. The same focal length is maintained throughout for taking all the results.

6.2 Coarse 3D reconstruction

Once the camera is calibrated, two images of the same scene are taken at different positions by moving the camera with the same focal length used for camera calibration. The two input images are shown in figure 6.2 and 6.3. Features are extracted using SURF and matched using Brute Force Matcher.

Number of features extracted from the first image = 6839

Number of Features extracted from the second image = 7138

Number of matches found using Brute Force Matcher = 6839

Number of matches found after the Ratio test = 860

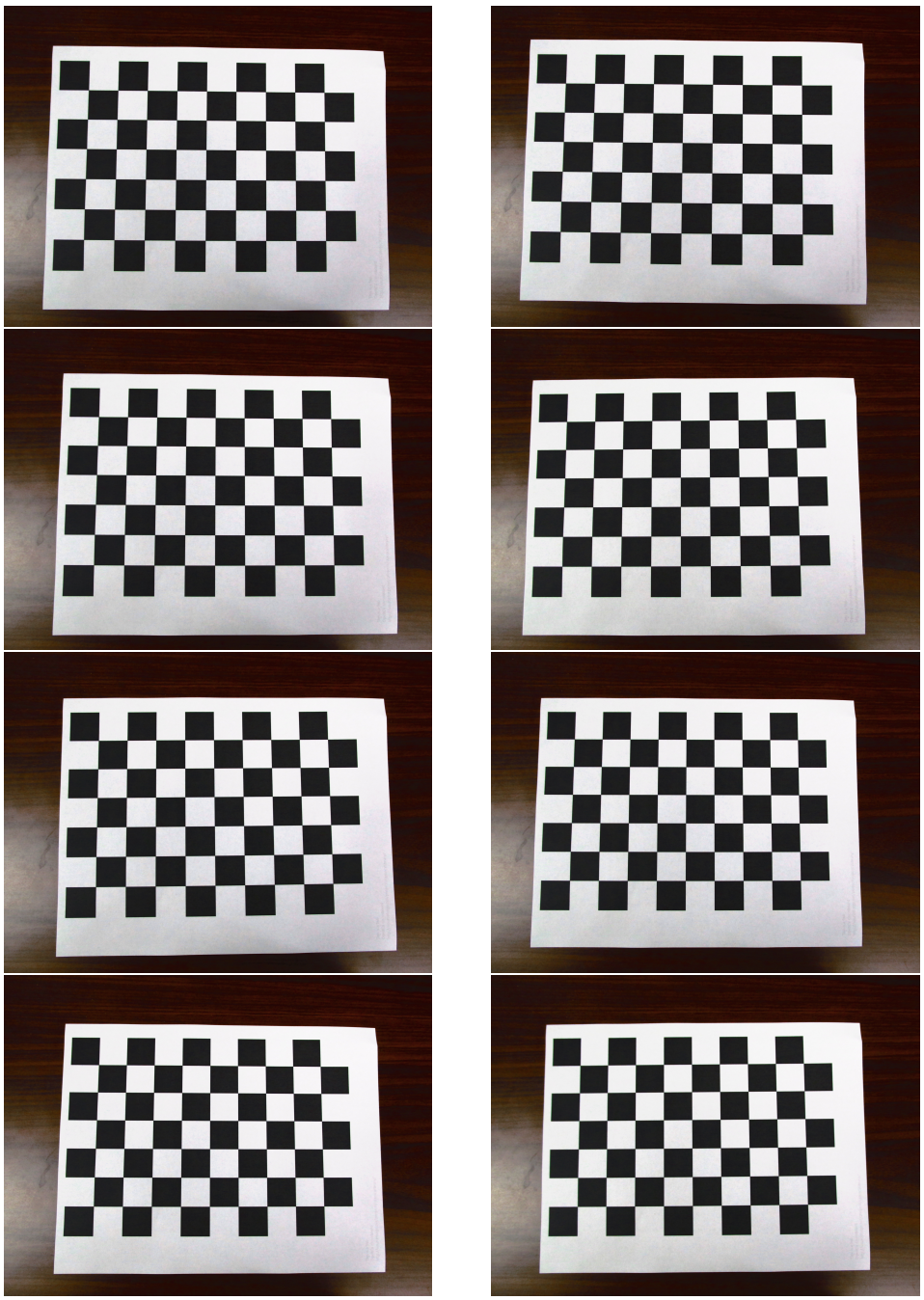

Figure 6.1: Input Checkerboard pattern at different positions

Figure 6.2: Input image at the first position

Figure 6.3: Input image at the second position

Initial Fundamental Matrix obtained from the 860 correspondence matches after ratio test:

$$F = \begin{bmatrix} -2.772436229134866 \times 10^{-9} & 4.482524822130676 \times 10^{-7} & -0.0005783909327605752 \\ -3.105722919622621 \times 10^{-7} & 1.6045291879357 \times 10^{-8} & -0.008948060348659193 \\ 0.0003577196703114516 & 0.00834701170539276 & 1 \end{bmatrix}$$

