Affine Reconstruction from multiple views using Singular Value Decomposition

Mohan Obeysekera

This report is submitted as partial fulfilment
of the requirements for the Masters Programme of the
School of Computer Science and Software Engineering,
The University of Western Australia,
2003

Abstract

Reconstructing three dimensional (3D) object shape from multiple views is a widely researched area in computer vision. There are many methods available for 3D reconstruction using different image types.

An image with perspective projection gives us the ability to understand and interpret features in the 3D world. In a perspective image, parallel lines in the 3D world appear to converge to a vanishing point, providing information about the third dimension. An affine image has orthographic projection to preserve parallelism such that when parallel lines in the 3D world are projected on to the image plane, they remain parallel. In this thesis, I explain the reconstruction of 3D affine structure from multiple affine images using the Singular Value Decomposition (SVD).

I use multiple affine images of the object ensuring that all images cover every feature in the object to be reconstructed. This poses a restriction on the reconstruction, since acquiring the complete object is not feasible. Hence, the reconstruction involves only the affine structure that is projected on the affine images. I identify feature points on affine images firstly, using a manual mechanism and later, using a tracking system. A measurement matrix is composed using the image points for all affine images. This matrix forms the basis for acquiring 3D affine coordinates.

The SVD decomposes a given matrix into three matrices: two orthogonal matrices and one singular matrix. I use the SVD to decompose the measurement matrix into three matrices to provide two solutions for the 3D affine coordinates. I employ a mechanism to form the metric structure from the two affine structures by using prior knowledge about the object in the 3D world.

I present my results using synthetic and real images. I analyse the results primarily on synthetic images and then use real images to examine the practicality of the method. In addition to using images with orthographic projection, I examine the behaviour of the reconstruction algorithm in the presence of noise, and images with perspective projection.

Keywords: Orthographic projection, factorization algorithm, singular value decomposition
CR Categories: I.2.10 Vision and Scene Understanding, I.4.5 Reconstruction

Acknowledgements

My sincere thanks go to all my mates who encouraged me with their handy hints and supportive ideas. It was tricky having a summer break with a project at the back of my mind...! But constant probing by my mates kept me on the track.

Above all, I am grateful to my supervisor, Dr Peter Kovesi, for always willing to discuss the subject matter, and providing me with invaluable advice and guidance throughout this project.

Contents

List of Figures

CHAPTER 1

Introduction

Three dimensional (3D) reconstruction of objects from multiple views is an ongoing research area in the field of Computer Vision [3, 2]. We interpret depth using various visual cues to understand the third dimension of an object, in the 3D world. However, for a given two dimensional (2D) image, we have the ability to visualise the third dimension through information on perspective projection from the image. The interest lies in the process of gathering this 2D image data and processing it to create the 3D structure. Hence, 3D reconstruction involves the use of techniques in computer vision to add the missing dimension to create the 3D space from 2D images.

There exists numerous methods to reconstruct 3D objects: stereo imaging focuses on 3D reconstruction by using two images [9]; single view reconstruction involves building the 3D structure using one view of the original image from perspective cues; single axis rotation adapts a method to create an object on a turntable by acquiring its 3D information from all angles. Another method known as *affine reconstruction* involves the use of images obtained from affine cameras to recreate the affine object space [3]. In this thesis, I discuss the approach to affine reconstruction from multiple affine images to rebuild the affine object space.

A camera maps 3D coordinates in real world to a 2D image plane. The general projective camera models central projection on to an image plane. Specializations of the general projective camera are classed into two categories. Cameras with a finite centre having a fixed focal length and cameras with centre at infinity [3]. Furthermore, cameras with centre at infinity fall into two groups: affine cameras and non-affine cameras. An affine camera maps points at infinity in 3D space to points at infinity in the 2D image plane. This implies that, for example, a set of parallel lines in world coordinates will remain parallel when the lines are projected on to the image plane. Therefore, the affine image of the object removes perspective projection and maintains an orthographic projection to preserve parallelism between planes and lines. A non-affine camera does not map points at infinity in 3D space to points at infinity in 2D image plane, although the camera centre is at infinity. Therefore, the image plane does not maintain parallel

features of the world scene. Affine cameras have more practical applications and usage than non-affine cameras.

Affine structure from motion was highlighted by Koenderink and van Doorn [7] showing the importance of acquiring orthographic images for affine reconstruction. Based on this explanation, Tomasi and Kanade [12] introduced a factorization algorithm from multiple 2D images. Also, with the introduction of the affine transformation equation by Mundy and Zisserman [9], it was evident that a 3D affine reconstruction was feasible using *affine images* (2D images with orthographic projection). The affine transformation equation included a translation matrix that was later eliminated by Reid and Murray [11] using a unique property of the affine camera: geometrically, an affine camera maps the centroid of the 3D scene to the centroid of the projected image. This is the main approach I examine in achieving a solution for affine reconstruction.

The main software package I have used in implementing the algorithm for affine reconstruction was MATLAB V6.1 under the Linux Red Hat V8 environment. Initial implementation involved manually digitizing points on 2D images with orthographic projection and later on I used the Kanade, Lucas, Tomasi (KLT) tracker [1] to obtain feature points for an automated feature detection mechanism. The KLT tracker software has been written in C language and runs under the Linux Red Hat V8 environment. Images from synthetic objects gave me the flexibility to experiment and test my approach whilst images from real objects confirmed my analysis.

This thesis is organized into five main sections. Chapter 2 outlines two camera models and their relevance to affine structure from orthographic projection. I study 3D transformations in Chapter 3 and discuss the invariant properties of the affine transformation. Having laid this foundation, I then move on to explain the factorization algorithm for 3D affine reconstruction in Chapter 4. I present a method for metric reconstruction from affine structure in Chapter 5, creating a scaled down 3D model given by 2D images, and texture mapping the 3D metric structure. Chapter 6 explains the method for feature point detection using the KLT tracker [1].

I present my results using synthetic and real images. Chapters 4, 5 and 6 contain results from the relevant chapter content and I focus my discussions primarily on synthetic images. Next, I use real images to confirm my discussions and analysis. Chapter 7 concludes by confirming the principal points and further work related to the subject area discussed throughout this thesis.

CHAPTER 2

Cameras and Projection

In this chapter, after an introduction to the decomposition of general projective camera models, I focus on perspective projection and cameras with a finite centre. The study then moves to orthographic projection and cameras with centre at infinity. Properties of affine cameras are discussed in detail, as they have a direct relation to the algorithm for affine reconstruction.

2.0.1 The general projective camera

The general projective camera is an engine that creates a planar image from 3D information through projection [9]. In notation form, it converts an inhomogeneous $(X, Y, Z)^\top$ point on to an inhomogeneous $(x, y)^\top$ 2D coordinates. General projective camera taxonomy is as follows:

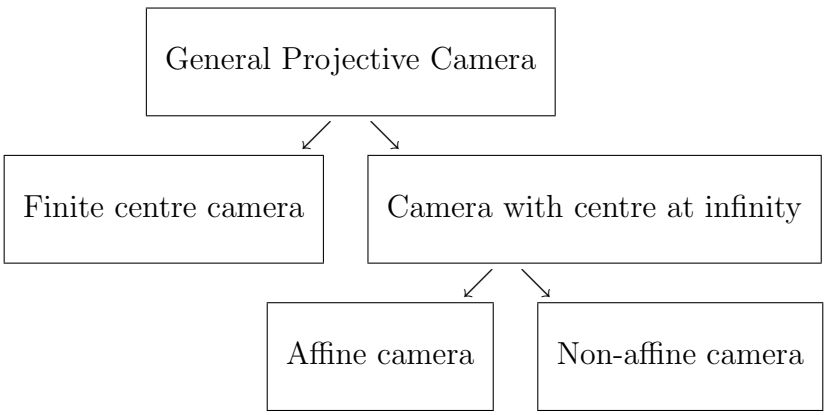

Figure 2.1: General projective camera taxonomy.

2.1 Cameras with a finite centre

2.1.1 Perspective projection

When we gauge the depth of a 3D scene using a 2D image, we compare the size of objects near the camera with objects in the background. If the given image encapsulates enough information, we then decide upon the level of depth depicted by the image. For example, consider the tiled floor in Figure 2.2.

Figure 2.2: Perspective projection of a tiled floor—world lines that are parallel appear to converge [3].

The tile arrangement gives us the ability to visualise converging lines. However, as the tile edges are parallel in real world, the projection of this parallelism is such, that on the 2D image, the tiles appear not to be parallel. This is known as perspective projection. If we extend two parallel lines beyond the image, they converge at a point called a vanishing point. Two vanishing points form a vanishing line. Figure 2.3 shows the vanishing line and two vanishing points of the image of the tiled floor.

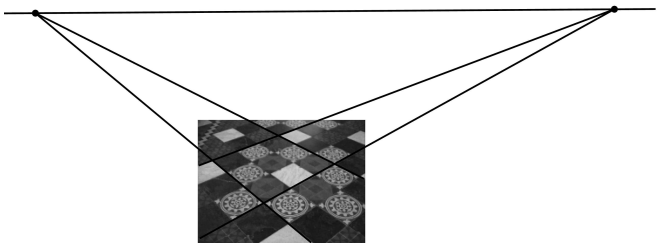

Figure 2.3: Two vanishing points of the tiled floor and the corresponding vanishing line [3].

2.1.2 Pinhole camera

The model that is easy to understand for a camera with a finite centre is that of the pinhole camera. The following figure illustrates the principal properties of a pinhole camera and its mapping of 3D coordinates.

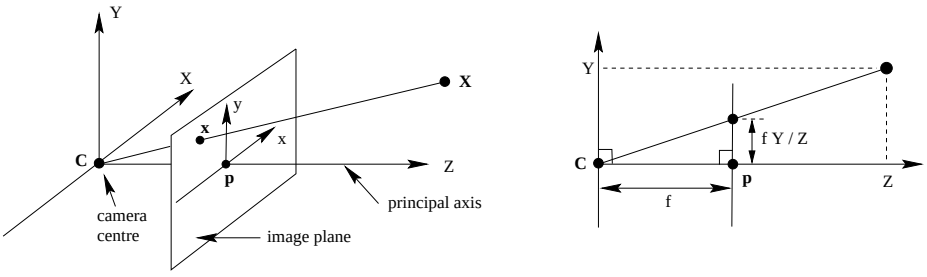

Figure 2.4: Principle properties of pinhole camera [3].

Based on Figure 2.4, basic camera notation can now be shown:

Retinal plane/image plane (R)	–	The plane on to which world coordinates are projected,
Optical/camera centre (C)	–	Position of the camera; a point that does not lie on the image plane,
Principal axis/optical ray (CZ)	–	The perpendicular line from C to the image plane,
Principal point (P)	–	Intersection of principal axis with image plane,
Focal length (f)	–	Finite distance between camera centre and principal point.

Using geometric properties, values for $(x, y)^T$ 2D projective coordinates are shown as,

$$x_c = f\frac{X_C}{Z_C}, \qquad y_c = f\frac{Y_C}{Z_C}. \tag{2.1}$$

Using homogeneous coordinates, Equation 2.1 is now expressed in matrix form,

$$\begin{bmatrix} Z_C x_c \\ Z_C y_c \\ Z_C \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix},$$

or in a more compact form,

$$\mathbf{x} = P_C \mathbf{X}, \tag{2.2}$$

where $\mathbf{x}$ is the 2D projective coordinates, P_C is the camera matrix for perspective projection, and $\mathbf{X}$ is the 3D world coordinates.

2.2 Cameras with centre at infinity

2.2.1 Orthographic projection

We now focus on orthographic projection to build our knowledge on affine cameras. Refer back to Figure 2.4 and consider a situation when the focal length of the projective camera has been extended by a significant amount. Let us observe the geometric changes the image undergoes when the focal length of the camera is gradually increased. Figure 2.5 below depicts a scenario similar to this.

Figure 2.5: From projective to weak perspective—on the left, the camera moves back but zooms in to keep the height of the board constant. Perspective diminishes as the distance from the subject increases. On the right, the same movement is maintained without zooming in.

The image sequence from Figure 2.5 shows how a transition occurs from strong perspective to weak perspective, as the focal length is increased. Therefore, theoretically it can be deduced that when the focal length is at maximum, there exists no perspective. Hence, for a given 2D image with no perspective, parallel lines in the 3D world remain parallel. This is known as orthographic projection.

