

3D Model Reconstruction with noise filtering using Boundary Edges

Lau Tak Fu

Name of supervisor: Dr. K.H. Wong

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Philosophy

in

Computer Science and Engineering

The Chinese University of Hong Kong

October 2003

The Chinese University of Hong Kong holds the copyright of this thesis. Any person(s) intending to use a part of or whole of the materials in the thesis in a proposed publication must seek copyright release from the Dean of the Graduate School.



Abstract of thesis entitled:

3D Model Reconstruction with noise filtering using Boundary Edges

Submitted by

Lau Tak Fu

for the degree of

M. Phil.

at The Chinese University of Hong Kong in (Oct , 2003)

Abstract

Structure from motion algorithms, for example the two-pass interleave bundle adjustment approach, can generate the pose and model. However, this approach assumes that good feature correspondences are available. But noise and feature mismatch may corrupt the correspondences and result in serious deformation of the model generated. In this thesis, a feature filter by using the silhouette clipping approach is proposed to select a good feature set from noisy data for model recovery of an object on a turntable. Synthetic as well as real images were tested using this method with good results.

摘要

很多移動影像結構重組都是把姿勢和模型結構分開處理。透過利用轉換束組調節方式，模型的姿勢和結構可以在同一過程被推算。這方式提供更佳效率。不過，這方式假定良好特徵點對應能夠被找到。噪音和特徵點錯配可能導致嚴重模型扭曲。在這論文裡，一個使用輪廓剪影技術的特徵點過濾器被提出來選擇一組好的特徵點。由一組從一個轉台上的物件錄得來的含噪音的特徵點。電算機合成影像及真實影像亦用來測試這方法。兩者都有好的結果。

Acknowledgment

I would like to show my deep thankfulness to my thesis supervisor, Prof. K.H. Wong for his inspirations and guidance during the proceeding of the project. I would also like to thank Prof. Pheng-Ann Heng and Prof. HanQiu Sun for being my thesis examiner. Also, I cordially thank Prof. W.K. Kan and Ms. Esther Lee for their supervision and support during the early stage research work. And I thank my fiancé Ann Ming Lui, my parents Mr. W.C. Lau and Mr. F.L. Chow, my brother Dr. Edward T.W. Lau and his wife Mrs. Anna Lau and my lovely sister Ms. Y.P. Lau sincerely for their prayer and emotional support during the writing of the thesis. Finally I would like to give all my thanks to the Lord who is the stream of strength in my life.

Table of Contents

1 – Introduction.....	9
1.1 Scope of the work.....	9
1.2 Main contribution.....	11
1.3 Outline of the thesis.....	12
2 – Background.....	14
2.1 Three dimensional models from images.....	14
2.2 Un-calibrated 3D reconstruction.....	14
2.3 Self calibrated 3D reconstruction.....	16
2.4 Initial model formation using image based.....	18
2.5 Volumes from Silhouettes.....	19
3 - Initial model reconstruct the problem with mismatch noise.....	22
3.1 Perspective Camera Model.....	24
3.2 Intrinsic parameters, Extrinsic parameters and camera motion.....	25
3.2.1 Intrinsic parameters.....	25
3.2.2 Extrinsic parameter and camera motion.....	27
3.3 Lowe’s method.....	29
3.4 Interleave bundle adjustment for structure and motion recovery from multiple images.....	32
3.5 Feature points mismatch analysis.....	38

4 - Feature selection by using look forward silhouette clipping.....	43
4.1 Introduction to silhouette clipping.....	43
4.2 Silhouette clipping for 3D model.....	45
4.3 Implementation.....	52
4.3.1 Silhouette extraction program.....	52
4.3.2 Feature filter for alternative bundle adjustment algorithm.....	59
5 - Experimental data.....	61
5.1 Simulation.....	61
5.1.1 Input of simulation.....	61
5.1.2 Output of the simulation.....	66
5.1.2.1 Radius distribution.....	66
5.1.2.2 3D model output.....	74
5.1.2.3 VRML plotting.....	80
5.2 Real Image testing.....	82
5.2.1 Toy house on a turntable test.....	82
5.2.2 Other tests on turntable.....	86
6 – Conclusion and discussion.....	89

Figure list

Figure 1 – the flow diagram of feature selection by using silhouette clipping.....10

Figure 3.1 – Flow diagram of alternative bundle adjustment method.....23

Figure 3.2 - Perspective projection of object onto an image.....24

Figure 3.3 - The relationship between pixel size and the value of an image point.....26