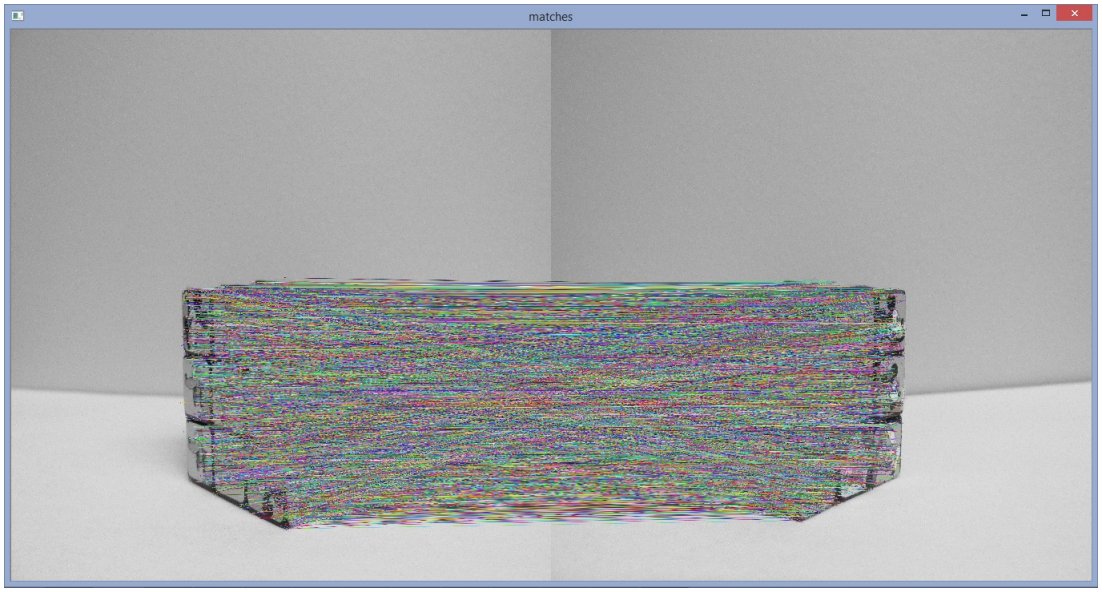

Figure 6.4: Initial matches from Brute Force Matcher

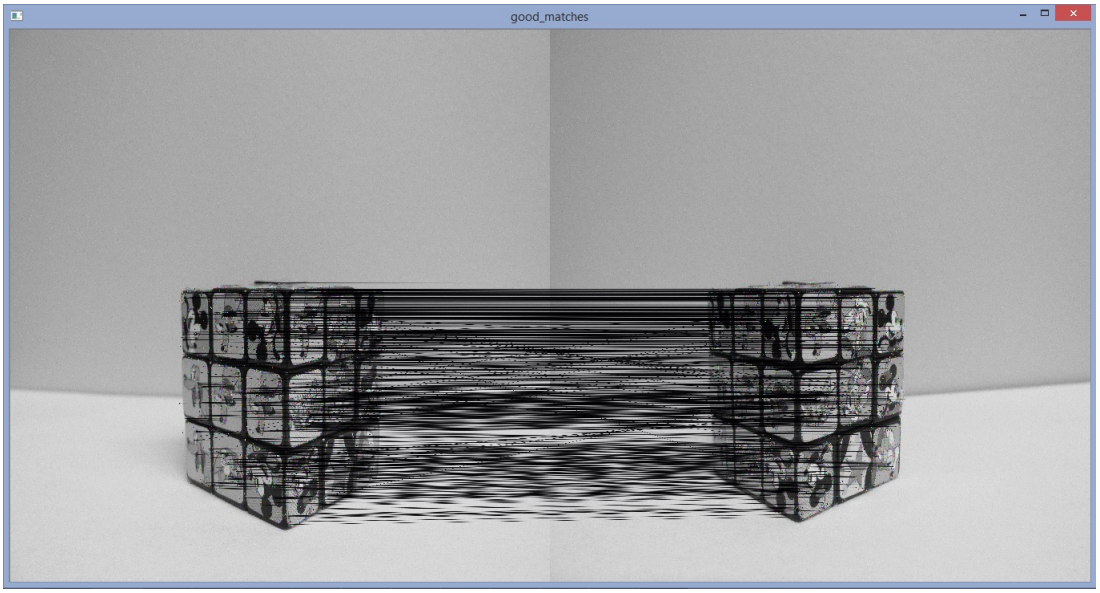

Figure 6.5: Matches after ratio test

Number of matches obtained from Fundamental matrix using RANSAC = 465

Final Fundamental matrix obtained from 465 correspondence matches:

$$F1 = \begin{bmatrix} -2.607874 \times 10^{-9} & 4.095421 \times 10^{-7} & -0.000536 \\ -2.570963 \times 10^{-7} & 8.274342 \times 10^{-9} & -0.008689 \\ 0.000292 & 0.008094 & \end{bmatrix}$$