This concept can be explained using the camera calibration matrix. The camera calibration matrix for a general projective camera is given as [3],

$$C = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & f \end{bmatrix}. \tag{2.3}$$

To scale the calibration matrix, we divide C by the focal length, f giving us,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 1 \end{bmatrix}. \tag{2.4}$$

Let focal length, $f \longrightarrow \infty$; then Equation 2.4 becomes,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.5}$$

Notice the change to the camera calibration matrix between Equation 2.3 and Equation 2.5, particularly the third column. In Equation 2.3 the projection is on to the plane $z = 1$. However, in Equation 2.5, the projection is on the x,y plane.

By definition, the principal point is a point on the image plane and the distance from camera centre to principal point is the focal length [3]. This is based on the assumption that the line from camera centre to the image plane is right angled. If, however, the focal length is gradually increased, the image plane distance increases correspondingly. When the focal length reaches infinity, the principal point also reaches infinity from the camera centre. Hence, the principal point is now on a special plane—the plane at infinity. This forms the basis for the definition of an affine camera.

2.2.2 Affine camera

Hartley and Zisserman [3] offer a definition for an affine camera as a general projective camera that has its focal point on the plane at infinity.

The camera matrix of the affine form is given as [2],

$$\mathbf{P_A} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & t_1 \\ m_{21} & m_{22} & m_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

or in more compact form,

$$\mathbf{P_A} = \begin{bmatrix} \mathbf{M}_{2\times 3} & \mathbf{t} \\ \mathbf{0}_3^\top & 1 \end{bmatrix}, \tag{2.6}$$

where $\mathbf{M}_{2\times 3}$ is the affine camera sub matrix and $\mathbf{t}$ is the translation matrix.

A similarity is drawn between Equation 2.6 and the camera calibration matrix for orthographic projection, Equation 2.5—the last row in both matrices has the form $(0, 0, 0, 1)$. It reaffirms that the affine camera is a special instance of the projective camera.

From the generalized form above, the affine camera matrix can be decomposed into three sub-units to understand its composition [3, 2],

$$\mathbf{P}_A = \begin{bmatrix} \alpha & \gamma & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{3\times 3} & \mathbf{t} \\ \mathbf{0}_3^\top & 1 \end{bmatrix}, \tag{2.7}$$

where α is the scale factor in the x–direction, β is the scale factor in the y–direction, γ is the skew, $\mathbf{R}_{3\times 3}$ is the rotation matrix, and $\mathbf{t}$ the translation matrix.

In simpler form, Equation 2.7 is now written as,

$$P_A = \begin{bmatrix} \text{3–space} \\ \text{affine} \\ \text{transformation} \end{bmatrix} \begin{bmatrix} \text{an orthographic} \\ \text{projection from} \\ \text{3–space} \\ \text{to image plane} \end{bmatrix} \begin{bmatrix} \text{an affine} \\ \text{transformation} \\ \text{of the image} \end{bmatrix}.$$

The main properties of an affine camera are three-fold [3, 9]:

- For a given projective camera matrix, if the principal plane is the plane at infinity, then it is an affine camera matrix.
- Parallel lines in the world scene are projected to parallel lines in the image. All parallel world lines intersect at vanishing points on the plane at infinity. These points at infinity are projected to points at infinity on the image plane under the affine projection.
- The direction of parallel projection $\mathbf{d}$, satisfies the equation $\mathbf{M}_{2\times 3}\mathbf{d} = 0$.

The use of an affine camera in the context of reconstruction has three main benefits. Kahl and Heyden [5] discuss the first benefit as the facility to obtain an affine image of the structure instead of perspective projection. Secondly, the algorithm coupled with geometry and algebra is in a more simpler form that leads to efficient and robust reconstruction. Thirdly, the affine camera is useful when producing invariants of 3D transformation groups that can be recovered from multiple views with uncalibrated cameras. In the next chapter I examine projective and affine 3D transformations, and their respective invariants.

CHAPTER 3

3D Transformations

This chapter illustrates the hierarchy of 3D transformations. The intention is to expand our understanding of affine cameras and find out how they are involved with affine transformation. To achieve this objective we first look at projective transformation and make a comparison with properties of affine transformation.

3.1 Hierarchy of 3D transformations

A 2D transformation involves the extraction of information from the world scene and placing this information on an image plane so as to create the planar view. On the other hand, a 3D transformation is the process of placing features from the world scene into a 3D virtual environment to visualize and model the world scene, under a scaling factor.

The hierarchy of 3D transformations starts from projective and then moves to affine and similarity. Each transformation has its invariant properties and two of the most important features of these transformations are parallelism and angles. Figure 3.1 illustrates the transformations of a cube.

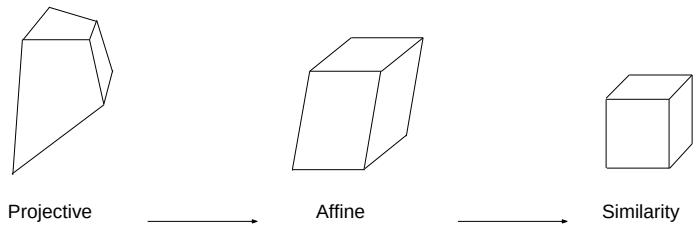

Figure 3.1: Transformation hierarchy of a cube in 3D space.

We focus on projective and affine transformations to understand the 3D affine structure.

3.2 3D projective transformation

All geometric effects that appear in a world scene can be projected so that features closer to the camera have larger dimensions than those in the background. This general mapping of perspective viewing is called projective transformation [9].

Mundy and Zisserman [9] outline two important points about projective transformation:

1. A projective transformation can represent any perspective projection.
2. A collective chain of perspective projections is always a projective transformation.

We can represent the projective transformation from one projective plane to another [9] using homogeneous coordinates in the form,

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_X \\ a_{21} & a_{22} & a_{23} & t_Y \\ a_{31} & a_{32} & a_{33} & t_Z \\ v_1 & v_2 & v_3 & v \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$

where $a_{11}...a_{33}$ is an invertible 3×3 matrix, t_X, t_Y, t_Z is a 3D translation in each direction, $\mathbf{v}=(v_1, v_2, v_3)^\top$ is a general vector, and v is a scale factor.

3.3 3D affine transformation

The advantage of having a projective transformation is to be able to view the scene in perspective. In simple terms, an affine transformation is a non-singular linear transformation which undergoes a translation. We compared the transformations at the beginning of this chapter from Figure 3.1 and noted the differences from a geometric point of view.

In matrix representation, a 3D affine transformation is denoted by [3],

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_X \\ a_{21} & a_{22} & a_{23} & t_Y \\ a_{31} & a_{32} & a_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}.$$

3.3.1 Invariants of affine transformation

Once an object is affine transformed, there are three geometric properties that remain invariant of the transformation [3]. These are as follows :

1. Parallel lines—A pair of parallel lines in the 3D world have a vanishing point at infinity. Under affinity, this vanishing point would still be mapped to another point at infinity. This means that after the affine transformation, the lines must remain parallel for them to have a vanishing point at infinity. Therefore, parallel lines remain parallel after an affine transformation.

2. Ratio of lengths of directional lines—Lines that are placed in the same direction contain a ratio that is invariant. It implies that the ratios of two lengths along parallel lines remain invariant and the ratio of two lengths that are not parallel is not invariant.

3. Ratio of areas—As parallel lines remain parallel after an affine transformation, the ratios of surface areas remain invariant. This builds on from the second invariant because the directional ratios of lengths determine the area and two areas would therefore have the same ratio.

   The invariants of affine transformation highlight the importance of affine geometry. Now we understand the properties of affine transformation and what information can be gathered by observing an affine object. The next chapter examines 3D affine reconstruction and introduces an algorithm for creating objects in the affine space.

CHAPTER 4

Affine Reconstruction

The focus of this chapter is on creating a 3D affine structure from images obtained from affine cameras. Firstly, I discuss the factorization algorithm [3], the foundation for 3D affine reconstruction. An essential component of factorization algorithm is the mathematical procedure of Singular Value Decomposition (SVD). After studying the SVD we then proceed to experiments that have been conducted on affine reconstruction, using images from synthetic objects and real objects.

4.1 The factorization algorithm

Tomasi and Kanade [12] introduced the factorization algorithm for an affine reconstruction using a *measurement matrix*. The algorithm assumes that, the image points to be measured, are present in all image views. As explained in Chapter 2, the camera matrix of an affine camera has the form,

$$P_A = \begin{bmatrix} \mathbf{M}_{2\times 3} & \mathbf{t} \\ 0_3^\top & 1 \end{bmatrix},$$

where $\mathbf{M}_{2\times 3}$ is the transformation matrix and $\mathbf{t}$ is the translation matrix.

A 2D inhomogeneous affine image coordinate can therefore be represented as,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{M}_{2\times 3} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}, \tag{4.1}$$

or in more compact form,

$$\mathbf{x} = \mathbf{M}\mathbf{X} + \mathbf{t}. \tag{4.2}$$

The significance of Equation 4.1 is two-fold: firstly, it only uses a sub-matrix from the affine camera matrix, decomposing the affine camera matrix into two areas. Secondly, it acquires the 2D affine coordinates $(x, y)^\top$ as a two stage process of transforming the 3D coordinates $(X, Y, Z)^\top$ and then translating them.

If all the estimated 3D world points of the reconstruction are converted to (x, y) coordinates using Equation 4.1, then we have the affine image of the world scene as viewed by an affine camera. These image coordinates are commonly known as the *projected image points*. Another set of image points called *observed image points* exists by observing the physical image of the object [3].

There exists a geometric error between projected image points and observed image points. The objective in reconstructing a 3D model of the scene using several views of the object is to minimize this geometric error so that we can use the refined set of points to build an affine structure.

Let us examine the minimization equation. Using Equation 4.2, we define the observed points as seen from the i^{th} camera view to be,

$$\hat{\mathbf{x}}_j^i = \mathbf{M}^i\mathbf{X}_j + \mathbf{t}^i, \tag{4.3}$$

where $\hat{\mathbf{x}}_j^i$ is the inhomogeneous observed image point j in image i, $\mathbf{M}^i$ is the motion matrix of camera i, $\mathbf{X}_j$ is the inhomogeneous world point j, and $\mathbf{t}^i$ is the translation vector of camera i.

If we assume the projected points for the affine object to be $\mathbf{x}_j^i$, then we can minimize the geometric error, also called the reprojection error, as [3]:

$$\mathbf{x}_j^i - \hat{\mathbf{x}}_j^i = \mathbf{x}_j^i - (\mathbf{M}^i\mathbf{X}_j + \mathbf{t}^i). \tag{4.4}$$

The aim is to minimize the above distance $(\mathbf{x}_j^i - \hat{\mathbf{x}}_j^i)$, for all image points over every image frame.

Therefore, from Equation 4.4, we can minimize the geometric error as [3]:

$$\min_{\mathbf{M}^i,\mathbf{t}^i,\mathbf{X}_j} \sum_{ij} (\mathbf{x}_j^i - \hat{\mathbf{x}}_j^i)^2 = \min_{\mathbf{M}^i,\mathbf{t}^i,\mathbf{X}_j} \sum_{ij} (\mathbf{x}_j^i - (\mathbf{M}^i\mathbf{X}_j + \mathbf{t}^i))^2. \tag{4.5}$$

Mundy and Zisserman [9] state that the translation vector $\mathbf{t}^i$ can be removed from Equation 4.5 above by making the centroid of image points as the origin of the coordinate system. This statement is supported by the mapping of 3D points by an affine camera: the centroid of a set of 3D coordinates is mapped to the centroid of the projected points [3], as illustrated below.

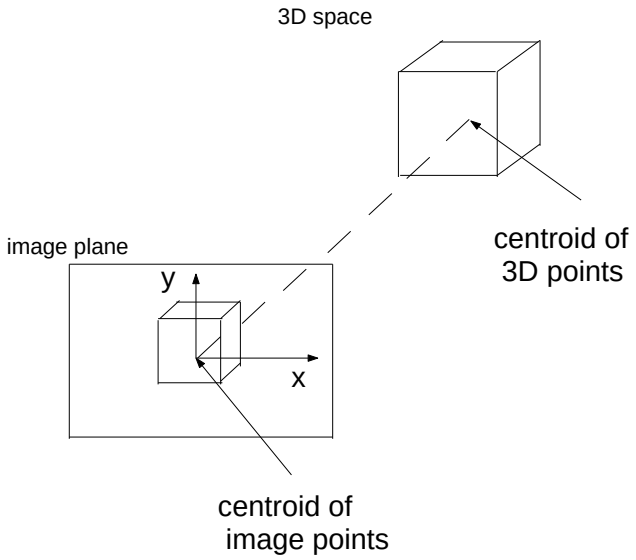

Figure 4.1: Centroid mapping from 3D space to image plane..