Figure 3.4 - the simulation sphere result from the images with and without noise.....42

Figure 4.1 Raw pictures of sphere face.....45

Figure 4.2 the view of a sphere face from reference frame view.....46

Figure 4.3 the view of the sphere from new pose view.....47

Figure 4.4 Use the silhouette to filter the out of bound feature.....48

Figure 4.5 – Flowchart of feature selection by silhouette clipping.....51

Figure 4.6 Examples of the input picture to snake algorithm.....55

Figure 4.7. The color histogram of the target object in Figure 4.6.....57

Figure 4.8 The result of the snake outline tracker.....58

Figure 5.1 - Sample picture of original sphere.....62

Figure 5.2 - Feature extraction for different no condition.....64

Figure 5.3 - Figure point with noise.....65

Figure 5.4 - The radius of the recovered 3D sphere feature distribution.....67

Figure 5.5 – The comparison of radius error between using filter and without using filter.....68

Figure 5.6 - Radius distribution with noise.....70

Figure 5.7 - The 3D feature graph for the recovered model without noise.....74

Figure 5.8 - The 3D features of the recovered model with noise.....76

Figure 5.9 - Sample pictures of robot test.....80

Figure 5.10 - The recovered results with and without using feature filter.....81

Figure 5.11 - Original pictures for toy house.....82

Figure 5.12 - The result of toy house before and after using feature filter.....84

Figure 5.13 - Original pictures for toy robot, flask and tomb respectively.....86

Figure 5.14 - The result of toy house, flask and tomb with and without using feature filter (wire frame).....87

Figure 5.15 - The result of toy robot, flask and tomb with and without using feature filter.....88

Chapter 1: Introduction

1.1 Scope of the work

In this thesis we investigate a way to select good features for multiple view 3D reconstruction. It is useful to enhance the result of existing model reconstruction algorithms. Specifically, my work is used to enhance the result of the reconstruction of an object on a turntable.

My implementation is specifically built on the algorithm of interleave bundle adjustment for structure and motion recovery from multiple images [61]. The algorithm has an advantage that the pose and model could be obtained simultaneously and efficiently. However, this algorithm, or any other Structure From Motion (SFM) algorithms, has a weakness that noise and feature mismatch easily distort the recovered model. The aim of my work is to select a better feature set to feed into the interleave bundle adjustment method [61] to improve its accuracy.

A filter is proposed to filter out the bad and mismatch features. That filter takes the interleave bundle adjustment result as its input. Then it uses the silhouette information of the target object to select the correct features and feedback to the interleave bundle

adjustment algorithm again. The process is re-iterated until the quality of the output model is satisfactory. The system is tested with both synthetic data and real images.

Here is an overview of our work:

Given a set of noisy features that is extracted from a sequence of pictures, it is able to recover the 3D model of the object in the movie. However, if the noisy feature set is directly used in the interleave bundle adjustment algorithm for 3D model recovery, the output model would be seriously distorted due to feature mismatch. Because the bad feature points would locate at any places in the picture, therefore a simple 2D filter is normally not be able to filter out those bad feature points.