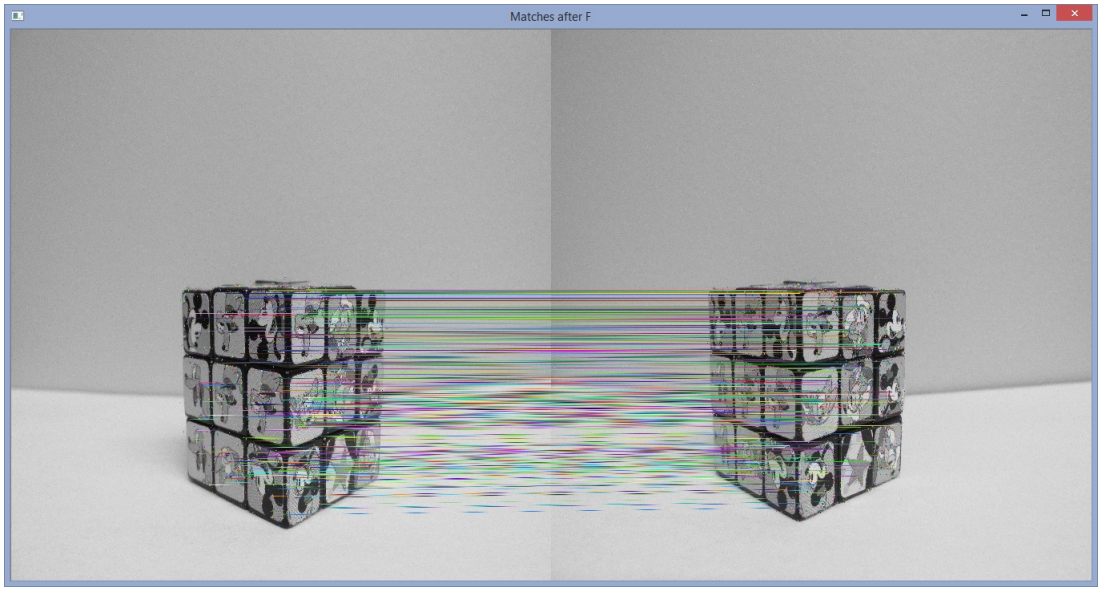

Figure 6.6: Final matches found from Fundamental Matrix using RANSAC

The camera matrix P:

$$P = \begin{bmatrix} -0.998122 & 0.000432 & -0.061262 & 0.987120 \\ -0.000274 & -0.999997 & -0.002584 & 0.017525 \\ 0.061263 & 0.002563 & -0.998118 & -0.159021 \end{bmatrix}$$

The epilines for the correspondence points in both the images are given inf figure 6.7 and 6.8.

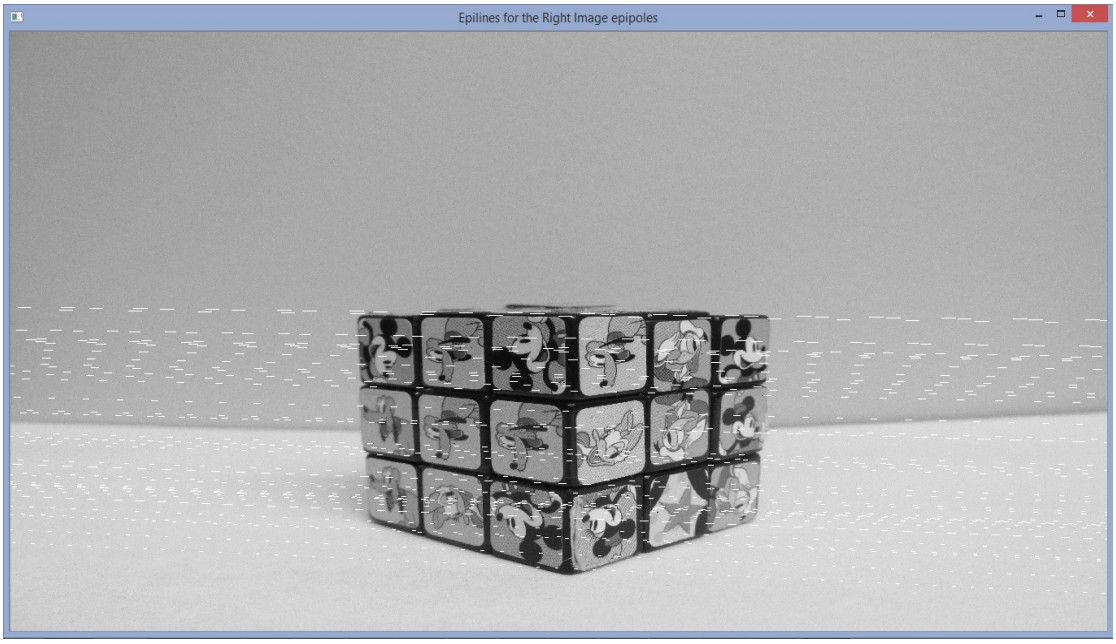

Figure 6.7: Epilines in the first image for the corresponding points in the second image

Figure 6.8: Epilines in the second image for the corresponding points in the first image

The triangulated 3D points from the 2D images are rendered using a tool called Plotviz [27]. The accuracy of the correspondence points between the images greatly affect the 3D reconstructed points. The 3D reconstructed points are given for the input correspondence matches obtained before and after refining the matches using Fundamental matrix with RANSAC to show visually how improving the accuracy of image matching points improves the results of 3D reconstructed points.