Therefore, image points are now recalculated with respect to the centroid. This implies that if the centroid of image points is made to be the origin then $\mathbf{t}^i = 0$. For this condition to be valid, all points considered (j) must be visible from every image frame (i).

Based on this derivation, Equation 4.5 now becomes,

$$\min_{\mathbf{M}^i,\mathbf{t}^i,\mathbf{X}_j} \sum_{ij} (\mathbf{x}_j^i - \hat{\mathbf{x}}_j^i)^2 = \min_{\mathbf{M}^i,\mathbf{t}^i,\mathbf{X}_j} \sum_{ij} (\mathbf{x}_j^i - \mathbf{M}^i\mathbf{X}_j)^2. \tag{4.6}$$

As the aim is to minimize the geometric error, ideally we want Equation 4.6 to be zero.

Therefore,

$$\min_{\mathbf{M}^i,\mathbf{t}^i,\mathbf{X}_j} \sum_{ij} (\mathbf{x}_j^i - \hat{\mathbf{x}}_j^i)^2 = 0. \tag{4.7}$$

Combining Equations 4.6 and 4.7 we can see that,

$$\min_{\mathbf{M}^i,\mathbf{t}^i,\mathbf{X}_j} \sum_{ij} (\mathbf{x}_j^i - M^i\mathbf{X}_j)^2 = 0,$$

$$\Rightarrow \qquad \mathbf{x}_j^i - \mathbf{M}^i\mathbf{X}_j = 0,$$

$$\Rightarrow \qquad \mathbf{x}_j^i = \mathbf{M}^i\mathbf{X}_j. \tag{4.8}$$

Hartley and Zisserman [3] introduced a *measurement matrix* $\mathbf{W}$, consisting the centred coordinates of the observed image points, such that,

$$\mathbf{W} = \begin{bmatrix} \mathbf{x}_1^1 & \mathbf{x}_2^1 & \dots & \mathbf{x}_n^1 \\ \mathbf{x}_1^2 & \mathbf{x}_2^2 & \dots & \mathbf{x}_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_1^m & \mathbf{x}_2^m & \dots & \mathbf{x}_n^m \end{bmatrix}, \tag{4.9}$$

where $\mathbf{x}_j^i$ is the (x, y) image coordinates of point j in image i with respect to the centroid, n is the number of points identified in every image, and m is the number of images available.

From Equation 4.8, as each $\mathbf{x}_j^i = \mathbf{M}^i\mathbf{X}_j$ then,

$$\mathbf{W} = \begin{bmatrix} \mathbf{M}^1 \\ \mathbf{M}^2 \\ \vdots \\ \mathbf{M}^m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots & \mathbf{X}_n \end{bmatrix}. \tag{4.10}$$

At this stage Hartley and Zisserman [3] stated that, in the presence of noise, Equation 4.10 will not be satisfied. As an alternative, they introduced matrix $\hat{\mathbf{W}}$: this matrix is a variant of $\mathbf{W}$ such that $\hat{\mathbf{W}}$ is as close as possible to $\mathbf{W}$ in the Frobenius norm.

Thus,

$$\hat{\mathbf{W}} = \begin{bmatrix} \hat{\mathbf{x}}_1^1 & \hat{\mathbf{x}}_2^1 & \dots & \hat{\mathbf{x}}_n^1 \\ \hat{\mathbf{x}}_1^2 & \hat{\mathbf{x}}_2^2 & \dots & \hat{\mathbf{x}}_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{x}}_1^m & \hat{\mathbf{x}}_2^m & \dots & \hat{\mathbf{x}}_n^m \end{bmatrix} = \begin{bmatrix} \mathbf{M}^1 \\ \mathbf{M}^2 \\ \vdots \\ \mathbf{M}^m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots & \mathbf{X}_n \end{bmatrix}.$$

Therefore we can re-define $\hat{\mathbf{W}}$ to be,

$$\hat{\mathbf{W}} = \hat{\mathbf{M}}\hat{\mathbf{X}}, \tag{4.11}$$

where $\hat{\mathbf{M}}$ is the $2m \times 3$ motion matrix and $\hat{\mathbf{X}}$ is the $3 \times n$ structure matrix.

The method of finding the required rank 3 matrix $\hat{\mathbf{W}}$ is by performing a mathematical operation known as the Singular Value Decomposition (SVD) on $\mathbf{W}$, truncated to rank 3. An analysis of SVD is shown in the next section. If SVD($\mathbf{W}$)=$\mathbf{UDV}^\top$ then,

$$\hat{\mathbf{W}} = \mathbf{U}_{2m\times 3}\mathbf{D}_{3\times 3}{\mathbf{V}_{n\times 3}}^\top. \tag{4.12}$$

From equations 4.11 and 4.12, we deduce that,

$$\hat{\mathbf{M}}\hat{\mathbf{X}} = \mathbf{U}_{2m\times 3}\mathbf{D}_{3\times 3}{\mathbf{V}_{n\times 3}}^\top.$$

By comparison,

$$\hat{\mathbf{M}} = \mathbf{U}_{2m\times 3} \qquad \text{and}$$
$$\hat{\mathbf{X}} = \mathbf{D}_{3\times 3}{\mathbf{V}_{n\times 3}}^\top, \tag{4.13}$$

or

$$\hat{\mathbf{M}} = \mathbf{U}_{2m\times 3}\mathbf{D}_{3\times 3} \qquad \text{and}$$
$$\hat{\mathbf{X}} = {\mathbf{V}_{n\times 3}}^\top. \tag{4.14}$$

Therefore, given a series of affine images, the 3D coordinates for the affine structure are determined to be either $\mathbf{D}_{3\times 3}{\mathbf{V}_{n\times 3}}^\top$ or ${\mathbf{V}_{n\times 3}}^\top$. Mundy and Zisserman [9] mention that by using two affine images and the above procedure, we can obtain the 3D affine coordinates.

4.2 Singular value decomposition

This is a mathematical functionality that involves the decomposition of a matrix into three individual components. I describe the standard SVD decomposition of a matrix A and discuss the properties of the decomposition. I then focus on the algorithm that enables the SVD of a given matrix.

4.2.1 Properties of the SVD

Given a $m \times n$ matrix $\mathbf{A}$, its SVD is written as [10],

$$\mathbf{A} = \mathbf{U}.\mathbf{D}.\mathbf{V}^{\top}, \tag{4.15}$$

where $\mathbf{U}$ is a $m \times m$ orthogonal matrix, $\mathbf{D}$ is a $m \times n$ diagonal matrix, and $\mathbf{V}$ is a $n \times n$ orthogonal matrix.

The following properties are exhibited by SVD [10, 6]:

1. As $\mathbf{U}$ and $\mathbf{V}$ are orthogonal,

$$\left(\mathbf{U}^{\top} \right) . \left(\mathbf{U} \right) = \left(\mathbf{I}_{m \times m} \right),$$

and

$$\left(\mathbf{V}^{\top} \right) . \left(\mathbf{V} \right) = \left(\mathbf{I}_{n \times n} \right).$$

2. The matrix $\mathbf{D}$ is a diagonal matrix with singular values. It has non-negative entries in descending order.

If $n > m$,

$$\mathbf{D} = \begin{matrix}\uparrow \\ m \\ \downarrow\end{matrix} \left(\underbrace{\begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_m \end{bmatrix}}_{m} \quad \underbrace{\begin{bmatrix} 0 & \dots & 0 \end{bmatrix}}_{(n-m)} \right),$$

where $\quad d_1 \geq \ldots \geq d_m \geq 0.$

If $m > n$,

$$\mathbf{D} = \underbrace{\begin{matrix}\uparrow \\ n \\ \downarrow \\ \\ \uparrow \\ (m\text{-}n) \\ \downarrow\end{matrix} \left(\begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix} \quad \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \right)}_{n},$$

where $\quad d_1 \geq \ldots \geq d_n \geq 0.$

3. The square root of the sum of all elements squared in matrix **A** gives the Frobenius norm of that matrix. The square root of the sum of all diagonal elements in matrix **D** also equates to the same value as the Frobenius norm.

   Hence, the Frobenius norm is mathematically given as:

   From matrix $\mathbf{A} \Rightarrow \sqrt{a_{11}^2 + a_{12}^2 + \ldots + a_{1n}^2 + a_{21}^2 + \ldots + a_{mn}^2}$

   From matrix $\mathbf{D} \Rightarrow \sqrt{d_1^2 + d_2^2 + \ldots + d_n^2}$

4. The SVD is defined in general, for matrices with more rows than columns ($m > n$). However, it can also be used for a matrix with more columns than rows ($m < n$). In such an instance, the SVD is computed only after additional rows of zeros have been inserted to have the same number of rows as columns (m=n).

4.2.2 Algorithm for SVD

Based on the discussion by [6, 8], we now study the algorithm for calculating the SVD of a given matrix. Prior to moving into the procedure for SVD algorithm, let us seek the *eigenvalues* and *eigenvectors* of a matrix.

If a given square matrix A exhibits the following property,

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \lambda \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix},$$

or in more compact form,

$$\mathbf{Aq} = \lambda \mathbf{q}.$$

Then, λ is the eigenvalue and $\mathbf{q}$ is the eigenvector for the given matrix $\mathbf{A}$. Therefore, the eigenvector multiplied by the matrix gives a vector ($\lambda\mathbf{q}$) proportional to the matrix itself. The constant of proportionality is defined as the eigenvalue [4].

To calculate the SVD of a given matrix $\mathbf{A}$,

1. We find the eigenvectors of $\mathbf{AA}^\top$ and $\mathbf{A}^\top\mathbf{A}$,
2. The eigenvectors of $\mathbf{A}^\top\mathbf{A}$ form the columns of matrix $\mathbf{V}$,
3. The eigenvectors of $\mathbf{AA}^\top$ form the columns of matrix $\mathbf{U}$,
4. The singular values of $\mathbf{D}$ are square roots of eigenvalues from $\mathbf{AA}^\top$ or $\mathbf{A}^\top\mathbf{A}$. We place these singular values in descending order across the diagonal of matrix $\mathbf{D}$.

4.2.3 Relationship of SVD with measurement matrix

Analysis of SVD and measurement matrix with the two solutions for affine coordinates (Equations 4.13 and 4.14) is presented below, with a theoretical example. Consider two affine images with four points visible in both images, as shown in Figure 4.2.

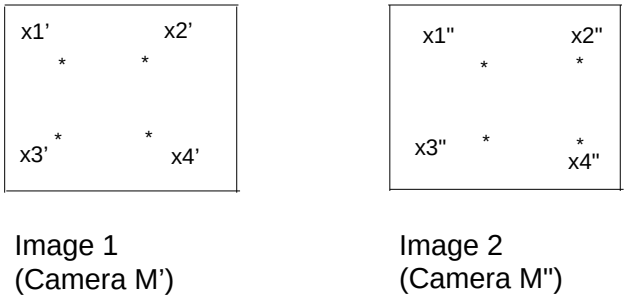

Figure 4.2: Four points considered for theoretical analysis for affine reconstruction using SVD.

We construct the measurement matrix to be,

$$\mathbf{W} = \begin{bmatrix} \mathbf{x}1' & \mathbf{x}2' & \mathbf{x}3' & \mathbf{x}4' \\ \mathbf{x}1'' & \mathbf{x}2'' & \mathbf{x}3'' & \mathbf{x}4'' \end{bmatrix}.$$

Taking the SVD of the measurement matrix gives us,

$$\text{SVD}(\mathbf{W}) = \mathbf{U}\mathbf{D}\mathbf{V}^\top.$$

Irrespective of the number of rows in the measurement matrix, the diagonal matrix $\mathbf{D}$, only contains positive values for the first three entries in the diagonal and the rest of the values are either zero or negligible. Hence we only consider the first three columns from $\mathbf{U}$ and first three rows from $\mathbf{V}^\top$.