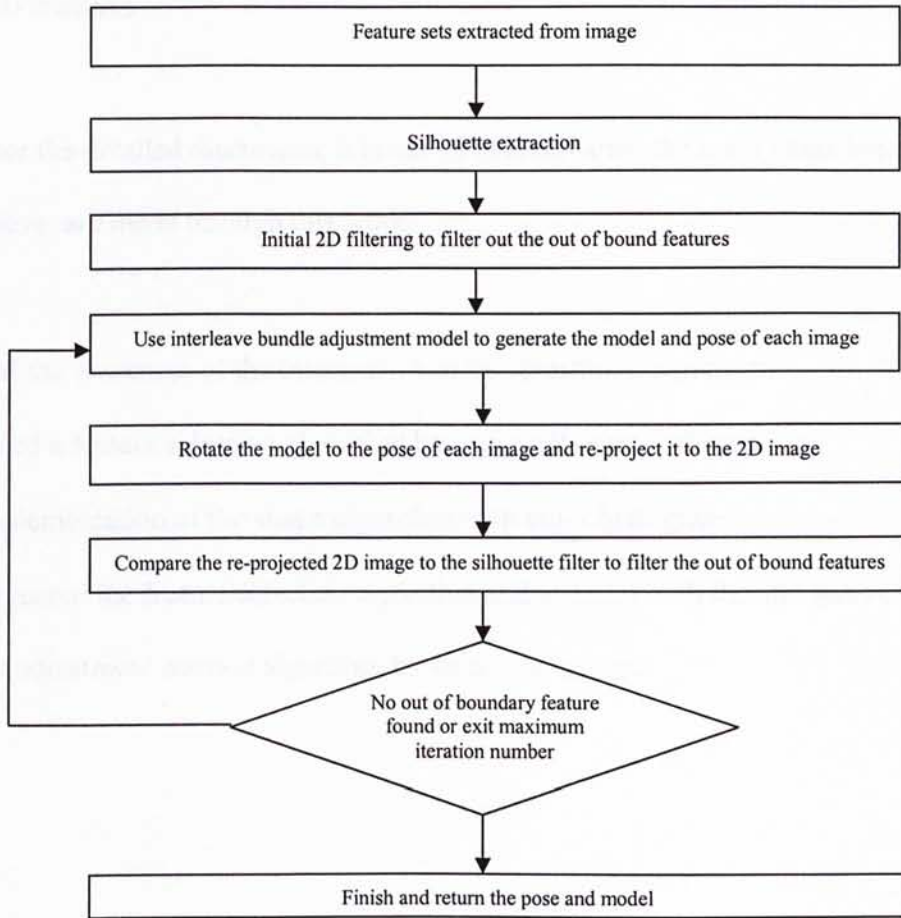


Figure 1 – the flow diagram of feature selection by using silhouette clipping

Therefore, we propose a 3D filter for feature point selection to improve the situation.

Figure 1 shows the basic flow diagram of our proposal. By using our algorithm, we are able to obtain a set of good features to feed into the alternative bundle adjustment algorithm. Our idea is implemented in Matlab code. In a synthetic data test, up to 14% error improvement is found in large noise condition with our algorithm. Our algorithm is also tested in real image cases and the results also have significant improvement.

1.2 Main contributions

Before we enter the detailed discussion, it is useful to summarize the main contributions, which we believe, are made through this work:

- Studied the weakness of the interleave bundle adjustment algorithm.
- Designed a feature selection algorithm by using silhouette information.
- An implementation of the snake algorithm with color histogram.
- Implemented the feature selection algorithm and tested it with the alternative bundle adjustment method algorithm by using real images.

1.3 Outline of the thesis

In Chapter 2 some background of structure from motion (SFM) is included. A study of various work on un-calibrated and calibrated SFM methods (2 views, 3 views and multiple views) are presented. The topics of dense matching and volume from silhouette are also discussed.

A detailed discussion about the interleave bundle adjustment algorithm is included in Chapter 3. The basic perspective projection, Lowe's algorithm and interleave bundle adjustment algorithm are also described. The noise and feature mismatch effect on interleave bundle adjustment algorithm are also discussed.

In chapter 4 a feature filter using silhouette clipping technique is proposed. A discussion about how the silhouette method is used to extract those bad features and how the bad features were pruned out from a 3D model are presented. The implementation details of the feature filter were presented in that chapter. Besides, a silhouette extraction scheme using a snake algorithm with color histogram is also presented. Some experimental results are included to demonstrate how the snake algorithm with color histogram approach overcomes a complex image background to locate the silhouette of the target object.

In Chapter 5 the experimental data of our algorithm is included. A synthetic sphere simulation is presented. The difference between before and after using the feature filter was studied. Real image tests are also presented in this chapter.

The conclusion is presented in Chapter 6. The future work is also discussed.

Chapter 2: Background

2.1 Three dimensional models from images

Stereopsis or Stereo vision, according to Trucco [57], is the construction of a 3D structure and the distance of a scene from two or more images which are taken from different viewpoints.

2.2 Un-calibrated 3D reconstruction

Two views

Early research on structure from motion concentrated on the study of the two views problem [58] [59]. It is assumed that a set of corresponding features can be obtained. By using two frames taken from an object, a fundamental matrix that incorporates the information of the camera motion can be found. When the intrinsic parameters of the camera are known, the extrinsic parameters can be obtained up to a scale factor. After the extrinsic parameters are known, the 3D locations of the features points can also be estimated. However, when the intrinsic parameters of a camera are hidden, it is called un-calibrated structure from motion.

In the field of feature-based structure from motion, two early approaches have been investigated by Faugeras [15] and Hartley [23]. They developed the fundamental matrix techniques that solve the projective structure of the two views geometry. A lot of work has been developed on how to find and use the fundamental matrix [9] [22] [34] [42] and some systems perform well on real image data [54] [55] [66].