Figure 6.9 shows the four different views of the 3D reconstructed points which is reconstructed from the correspondence point matches before refining the matches using Fundamental matrix with RANSAC. It can be seen that there are many wrong points in 3D around the cube. Figure 6.10 shows the four different views of final 3D reconstructed points which is reconstructed from the correspondence point matches after refining them using Fundamental matrix with RANSAC. It is clear from the figure 6.10 that there are no wrong points in the reconstructed image. This eliminates the need to adjust the 3D points after reconstruction. The Reprojection error is calculated to measure the performance of the system.

CALCULATING THE REPROJECTION ERROR

Let (x_{Oi}, y_{Oi}) represent the 2D points of either of the input correspondence matches between two images. Let (x_{Ri}, y_{Ri}) be the 2D points calculated by reprojecting the reconstructed 3D

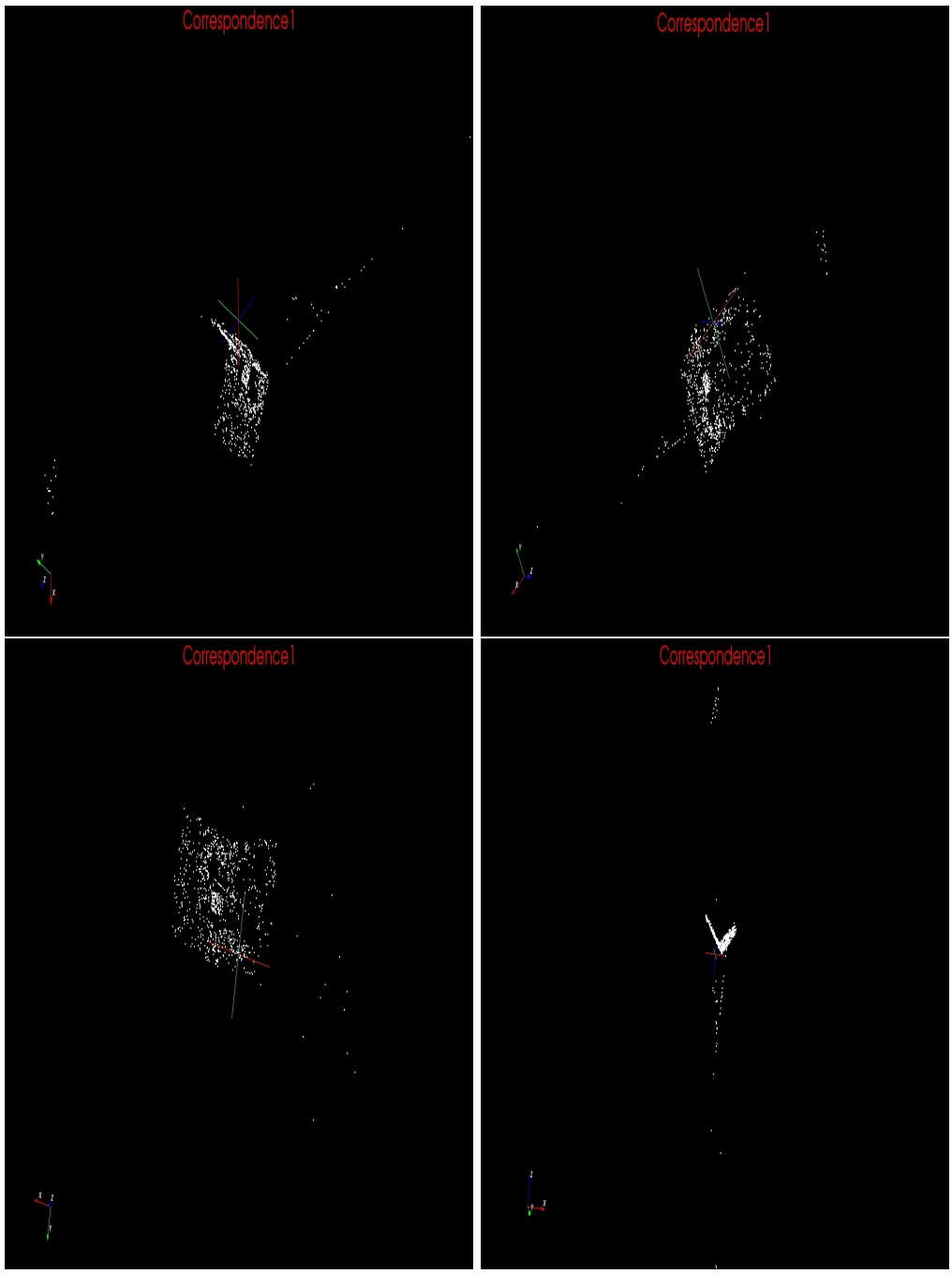

Figure 6.9: Four different views of 3D Reconstructed points of the cube before refining the image matching points