Therefore, $\mathbf{W}$ is decomposed as,

$$\mathbf{W} = \underbrace{\begin{bmatrix} \begin{bmatrix} & M' & \end{bmatrix} \\ \begin{bmatrix} & M'' & \end{bmatrix} \end{bmatrix}}_{4\times3 \;\; motion\; matrix} \underbrace{\begin{bmatrix} X1 & X2 & X3 & X4 \\ Y1 & Y2 & Y3 & Y4 \\ Z1 & Z2 & Z3 & Z4 \end{bmatrix}}_{3\times4 \;\; structure\; matrix}.$$

The motion matrix and structure matrix are computed from $\mathbf{U}$, $\mathbf{D}$, $\mathbf{V}^\top$ as shown in Equations 4.13 and 4.14.

We can reproduce the 2D coordinates of the affine image, as seen by either camera $\mathbf{M}'$ or $\mathbf{M}''$, if the corresponding motion matrix is multiplied by the structure matrix. Thus, there is a direct relationship between the motion matrix and the structure matrix with each affine image.

4.3 Results—synthetic images

I present the results of 3D affine reconstruction broadly from two angles. I use the factorization algorithm [3] and the SVD, to arrive at solutions for 3D affine coordinates as we have seen in Equations 4.13 and 4.14. Firstly, a synthetic object is created in 3D space and its 2D images, with both orthographic and perspective projection, are used for reconstruction to verify the algorithm. For the synthetic object, I assume that all points are visible from every image frame so that the affine structure can be analyzed without difficulty. Secondly, I perform the same exercise for 2D images with weak perspective, from a real object. MATLAB V6.1 has been used for matrix computation and 3D modeling, in a Linux environment. Results pertaining to each affine reconstruction are discussed at the end of each experiment.

4.3.1 Orthographic projection

Figure 4.3 shows the original position of the synthetic object that has been created.

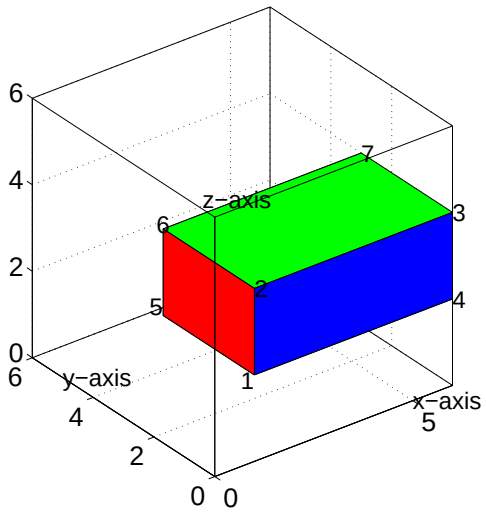

Figure 4.3: Original position of the synthetic object. The object is an oblong having a length of five units, width of three units and a height of two units. Each corner of the object is numbered for identifying edges and surfaces during analysis.

The object is rotated about x,y and z axis at one degree intervals per rotation to visualise orthographic projection. The object position and its corresponding orthographic projection captured at four different stages during the rotation, are given in Figure 4.4.

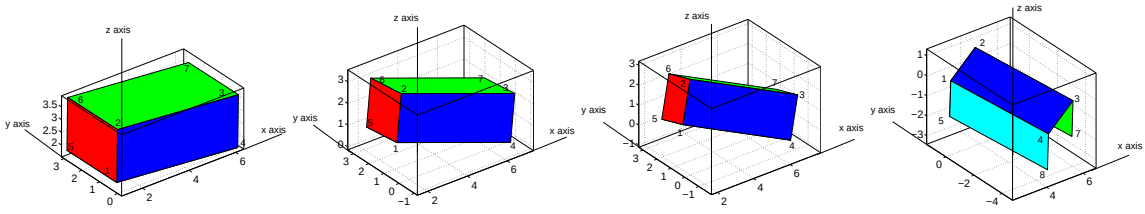

Figure 4.4: Rotation of the 3D object, at four different stages.

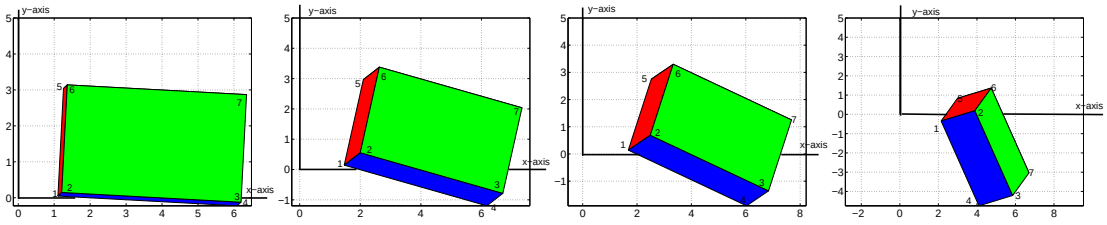

Figure 4.5: Corresponding orthographic views of the rotated object. The x,y coordinates of these images are used for affine reconstruction.

I subjected the object to 180 rotations at one degree intervals and the results are graphically demonstrated below. As we have two solutions to determine the affine structure in Equations 4.13 and 4.14, I show the results from both equations.

I labeled the solutions as follows:

$$\text{First Solution} - \hat{\mathbf{X}}_{\mathbf{1}} = \mathbf{D}_{3\times3}\mathbf{V}_{n\times3}{}^{\top}$$

$$\text{Second Solution} - \hat{\mathbf{X}}_{\mathbf{2}} = \mathbf{V}_{n\times3}{}^{\top}$$

1. Rotation about the x axis

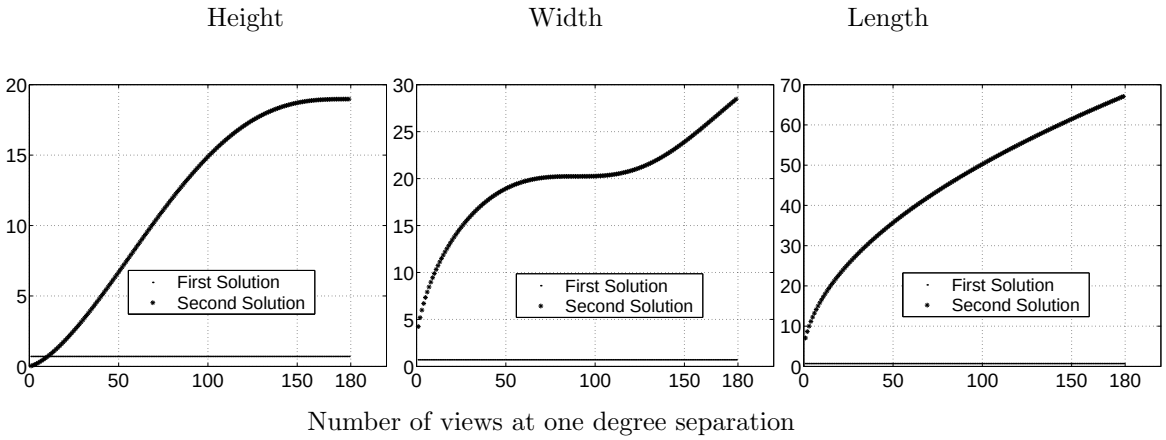

Figure 4.6: Rotation of the 3D object about the x axis. The height refers to $distance_{12}$, width refers to $distance_{15}$ and length refers to $distance_{14}$ from Figure 4.3. The significance of the three graphs is the behaviour of physical dimensions of the affine structure created from first solution–length, width and height remain constant irrespective of the number of views. Unlike the first solution, the results from the second solution show that dimensions are dependent on number of rotations.

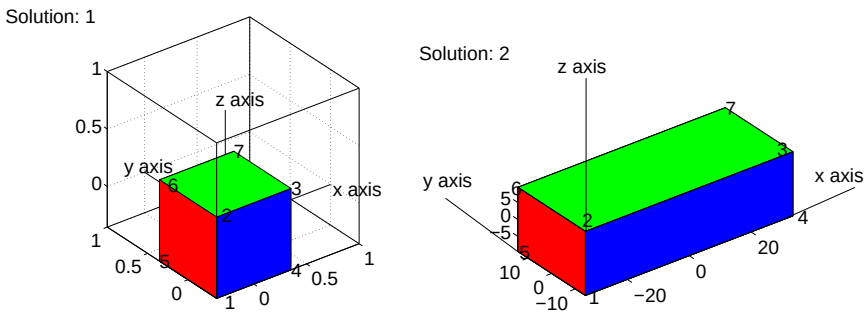

Figure 4.7: Affine 3D structures of the object after 180 rotations about the x axis at one degree separation. The 3D structure has been repositioned by treating the centroid as the coordinate origin.

2. Rotation about the y axis

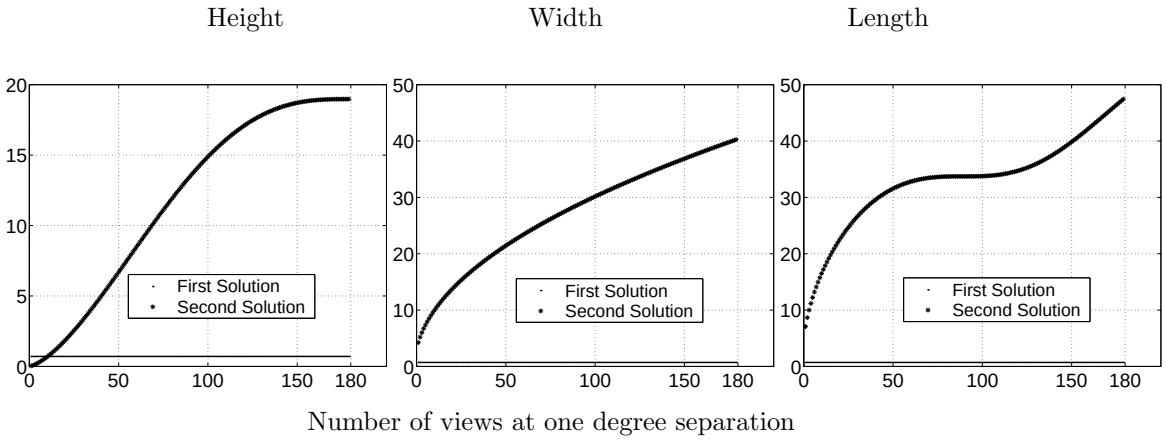

Figure 4.8: Rotation of the 3D object about the y axis. The graphs demonstrate a similar behaviour in physical dimensions, as before. In the case of first solution, any changes in the rotation angle does not appear to have an impact on the final structure. However, we can see that using the second solution, there is a gradual increase in the dimensions considered, when the object is rotated 180 degrees.

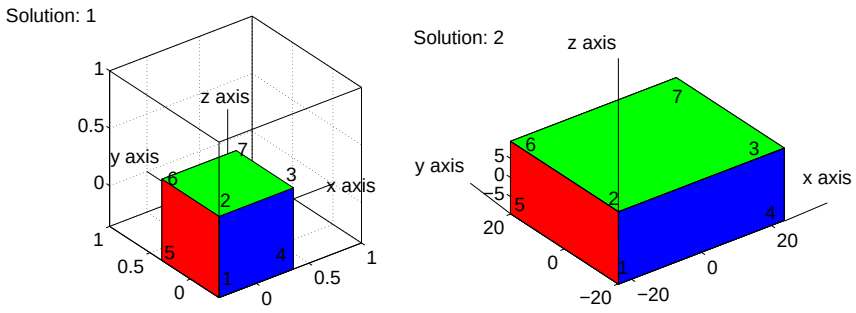

Figure 4.9: Affine 3D structures of the object after 180 rotations about the y axis at one degree separation.

3. Rotation about the z axis

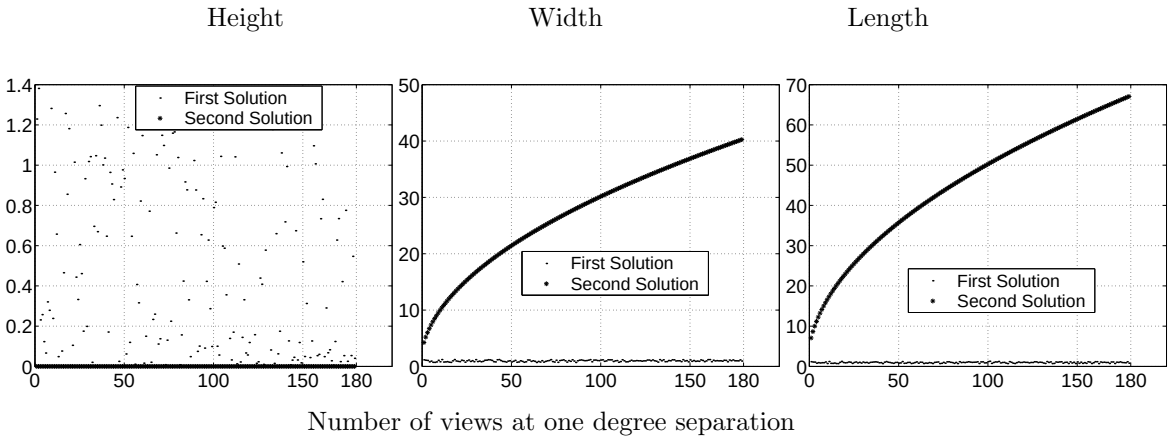

Figure 4.10: Rotation of the 3D object about the z axis. The height graph confirms that the camera cannot obtain any information on the third dimension (height) when motion of the object plane is parallel to the image plane. This has an impact on the overall solutions for the 3D coordinates. Figure 4.11 verifies this argument.

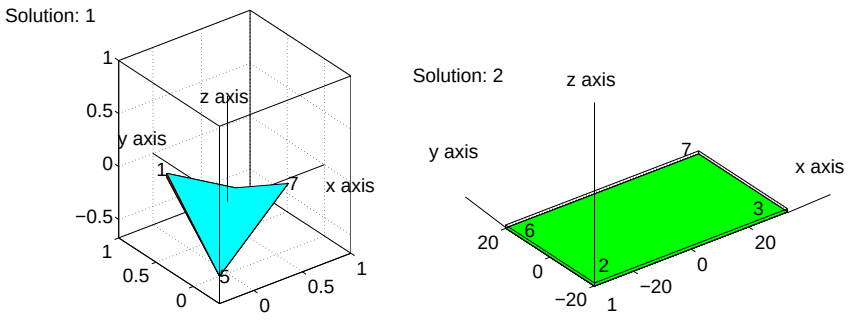

Figure 4.11: Affine 3D structures of the object after 180 rotations about the z axis at one degree separation.

4. Rotation about all three axes

   We have seen the affine structures when the object is rotated about a given axis. I now show the affine structure when the object is rotated about all three axes using 180 iterations to rotate the object. Figure 4.12 shows three distinct positions of the original object and Figures 4.13, 4.14 give the corresponding views of the affine structures.

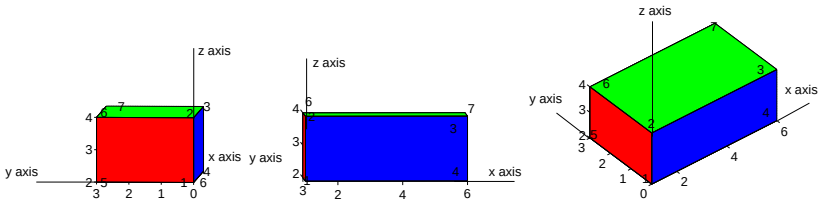

Figure 4.12: Three views of the original object.

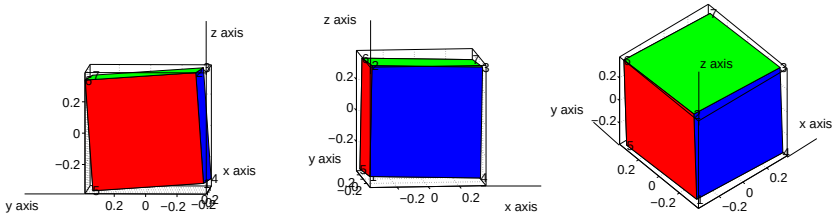

Figure 4.13: Corresponding views of the affine structure obtained from the first solution.

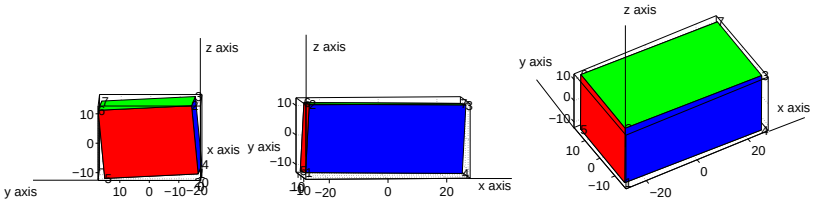

Figure 4.14: Corresponding views of the affine structure obtained from the second solution.

As explained in Chapter 3, the reconstructed 3D objects preserve parallelism. The original object in 3D space (Figure 4.3) has been reconstructed such that parallel lines in the world scene remain parallel in the affine environment.

In this reconstruction, we observe two main distinctions. Firstly, lines that are perpendicular in the original object are not perpendicular in the affine object. For example, consider the frontal views–the affine structure has deformed and in the process, an angular difference has arisen between the horizontal plane and perpendicular plane. Secondly, different scaling factors operate along each axis. This is the reason for affine structure in Figure 4.13 to 'appear' as a cube. Also, we can see from the frontal views that the scaling factors along the z axis are different for all three Figures 4.12, 4.13 and 4.14.

As the minimum number of affine images required for 3D reconstruction is two [9], if corresponding points in each image can be matched using a matching algorithm, then the 3D affine structure can be reconstructed [3]. I demonstrate this under Section 4.4, for a pair of real images. Next, we examine the behaviour of affine reconstruction under orthographic projection with noise.

4.3.2 Orthographic projection with noise

Correspondence matching across a series of given images, is a heavily researched area in computer vision [3]. We can either manually identify and match corresponding points or use a tracking solution to detect features [1]. But in either case, precision accuracy for feature matching is difficult to achieve. Noise induced orthographic projection simulate an environment when observed image points are tracked for affine reconstruction. I present the 3D affine reconstructions obtained from 2D image coordinates for the same synthetic object (Figure 4.3), under noise induced projection.

The following figures exhibit top views of the affine reconstruction with a normal distribution of noise.

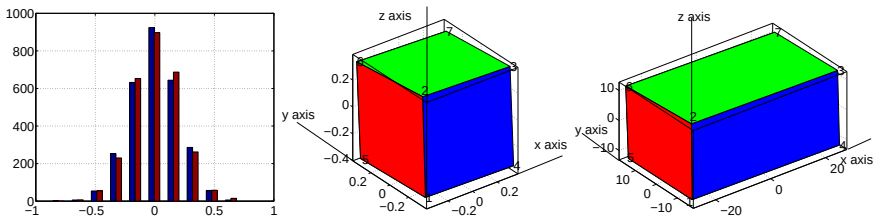

Figure 4.15: Histogram of noise with a standard deviation of 0.2 for x,y coordinates; Corresponding views of the affine structure from the first and the second solution.

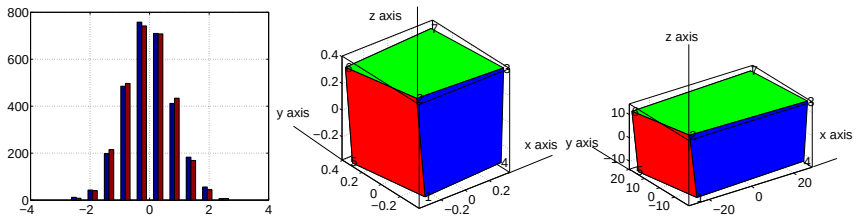

Figure 4.16: Histogram of noise with a standard deviation of 0.8 for x,y coordinates; Corresponding views of the affine structure from the first and the second solution.

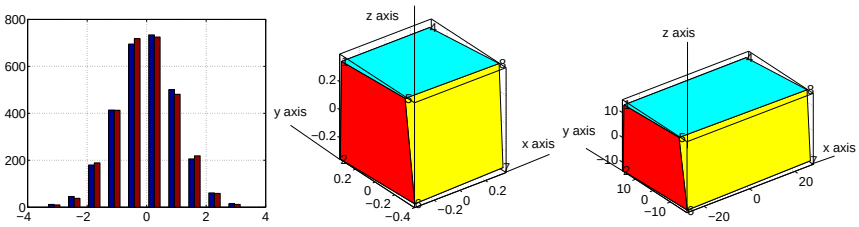

Figure 4.17: Histogram of noise with a standard deviation of 1.0 for x,y coordinates; Corresponding views of the affine structure from the first and the second solution. We can see from Figures 4.15 and 4.16 that due to distortion, the affine structure does not hold the parallelism property. In fact, the affine structure given by Figure 4.17 violates the final position as the algorithm places the reconstructed affine object after rotating about the x axis.

Figure 4.18 illustrates the behaviour of height, width and length of the 3D object in the presence of noise. The graphs confirm that minimized noise levels achieve accurate affine reconstruction and preserve affine properties.

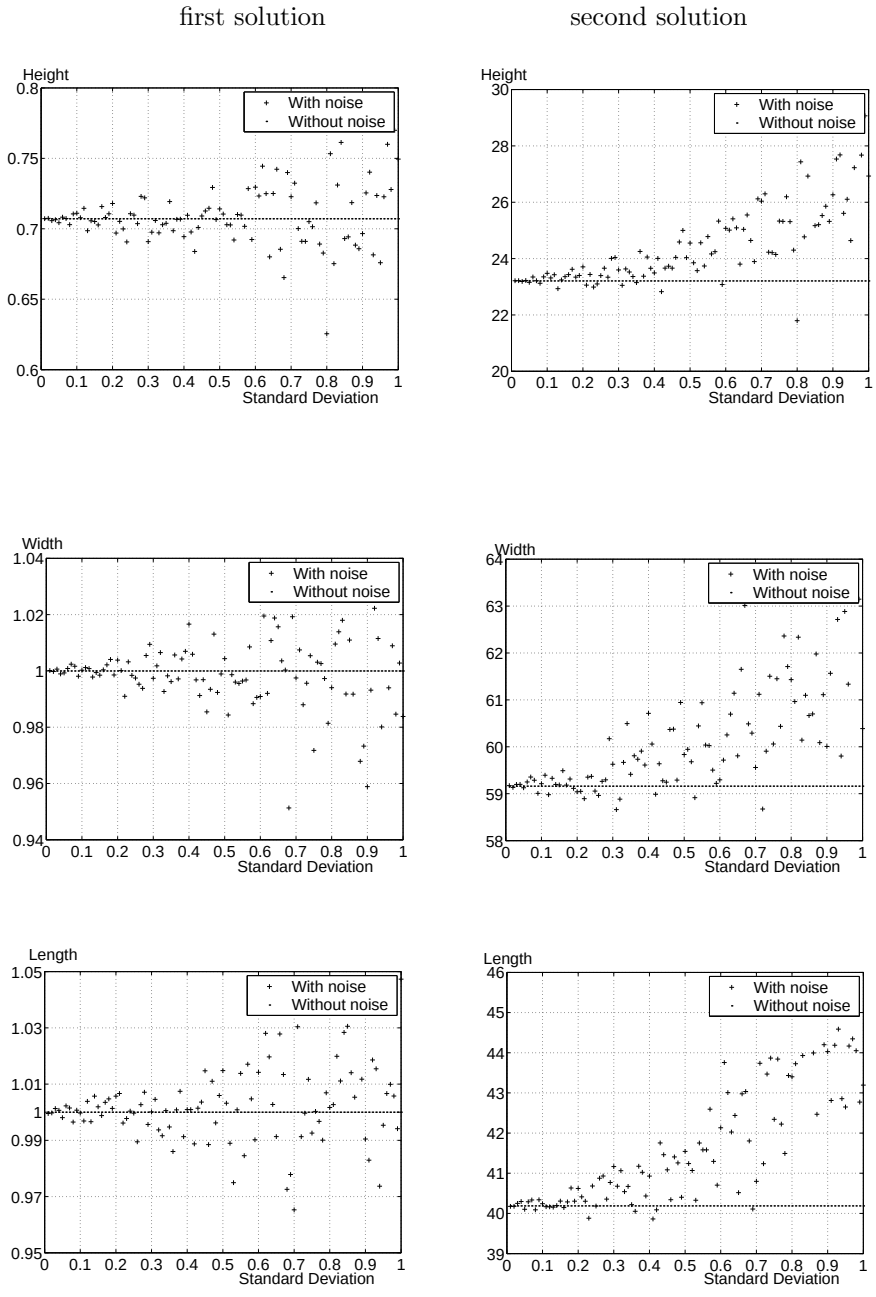

Figure 4.18: Behaviour of affine structure in the presence of noise.

4.3.3 Perspective projection

We saw in Section 4.3.2 that affine reconstruction is susceptible to noise. Feature point matching must be precise for the reconstruction to be be affine. This section presents results for images of the same synthetic object (Figure 4.3) with perspective projection.