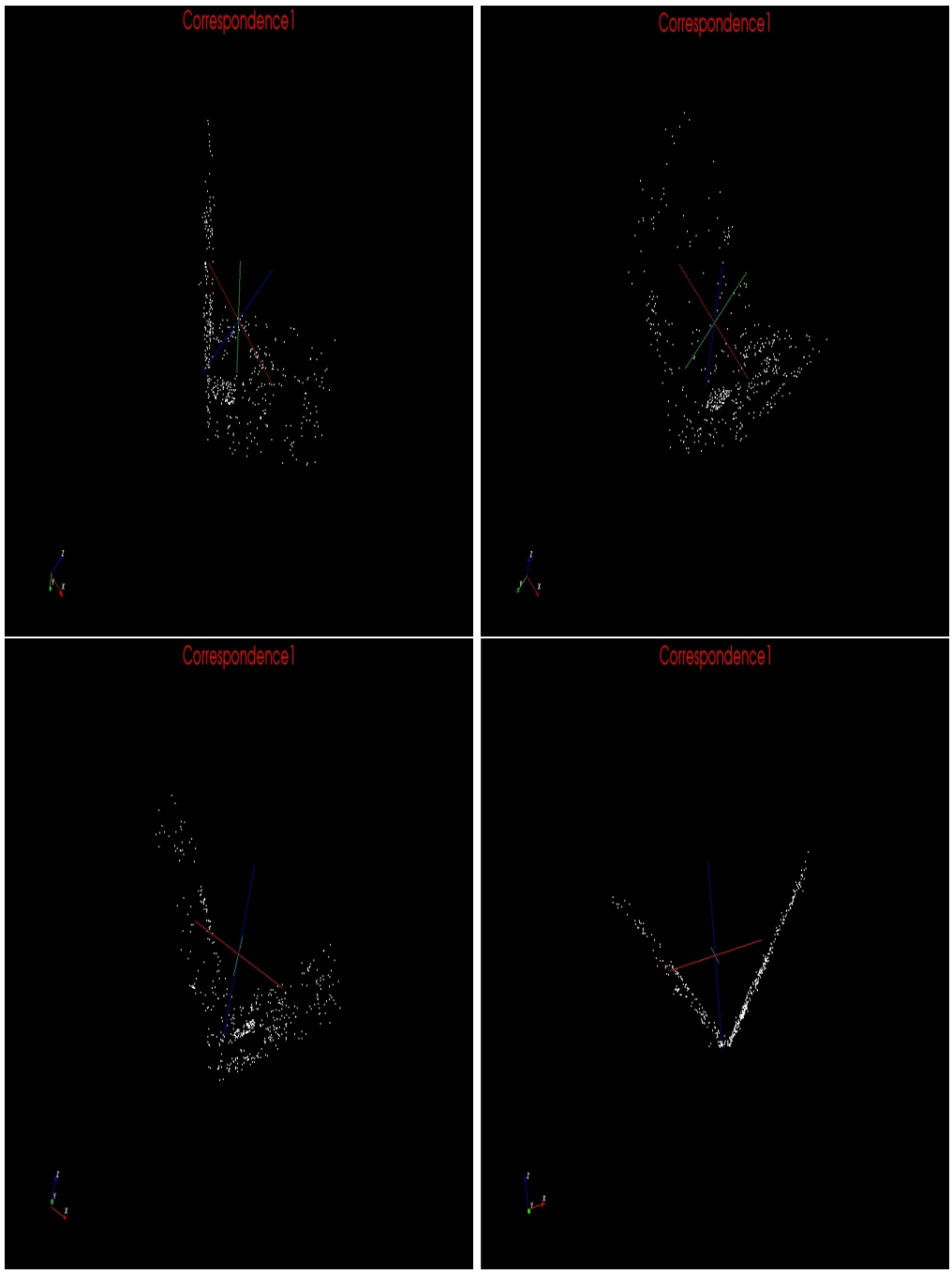

Figure 6.10: Four different views of 3D Reconstructed points of the cube after refining the image matching points

points using (1.7). Then the average reprojection error ϵ is given by

$$\epsilon = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{(x_{Ri}-x_{Oi})^2+(y_{Ri}-y_{Oi})^2}{x_{Ri}^2+y_{Ri}^2}} \tag{6.1}$$

where n is the total number of correspondence points. The average reprojection error is calculated for the reconstruction before and after refining the matches.