I show how the reconstructed structure reaches a more accurate affine state as the images used for the algorithm weaken their perspective, reaching orthographic projection. I employ both solutions (Equations 4.13 and 4.14) to reconstruct the affine structure.

The graphical demonstration uses the frontal surface adjoining vertices 1-2-6-5 of the 3D object, to verify parallelism. Ideally, reconstructed affine structure bears a frontal surface in the shape of a parallelogram. As the opposite sides of a parallelogram are equal, the ratio of the lengths of the opposite sides must be unity. This forms the basis for my argument: if the ratio of the opposite sides reaches unity then, the considered surface forms a parallelogram.

Perspective projection of the 3D object:

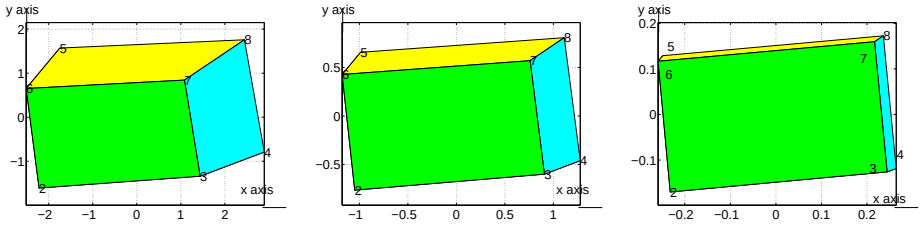

Figure 4.19: As the camera centre is moved away from the object, perspective diminishes and 'appears' to have orthographic projection.

Results from first solution ($\hat{\mathbf{X}}_\mathbf{1} = \mathbf{D}_{3\times3}\mathbf{V}_{n\times3}{}^\top$):

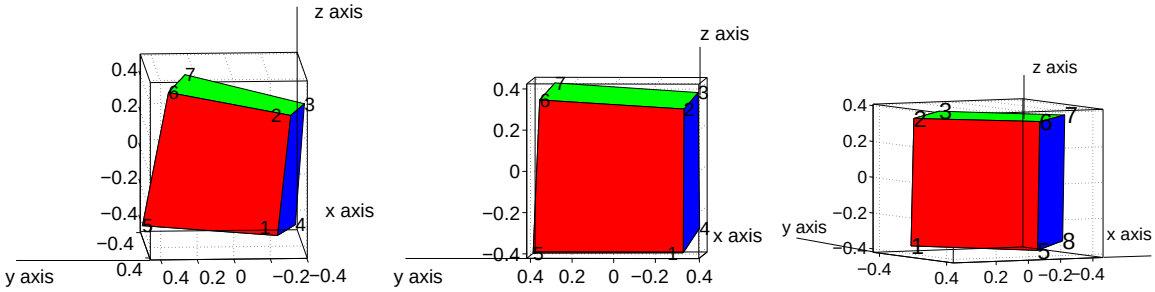

Figure 4.20: Corresponding 3D reconstructions. As perspective weakens, the 3D structure inclines to be affine.

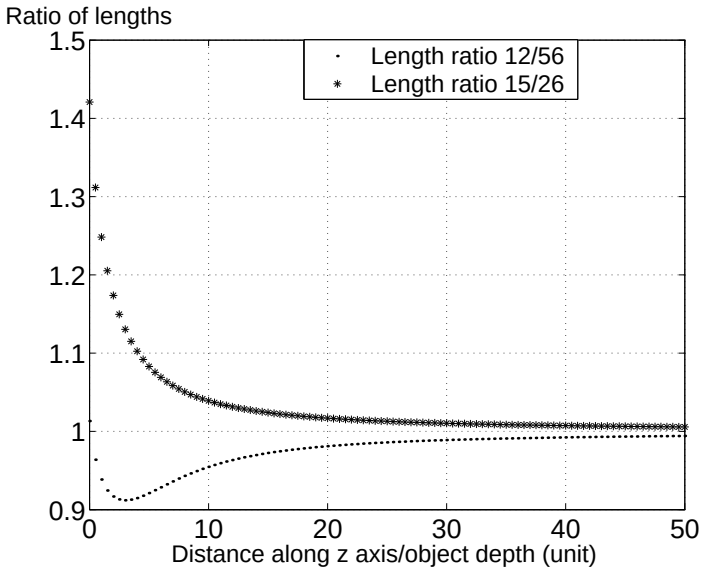

Figure 4.21: Graph verifying parallelism from solution 1. The length refers to the numbered edges on the original object. As the camera is moved along the z axis, away from the object, the length ratio reaches unity, indicating a more parallel structure.

Results from second solution ($\hat{\mathbf{X}}_\mathbf{2} = \mathbf{V}_{n\times 3}{}^\top$):

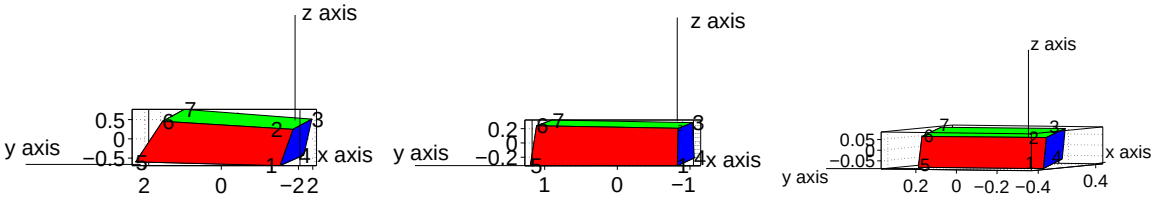

Figure 4.22: Corresponding 3D reconstructions. As perspective weakens, the 3D structure inclines to be affine. The frontal surface of the structure is shown for comparison.

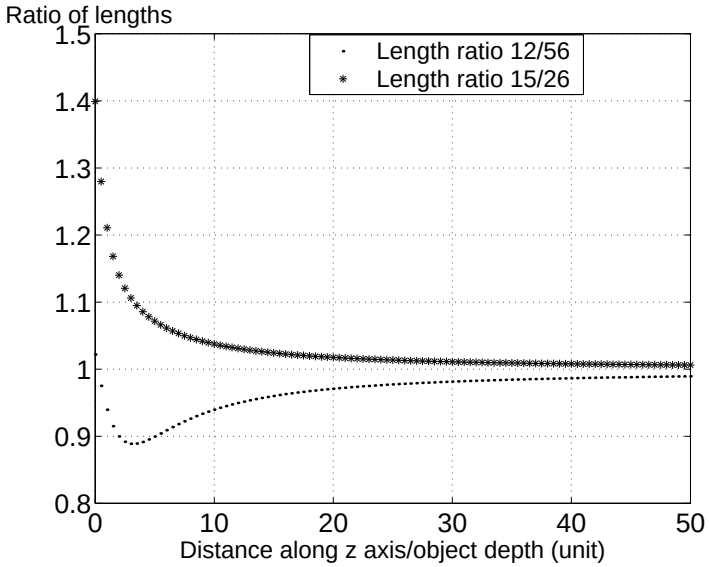

Figure 4.23: Graph verifying parallelism from solution 2. The length refers to the numbered edges on the original object. As the camera is moved along the z axis, away from the object, the length ratio reaches unity, indicating a more parallel structure.

The factorization algorithm presented by Hartley and Zisserman [3] for affine reconstruction uses images with orthographic projection. As I have shown, the reconstructed structure becomes affine when perspective is diminished. The graphs from Figure 4.21 and Figure 4.23 support this argument and verify that the images used for affine reconstruction must either have weak perspective or orthographic projection.

4.4 Results—real images

Real images with weak perspective are subjected to affine reconstruction, in this section. I used two images to show that an affine structure can be acquired using an image pair. However, the image pair must satisfy one significant condition: all points considered for affine transformation must be visible from both images [3]. I used a manual digitizing method to identify point correspondences from both images. This avoided the need to employ a mechanism to track feature points.

4.4.1 Image points identification

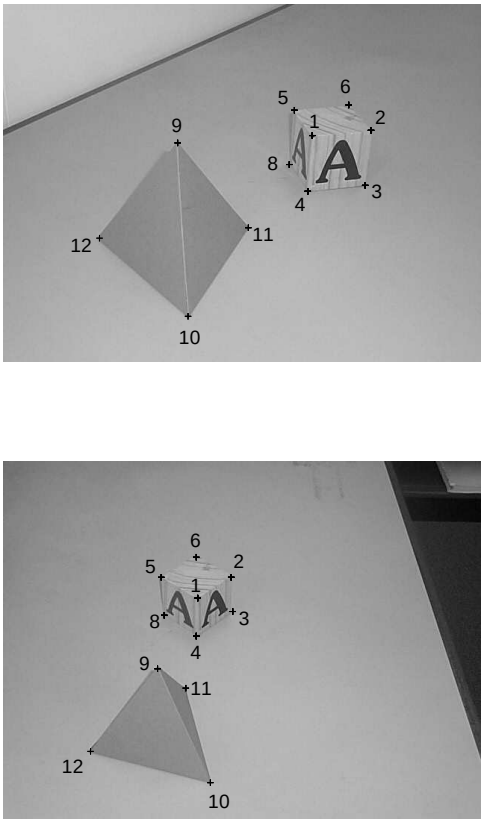

Figure 4.24: Identified points in real images. I selected a set of 11 points from each image and numbered them sequentially. Note that image point number 7 does not exist as it is the eighth corner point in the cube.

4.4.2 Affine reconstruction

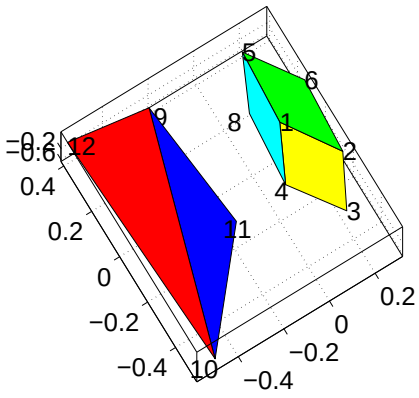

Figure 4.25: Reconstructed 3D space from first solution.

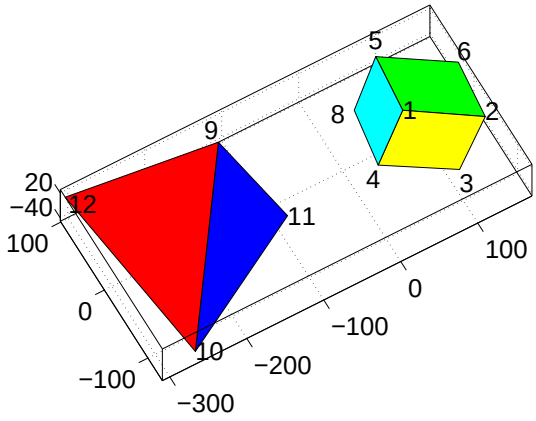

Figure 4.26: Reconstructed 3D space from second solution.

We can see that the centroid from each 2D image has been mapped to the 3D space to form the centroid in the reconstruction. Although the original images contain a cube and a tetrahedron, a relationship cannot be established for features that do not appear in the two images. For example, we cannot assume that the cube has a point 7 that makes the whole structure to be cubical or the points 11 and 12 have a direct link between themselves to form the structure that appears to be a tetrahedron. Therefore, in this case, affine reconstruction involves the formation of 3D affine space only with the available information.

CHAPTER 5

Metric Reconstruction and Texture

Thus far we have followed a mechanism for reconstructing the affine 3D structure for a given set of images. Invariant properties of affine structure gave us an opportunity to visualize parallel planes and lines. However, we can proceed a step further to reconstruct the metric structure from the images to give a true 3D structure of the world scene.

In this chapter, I present a simple algorithm based on [7] for metric reconstruction using a series of images and their corresponding affine structures. I use synthetic and real images we are already familiar with, for metric reconstruction and texture mapping.

5.1 Metric reconstruction

5.1.1 Method

I explain a procedural approach to metric reconstruction as follows:

1. The method for metric reconstruction is based on three assumptions:
   (a) One point, chosen to be the coordinate origin, exists across all images. This point may or may not be a feature point of the structure considered for reconstruction.
   (b) Three other points in orthogonal directions in the real world to the assumed origin point exist in all images.
   (c) Ratios of lengths between the origin and the three points from real world are known, in advance.
2. Firstly, we identify the four points required, in all images, using a manual process. This creates a coordinate frame across all images.

3. The factorization algorithm outlined by [3] is employed to create the affine structure of the identified coordinate frame.

4. The units of length from the origin to each of the three points in affine frame are calculated. Ratios between lengths in the affine frame and real world form a direct relationship as illustrated below in Figure 5.1.