Table 6.1: Average Reprojection Error

Method	Average Reprojection error
Before refining the matches	0.01
After refining the matches	0.002

From table 6.1 it can be seen that the reprojection error has been improved after refining the correspondence matches using Fundamental matrix with RANSAC by eliminating the outliers that does not satisfy the epipolar geometry.

6.3 Dense 3D reconstruction

Dense image correspondence between the images are obtained using thin-plate spline interpolation. This is implemented using C++ and OpenCV. Once the corresponding image matches are found,the performance of feature matching is improved by adding the epipolar constraint. The threshold range for the epipolar constraint are found adaptively using the frequency distribution of deviation of epipolar constraint.For the two input images,

The total number of correspondence matches found using thin-plate spline = 29,14,303

The frequency distribution of deviation of epipolar constraint is given in figure 6.11. The epipolar constraint $x'^T Fx$ is calculated for all the correspondence points obtained from thin-plate spline. The value of the epipolar constraint is taken between -1 and 1 with a bin width of 0.05. The number of correspondence matches whose epipolar constraint value falls between -1 and 1 is plotted as shown in figure 6.11. Any value deviating from zero is considered as error. Since it is not practically possible to have the constraint as zero, a small deviation from zero can be considered for reconstruction. The more the value deviates, the higher is the reprojection error. The user can decide how much deviation to be considered depending on the number of satisfying points required to reconstruct the structure of the object. Based on the percentage of correspondence points chosen by the user, the program calculates the lower and upper threshold range for the epipolar constraint. For the input cube images, 75 percent of the total dense correspondence points is given as input.

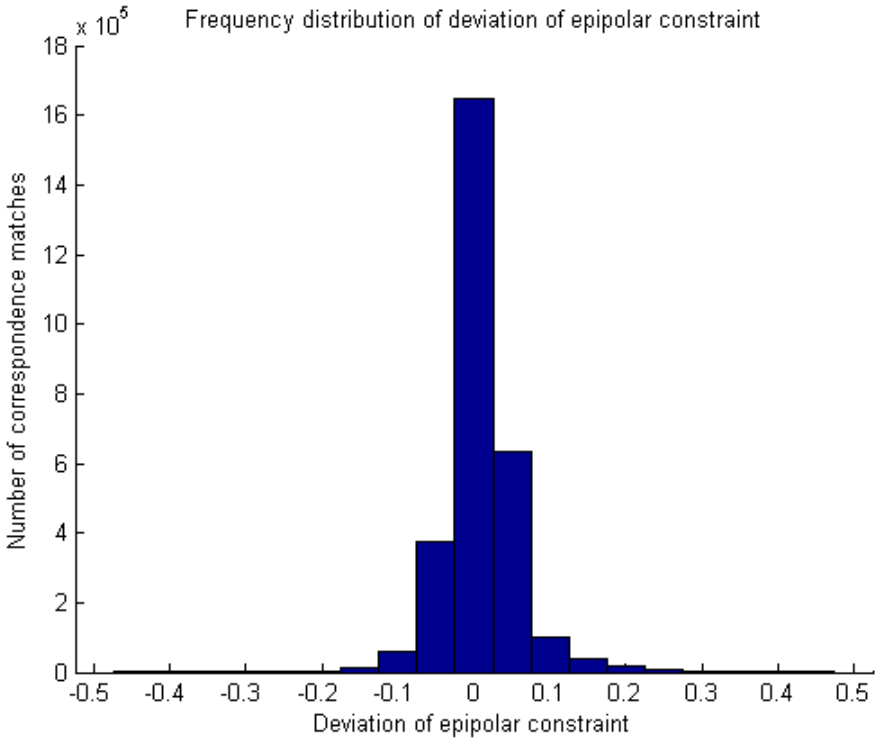

Figure 6.11: Epilines in the first image for the corresponding points in the second image

- The lower threshold = -0.025
- The upper threshold = 0.075
- Thus the correspondence points, $-0.025 \leq x'^T F x \leq 0.075$ are considered for 3D reconstruction.
- The total number of correspondence matches found after applying epipolar constraint = 22,85,824
- The constraint based on intensity is applied using (5.11).
- The total number of correspondence matches found after applying intensity constraint = 16,64,267
- The reprojection error after applying intensity and epipolar constraint = 0.0020698

Different views of dense 3D reconstruction of the cube is given below. The 3D points are rendered using Meshlab [26].