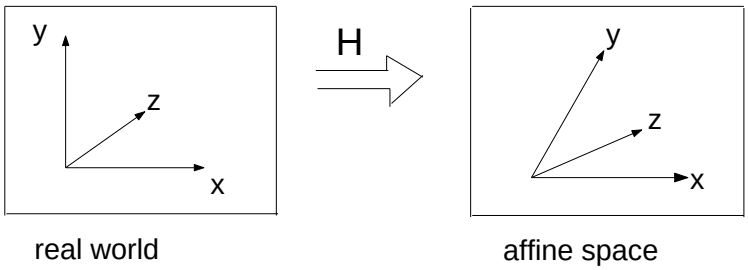

Figure 5.1: Transforming coordinate frame from real world to affine space.

5. I deduce matrix H from the available ratios between real world and affine space.

6. If matrix H transformed real world coordinate frame to affine space coordinate frame, then the inverse of matrix H should transform affine space point to world points. This forms the basis for the algorithm for metric reconstruction.

7. This method is primarily dependent on accurately identifying four points that form the coordinate frame in all images. Also, real world information is required to discover three lines in orthogonal directions and to calculate their ratios. Next, I discuss results generated by this mechanism for metric reconstruction.

5.1.2 Results—synthetic images

I used four images from the rotated object for this exercise and defined the coordinate frame for the synthetic object: point 1–origin; origin to point 4–x direction; origin to point 5–y direction; origin to point 2–z direction. Figure 5.2 shows the identified coordinate frame from the first and the fourth images.

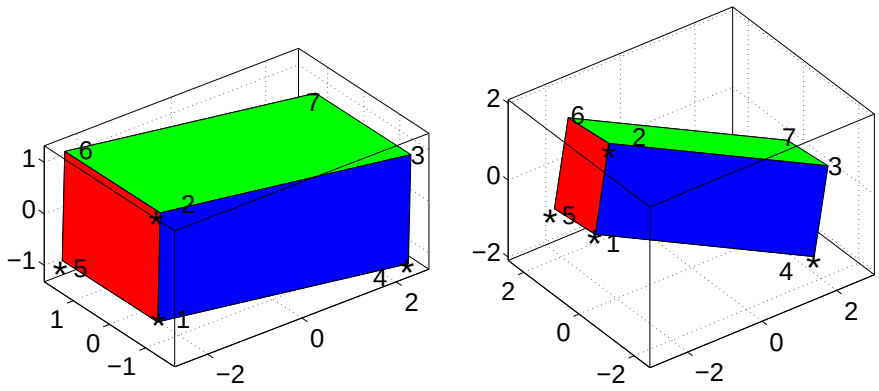

Figure 5.2: Four points 1,2,4,5 identify the coordinate frame on the image.

I calculated the transformation matrix H based on the method outlined above. Although the affine reconstruction algorithm provides two solutions, due to transformation, the metric reconstruction method generates a single set of 3D coordinates. This confirms that there can be only one structure in existence from a metrical point of view. The transformed structure from affine to metric is shown in Figure 5.3, in comparison with the original object.

	Original 3D Object	Metric Reconstruction
Height *(distance 12)*	2	2.0072
Width *(distance 15)*	3	3.0499
Length *(distance 14)*	5	4.9635

Table 1: Table comparing dimensions of original object and its metric reconstruction.

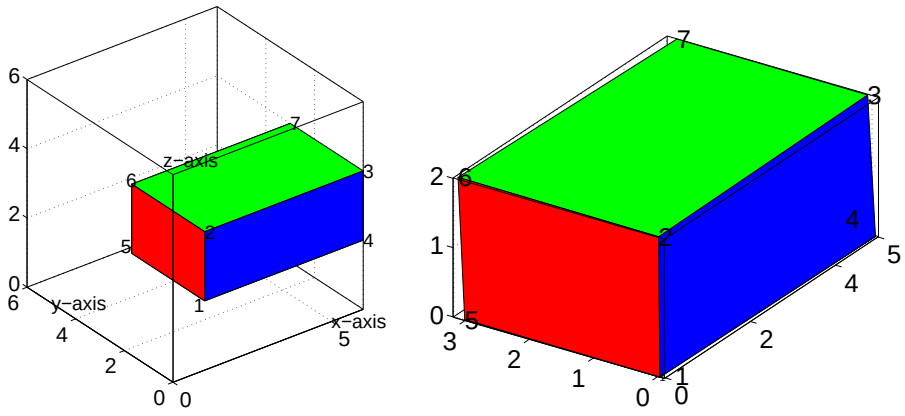

Figure 5.3: Original object (left) in comparison with the metric reconstruction (right). The latter has the same dimensions for ratios length:height:width as the original object, as indicated by Table 1. Since the origin for the coordinate frame has been redefined, it now lies on image point 1 in the metric reconstruction.

5.1.3 Results—real images

The pair of real images taken for affine reconstruction were used in this exercise. I treated the cube as the coordinate frame using image point 4 (Figure 4.24) as the origin. As the cube has uniform dimensions in all three, x, y, z directions, I considered $distance_{43}$ to be x, $distance_{48}$ to be y and $distance_{41}$ to be z. The metric reconstruction of the tetrahedron and the cube is shown in Figure 5.4, comparing the reconstruction with a real image. Figure 5.5 shows the comparison of an image taken from a different angle and the corresponding metric reconstruction from the same angle.

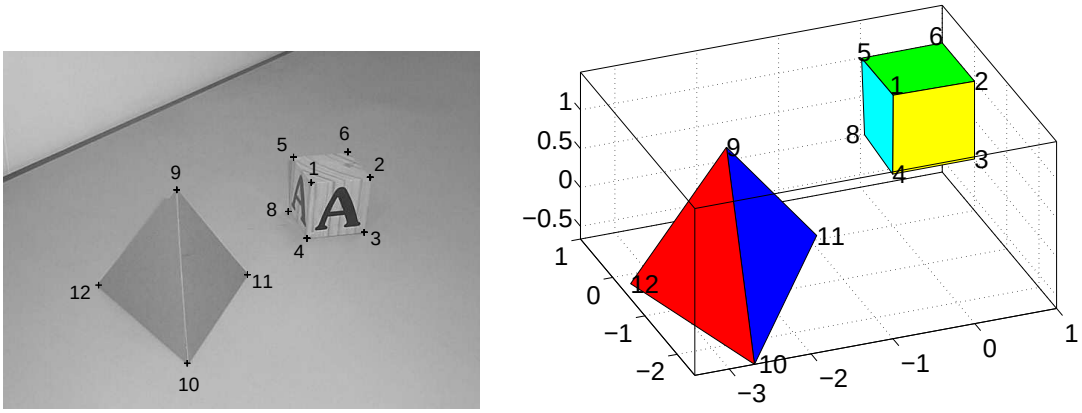

Figure 5.4: A comparison of real image and its metric reconstruction. Point 4, at the base of the cube is treated as coordinate origin.

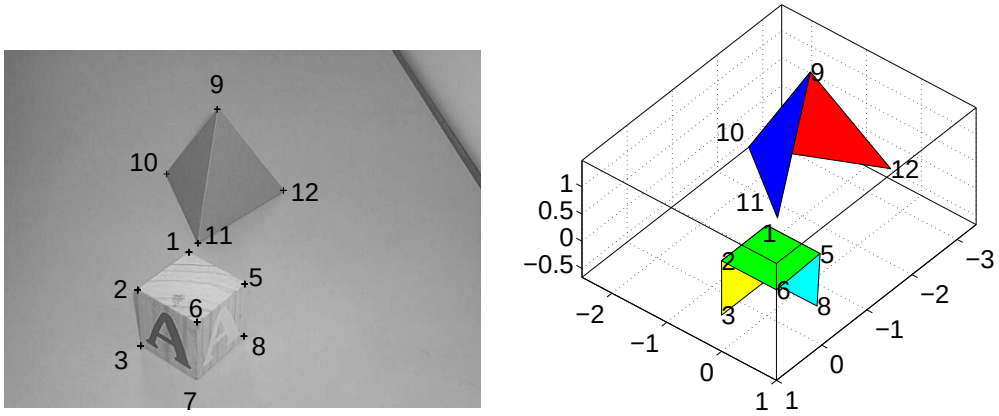

Figure 5.5: Metric structure from a different angle–the real image shows the point 7 of the cube that was hidden from Figure 4.24. However, as this information was not available for the reconstruction algorithm, point 7 is not shown in the metric reconstruction.

5.2 Texture mapping

Texture mapping is the notion of rendering surfaces of a modelled 3D object to reflect a true representation of colours and texture from its original structure. The resulting texture on the surface of the modelled object has a high definition of colour variation and minute detail. In presenting the results I only focus on texture mapping objects reconstructed from real images. The reason for not employing objects from synthetic images is because the definition of colour and brightness on synthetic objects are uniform, and do not highlight the properties of surface texture.

5.2.1 Texture mapped affine reconstruction

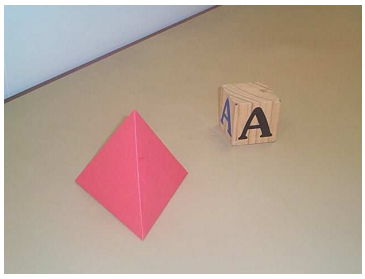

Figure 5.6: Image of the two objects captured from a camera angle.

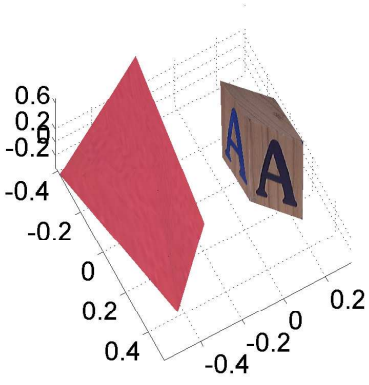

Figure 5.7: Affine reconstruction from first solution with texture mapping. The reconstructed objects are rotated to have a similar view shown by Figure 5.6.

5.2.2 Texture mapped metric reconstruction

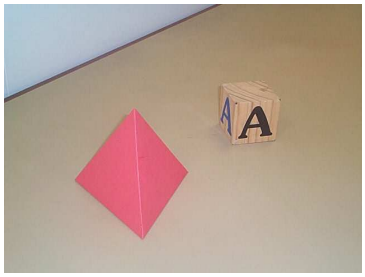

Figure 5.8: Image of the two objects captured from a camera angle.

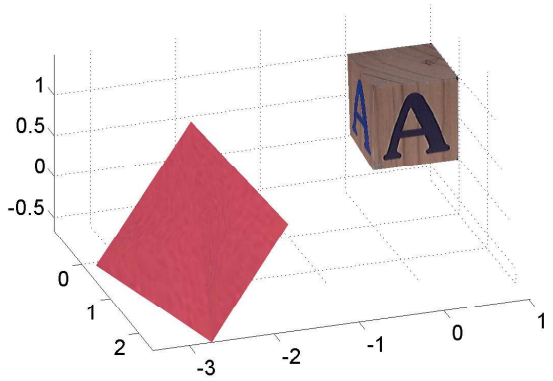

Figure 5.9: Metric reconstruction with texture mapping. The reconstructed objects are rotated to have a similar view shown by Figure 5.8.

CHAPTER 6

Real Image Sequences

The objective of this chapter is to identify feature points in a given series of 2D images analogous to a video clip without the need of manual intervention and then perform an affine reconstruction of the object in focus.

6.1 Method

The procedure for using the tracking programme is as follows:

1. I used the Kanade-Lucas-Tomasi (KLT) tracker [1] to identify feature points in a given series of images. Each detected feature point denotes a (x,y) position that corresponds to all images.

2. When using the KLT tracker, it was vital to keep the camera movement to a minimal angle between one image frame and another. As the KLT tracker can be customised to adjust the tracking context, I modified the KLT tracker to detect feature points on one image based on the movements of feature points in the previous image. Therefore, the tracked feature points cascaded down the image sequence and the final image contained all points relevant to all images in the sequence.

3. These image points were then redefined with respect to their centroid as explained in Chapter 4 and applied the factorization algorithm. Through this, I obtained a collection of affine 3D points.

4. I used the initial collection of (x,y) features to build a relationship matrix adjoining the vertices given by Delaunay triangulation. The same matrix was then used to join the corresponding vertices for the 3D affine coordinates creating a mesh diagram.

6.2 Tracked feature points

An image sequence of 70 images of the real object were taken for this exercise. The KLT tracker detected 300 feature points across all images that were used for reconstruction. Figure 6.1 demonstrates detected feature points across three selected images in the image sequence.

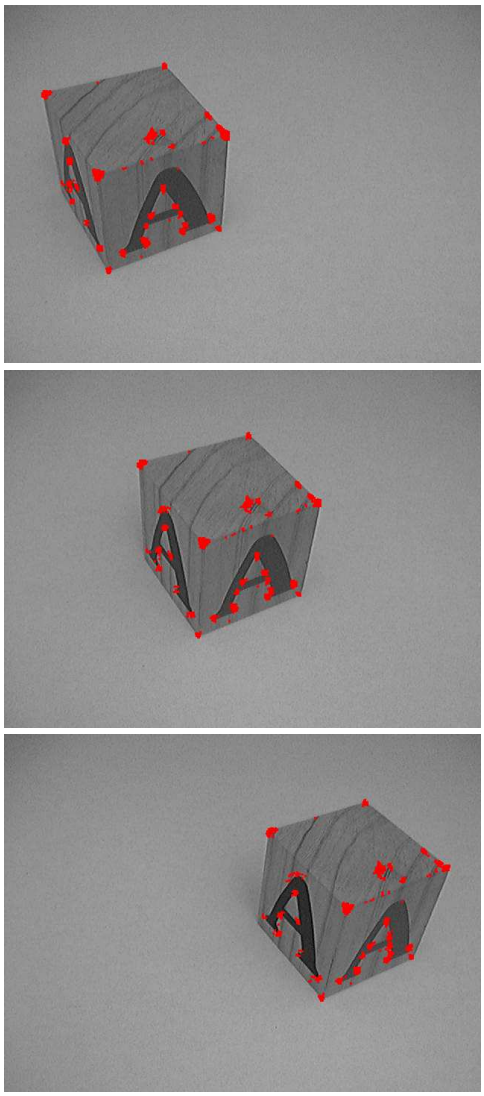

Figure 6.1: Feature point detection by the KLT tracker across three real images. Conglomeration of points indicate the outer perimeter of the 3D object.

6.3 Affine reconstruction from feature points

Figures 6.2 and 6.3 show the 3D mesh diagrams from the two solutions for 3D affine coordinates–Equations 4.13 and 4.14–using information from Delaunay triangulation.

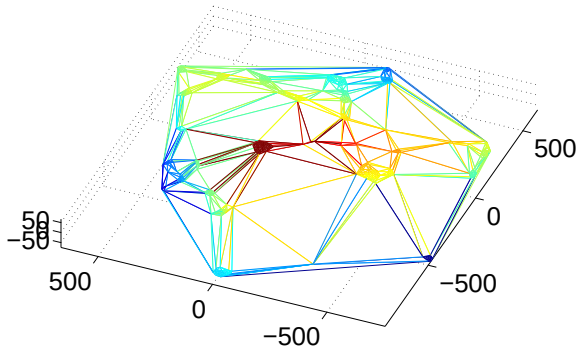

Figure 6.2: Mesh diagram for 3D affine structure from solution 1. Delaunay triangulation demonstrates the best attempt at connecting the feature points of the given 3D cloud of points.

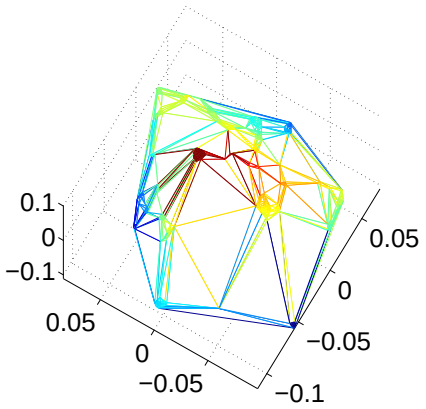

Figure 6.3: Mesh diagram for 3D affine structure from solution 2. Delaunay triangulation demonstrates the best attempt at connecting the feature points of the given 3D cloud of points.

CHAPTER 7

Conclusion

In this chapter I intend to review the results achieved throughout the project, highlighting the framework for affine reconstruction. I discuss further work involved in this subject area and present my final conclusions.

7.1 Final results

I showed how a sequence of affine images can be used for reconstructing the subject concerned, in affine space. I employed the SVD as the mathematical function for this reconstruction and demonstrated its effectiveness under noise and non-affine images with perspective projection.

Furthermore, I showed the method for achieving metric reconstruction by transforming the affine structure. The overall procedure can be condensed into a flow diagram as follows:

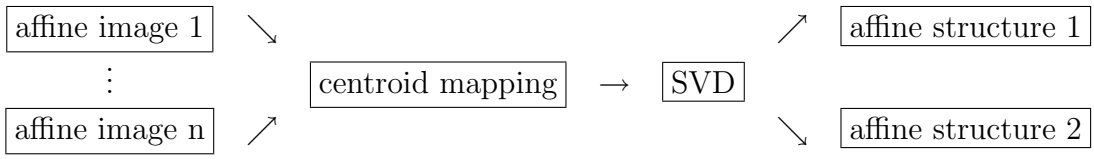

Figure 7.1: Flow diagram for affine reconstruction

7.2 Further work

Kahl and Heyden [5] demonstrate a method for handling the missing data problem where all image points are not visible from every image. They derive a corresponding matching constraints algorithm for the affine camera. In this thesis I explicitly stated that the matching image points from images must exist in every image for that image to be taken into consideration. However, an area to be pursued is not to discard images that may not always contain all image points. Hence, an algorithm is needed to either calculate the absent points from the image or to disregard these points when taking the SVD of the measurement matrix.

Furthermore, the factorization algorithm can be extended to create the complete affine structure by joining two half structures. For example, in creating a cubic structure, if two half structures are constructed, then by matching the corresponding edges and vertices the completed affine cube can be reconstructed.

The mechanism to determine the metric structure I explained in this thesis, is error prone as it involves user intervention. Hartley and Zisserman [3] explain a more robust approach to metric reconstruction from affine structure, using the image of the absolute conic.

7.3 Final conclusion

Theoretically, images used for affine reconstruction using the SVD must have orthographic projection. All points from affine images must be visible from every image for the points to be transformed into affine space. Two affine images are sufficient for an affine reconstruction of common object space covered by the images. However, the angular difference between the images governs the depth of affine reconstruction. Although the reconstruction is affine, invariant properties of affine structure give us much practical information about the dimensions of the structure in real world.

The SVD is a robust mathematical function to employ when reconstructing the affine structure using the factorization algorithm. The unique properties demonstrated by the SVD make it highly tolerant to images with noise. In practice, this feature gives us the ability to use images with a weak perspective for reconstruction.

In conclusion, the factorization algorithm of Tomasi and Kanade [12] presents a simple yet very effective technique for reconstructing the 3D structure of object space by using two images from uncalibrated cameras.

APPENDIX A

Original research proposal

Title: Affine reconstruction from multiple views using Singular Value Decomposition

Author: Mohan Obeysekera

Supervisor: Dr Peter Kovesi

Background

As a broad definition, a camera can be termed as a mapping between the object space and an image plane. The general projective camera is used to model central projection, whereby points on one plane are mapped to points on another. Furthermore, the general projective camera can be primarily divided into two categories of models namely, cameras with a known, finite centre and cameras with centre at infinity. The latter can further be subdivided into affine cameras and non-affine cameras [3]. An affine camera maps points at infinity in the object space on to points at infinity in the image plane. This is commonly known as orthogonal projection.

The underlying principle in affine reconstruction is to detect the plane at infinity in the given images. Once the plane at infinity is located, it can then be correctly placed in such a manner that points/lines at infinity in object space are mapped to points/lines in the image plane. This transformation that is involved in placing the plane at infinity is affine transformation. Therefore, the true reconstruction that has undergone affine transformation is said to be an affine reconstruction [3].

Koenderink and van Doorn [7] discuss, in depth, creating an affine structure from motion by firstly outlining a theoretical view of four points and then illustrating it through an example. Affine reconstruction has also been discussed by [3], using factorization algorithm of Tomasi and Kanade [12].

Aim

Affine reconstruction involves finding a 3D model such that when it is projected into the image plane, the error between the projected model points and the observed points is minimized. However, due to the nature of this projection, there exists a discrepancy between the projected image points and observed image points. As the former is used for affine reconstruction, the reconstructed version of the object space would contain afore mentioned discrepancies, giving an inaccurate view.

The objective of this research project is to build an affine reconstruction from multiple images by minimising the geometric error between the projection of the reconstruction onto the image plane and the observed image points. Once this geometric error (also known as the reprojection error) is minimized, a better, accurate reconstruction of the object space can be obtained.

Affine reconstruction has the potential to be taken a further step forward by having a metric reconstruction either using metric information from object space or using auto-calibration methods.

Method

Firstly, the focus is placed on images acquired by affine cameras and understanding their properties in comparison with perspective images. This exercise is used to identify and re-establish the uniqueness of these images. Secondly, as explained by [3], using a transformation matrix, inhomogeneous image points are obtained using inhomogeneous world points and a translation vector. This is then applied to all the world points and a matrix of (x,y) inhomogeneous image points is formulated.

Based on the theory that, an affine camera maps the centroid of a set of 3D points to the centroid of their projections [3], the matrix of (x,y) points can be expressed as a linear set of equations which can be minimized using Singular Value Decomposition (SVD), resulting in a matrix $\mathbf{W}$. Furthermore, another matrix $\hat{\mathbf{W}}$ is now written with a strong resemblance to $\mathbf{W}$ in the Frobenius norm. It can then be deduced that the geometric error is equivalent to the difference between $\mathbf{W}$ and $\hat{\mathbf{W}}$.

Schedule

Date	Deadline
Fri, 01st November 2002	Understand literature–Tomasi Kanade factorization algorithm; Singular Value Decomposition.
Mon, 04th November 2002	Copies of updated project proposal due to course coordinator and supervisor.
Fri, 20th December 2002	Perform experiments with simulated data for orthographic projection with and without noise.
Fri, 31st January 2003	Perform experiments with simulated data for perspective projection and determine the errors involved.
Fri, 28th February 2003	Experimentation with real images from real cameras: calibrated cameras; uncalibrated cameras.
Fri, 14th March 2003	Surface reconstruction and rendering from obtained image points.
Mon, 31st March 2003	Commencement of dissertation.
Mon, 28th April 2003	Completion of first draft of dissertation.
Fri, 02nd May 2003	First draft of dissertation due to supervisor.
Wed, 14th May 2003	First draft returned by supervisor; seminar titles and abstract due to course coordinator.
Mon, 02nd June 2003	Project seminars.
Mon, 09th June 2003	Two copies of final project dissertation due in the School Office.
Mon, 23rd June 2003	Dissertation returned for final corrections.
Mon, 30th June 2003	Corrected dissertation returned to the School Office.

Bibliography

[1] Birchfield, S. (1996), *KLT:An Implementation of the Kanade-Lucas-Tomasi Feature Tracker*, [Online], Available from: http://robotics.stanford.edu/∼birch/klt [02 April 2003].

[2] Faugeras, O., and Luong, Q.-T. *The Geometry of Multiple Images*. MIT Press, 2001.

[3] Hartley, R., and Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[4] Jennings, A. *Matrix Computation for Engineers and Scientists*. John Wiley and Sons, 1977.

[5] Kahl, F., and Heyden, A. Affine structure and motion from points, lines and conics. *International Journal of Computer Vision Vol. 33*, No. 3 (1999), 163–180.

[6] Khademhosseini, A. (2002), *Singular Value Decomposition (SVD) tutorial*, [Online], Available from: http://web.mit.edu/be.400/www/SVD/Singular_Value_Decomposition.htm [16 February 2003].

[7] Koenderink, J. J., and Van Doorn, A. J. Affine structure from motion. *Journal of the Optical Society of America(A) Vol. 8*, No. 2 (February 1991), 377–385.

[8] Kwon, Y.-H. (1998), *Singular Value Decomposition*, [Online], Available from: http://kwon3d.com/theory/jkinem/svd.html#svd [20 February 2003].

[9] Mundy, J. L., and Zisserman, A. *Geometric Invariance in Computer Vision*. MIT Press, 1992.

[10] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992.

[11] Reid, I. D., and Murray, D. W. Active tracking of foveated feature clusters using affine structure. *International Journal of Computer Vision Vol. 18*, No. 1 (1996), 41–60.

[12] Tomasi, C., and Kanade, T. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision Vol. 9*, No. 2 (1992), 137–154.