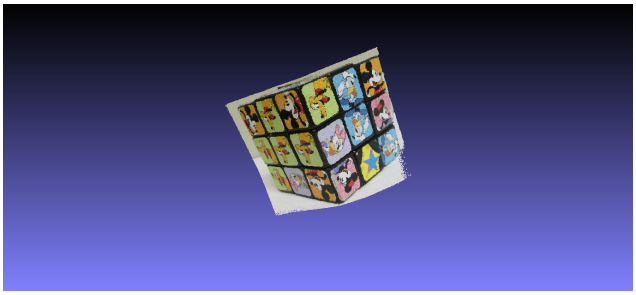

Figure 6.12: Different views of 3D reconstruction of cube

Figure 6.13: Different views of 3D reconstruction of cube

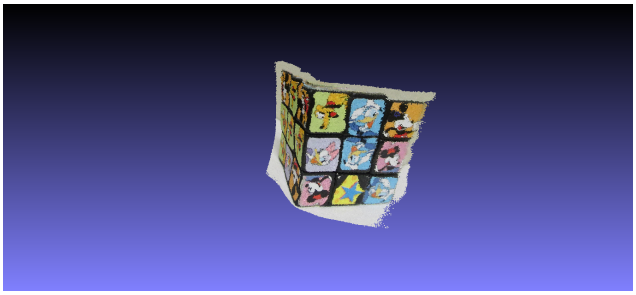

Figure 6.14: Different views of 3D reconstruction of cube

Figure 6.15: Different views of 3D reconstruction of cube

6.4 Systematic evaluation of 3D reconstruction system

The dense 3D reconstruction system is evaluated using three methods as explained below.

6.4.1 Texturing the 3D points

The 3D points are colored/textured using the intensity values from the two 2D images. To verify the correctness of the dense correspondence points visually, the intensity values used to color the dense 3D points are obtained by mixing equal half of the intensity values from both the images. This is given by

$$I_f = (0.5 \times I_1) + (0.5 \times I_2) \tag{6.2}$$

where I_f is the intensity value used to color the 3D points. I_1 and I_2 are the intensity values of each pixels in the first and second image respectively.

Figure 6.16: Input images for texturing

Figure 6.17: Textured 3D points obtained by taking 50 percentage of intensity values from both the input images

It can be seen from Figure 6.17 that, the textured 3D points resembles the original input images in spite of mixing the intensity values from both the images.

6.4.2 Reprojection error

The reprojection error is calculated for dense 3D reconstruction which is similar to the calculation done in coarse 3D reconstruction. Let (x_{Oi}, y_{Oi}) represent the 2D points of either of the input correspondence matches between two images. Let (x_{Ri}, y_{Ri}) be the 2D points calculated by reprojecting the reconstructed 3D points using (1.7). Then the average reprojection error ϵ is given by

$$\epsilon = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{(x_{Ri}-x_{Oi})^2+(y_{Ri}-y_{Oi})^2}{x_{Ri}^2+y_{Ri}^2}} \tag{6.3}$$

where n is the total number of correspondence points. The average reprojection error for the reconstruction of cube given in Figure 6.16 = 0.002

6.4.3 Comparison between real object and simulated measurement

The simulated 3D model is measured and compared with the real object measurement. The distance between different points are measured in real object as well as the simulated model. If both the measurements match each other, then the 3D reconstruction system is said to be accurate. For the cube object mentioned above, the distance between the top and bottom edge is measured at various places in both real and simulated model. Since the cube has equal dimension at all corners, this is verified by calculating the distance between the edges in all corners. In figure

Figure 6.18: Measurement of the reconstructed model

6.18, different lines shows the distance measure between two points considered for comparison with real objects. It can be seen from figure 6.19 that the real object and simulated measurement match each other. The overall error in measurement = 0.001

Distance between the points	X_1	X_2	Y_1	Y_2	Z_1	Z_2	Real object measurement	Simulated measurement
0 - 1	0.139	0.079	-0.201	2.947	7.443	7.617	3.15±0.001	3.153
2 - 3	-1.026	-1.153	-0.158	2.984	8.513	8.751	3.15±0.001	3.153
4 - 5	-1.930	-1.980	-0.194	2.956	9.361	9.499	3.15±0.001	3.153
6 - 7	1.240	1.210	-0.204	2.948	8.329	8.411	3.15±0.001	3.153
8 - 9	1.931	1.926	-0.230	2.920	9.035	9.181	3.15±0.001	3.153
10 - 11	0.513	0.482	0.831	1.883	7.799	7.929	1.05±0.001	1.060
12 - 13	-0.316	-0.357	0.849	1.904	7.993	8.088	1.05±0.001	1.060

Figure 6.19: Comparison table between real object and simulated measurement

6.5 Conclusion

The coarse and dense 3D reconstruction of a scene from 2D images is implemented using C++ and OpenCV. The accuracy of the correspondence points found using SURF is improved by adding a constraint using fundamental matrix. This improved the accuracy of the coarse 3D reconstruction which id shown by calculating the reprojection error. Dense 3D correspondence points id found using thin-plate spline interpolation and its performance is improved using intensity and epipolar geometry based constraints. The reprojection error of the dense 3D reconstruction system is calculated and compared with the reconstruction done using thin-plate spline correspondence. Adding additional constraints improved the accuracy of the reconstructed points. Systematic evaluation of 3D reconstruction is done by texturing, calculating reprojection error and comparing the real object and simulate measurement. One drawback of using thin-plate spline interpolation for dense correspondence matches is that the final dense point matches depends on the initial control points. If the initial control points does not have enough feature points all over the object, then the interpolation might not be good. Therefore the future work can be focused on finding a good geometry or feature based matching that will give uniform initial correspondence matches for the object.

References

[1] Moons, T., Van Gool, L., & Vergauwen, M. (2009). *3d Reconstruction from Multiple Images: Part 1: Principles*. Now Publishers Inc.

[2] Faugeras, O. D., & Toscani, G. (1986, June). The calibration problem for stereo. In *Proceedings of the IEEE Conference on Computer Vision* and *Pattern Recognition* (Vol. 86, pp. 15-20).

[3] Tsai, R. Y. (1986). An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1986*.

[4] Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *Robotics and Automation, IEEE Journal of, 3*(4), 323-344.

[5] Maybank, S. J.,& Faugeras, O. D. (1992). A theory of self-calibration of a moving camera. *International Journal of Computer Vision, 8*(2), 123-151.

[6] Luong, Q. T., & Faugeras, O. D. (1997). Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of computer vision, 22*(3), 261-289.

[7] Hartley, R. I. (1994, June). An algorithm for self calibration from several views. In *CVPR* (Vol. 94, pp. 908-912).

[8] Caprile, B., & Torre, V. (1990). Using vanishing points for camera calibration. *International journal of computer vision, 4*(2), 127-139.

[9] Hartley, R. I. (1994). Self-calibration from multiple views with a rotating camera. In *Computer VisionECCV'94* (pp. 471-478). Springer Berlin Heidelberg.

[10] Moravec, H. P. (1981, August). Rover Visual Obstacle Avoidance. In *IJCAI* (pp. 785-790).

[11] Harris, C., & Stephens, M. (1988, August). A combined corner and edge detector. In *Alvey vision conference* (Vol. 15, p. 50).

[12] Zhang, Z., Deriche, R., Faugeras, O., & Luong, Q. T. (1995). A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence, 78*(1), 87-119.

[13] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision, 60*(2), 91-110.

[14] Zhang, Z. (2000). A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22* (11), 1330-1334.

[15] Faugeras, O. D., Luong, Q. T., & Maybank, S. J. (1992, January). Camera self-calibration: Theory and experiments. In *Computer VisionECCV'92*(pp. 321-334). Springer Berlin Heidelberg.

[16] Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

[17] Bradski, G. (2000). The opencv library. *Doctor Dobbs Journal, 25*(11), 120-126.

[18] Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer VisionECCV 2006* (pp. 404-417). Springer Berlin Heidelberg.

[19] Bookstein, F. L. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence, 11*(6), 567-585.

[20] Chen, P., & Li, X. (2014). Effective Volumetric Feature Modeling and Coarse Correspondence via Improved 3DSIFT and Spectral Matching. *Mathematical Problems in Engineering, 2014*.

[21] Iyengar, S. S., Li, X., Xu, H., Mukhopadhyay, S., Balakrishnan, N., Sawant, A., & Iyengar, P. (2012). Toward More Precise Radiotherapy Treatment of Lung Tumors. *IEEE Computer, 45*(1), 59-65.

[22] Xu, H., & Li, X. (2013). A Symmetric 4D Registration Algorithm for Respiratory Motion Modeling. In*Medical Image Computing and Computer-Assisted InterventionMICCAI 2013* (pp. 149-156). Springer Berlin Heidelberg.

[23] Xu, H., & Li, X. (2013, April). Consistent feature-aligned 4d image registration for respiratory motion modeling. In *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on* (pp. 584-587). IEEE.

[24] Xu, H., Chen, P., Yu, W., Sawant, A., Iyengar, S. S., & Li, X. (2012, November). Feature-aligned 4D spatiotemporal image registration. In *Pattern Recognition (ICPR), 2012 21st International Conference on* (pp. 2639-2642). IEEE.

[25] Li, X., & Iyengar, S. S. (2014). On Computing Mapping of 3D Objects: A Survey. *ACM Computing Surveys (CSUR), 47*(2), 34.

[26] Cignoni, P., Corsini, M., & Ranzuglia, G. (2008). Meshlab: an open-source 3d mesh processing system. *Ercim news, 73,* 45-46.

[27] Bae, S. H., Choi, J. Y., Qiu, J., & Fox, G. C. (2010, June). Dimension reduction and visualization of large high-dimensional data via interpolation. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing* (pp. 203-214). ACM.

Vita

Padmapriya Ravi was born in Chennai,India in 1991 to Ravi Kuppuswamy and Shanthi Ravi. She got her Bachelor of Engineering in Electronics and Communications Engineering from Anna University,Chennai,India in the year 2012. She joined LSU in fall 2012 to pursue a Masters degree in Electrical and Computer engineering.