Three views

Model recovery using three views have also been studied. This is called the trifocal tensor technique. Spetsakis and Aloimonos [49] developed the calibrated case of trilinearities to transfer line and points. Later the un-calibrated case is studied for points [47], or lines [24] or both [21]. Torr and Zisserman [53] [53] then suggested robust computation methods for trifocal tensor and the properties of this method are well studied by Shashua and Avidan [46].

Multi-views

Many researchers have extended the method for two images to multiple images. One method for multiple images structure recovery begins using two views, then sequentially adding new images to refine the model [5] [9]. A detailed study about the relationship between two views, three views to multiple views structure from motion have been analyzed by Fitzgibbon and Zisserman [17].

Moreover, Heyden [25], Triggs [56] and Faugeras and Mourrain [14] also studied the problem of multiple view images recovery. Some important work has been done for the full perspective camera model, for example Azerbayejani [3], Cui [12], Spetsakis and Aloimonos [49] and Szeliski and Kang [51]. For the un-calibrated cases, one remarkable work is the factorization approach from Tomasi and Kanade [52] based on an affine camera model. An enhancement which is able to solve the missing data case was proposed by Jacobs [26].

To refine the structure from multiple sequential images, Kalman filter is a useful tool [27]. Another approach is bundle adjustment [60] [65]. That idea was incubated by Mohr [39] in an early paper and then it was formally entitled as bundle adjustment [48].

2.3 Self calibrated 3D reconstruction

Up to now our discussion focuses on un-calibrated structure from motion. This approach generates a projective structure output. However, in many applications, the projective structure is not accurate enough and the metric structure of scenes is preferred. A popular method to achieve the metric structure is to assume fixed intrinsic camera parameters.

This problem was first studied by Maybank and Faugeras [37]. Then Luong [34] and Zeller [63] [64] proposed a pair-wise calibration method which was inspired by Kruppa equations [30].

Another approach is to compare the intrinsic parameters difference between computation and factorization of the camera projection matrices. Hartley [19] obtained the metric structure by minimizing that difference. A few years later Trigg [68] proposed an absolute quadric approach.

Some researchers simplify the problem by solving a particular case of the problem. They study the problem based on a restricted motion, for examples, a purely translating camera is studied by Moons [40] [41], a stratified approach is studied by Armstrong [1], a rotating camera case is studied by Hartley [20] and a planar motion is studied by Zisserman and Hartly [2] and Faugeras, Quan and Sturm [13].

In addition, Zisserman [189] proposed to impose, a posteriori, constraints on intrinsic parameters. Pollefeys [43] demonstrated how to use Kalman filter and bundle adjustment method to filter the feature outliers in long images sequences. Chang and Wong [61] proposed an interleave bundle adjustment method to acquire both the pose and model information simultaneously.

2.4 Initial model formation using image based

Dense stereo matching

Another approach is to compute the depth map by using correlation, or dense stereo matching. That method generates textured 3D models but the results are often not detailed enough. Moreover, some important scene features are missing so that the recovered models are often incomplete. However, an interesting aspect is that when a model is obtained from a set of sequential images, the poses of the camera are also known. That output could be feed into existing stereo matching algorithms to obtain a more detailed 3D model.

Matthies, Szeliski and Kanade[36] introduced an early work to estimate depth from image sequences. They suggested a pixel-based algorithm that estimates depth and depth uncertainty at each pixel and incrementally refined these estimates over time. Kalman filtering is used to extrapolate and update the pixel-based depth representation.

Comparing to feature-based Kalman filtering algorithm, this method is an effective way to extract depth from lateral camera translations.

The advantage of the dense map method is that it gives detailed surface estimates and the accuracy of the multi-view aggregation is improved. However, in stereo processing, the lateral camera translation creates a tradeoff problem between precision and accuracy in matching. If the baseline of the camera is short, the estimated distance would be less

precise due to narrow triangulation. For more precise distance estimation, a longer baseline is preferred. Unfortunately, the tradeoff of this improvement is that a larger disparity searching range is needed. For this, Okutomi and Kanade[29] introduced a Sum of the Sum of Square Different (Sum of SSD) SSSD-in-inverse-distance function to exhibit a unique and clear minimum at the correct matching position.

Another important work about correlation including the level-set algorithm was proposed by Faugeras and Keriven [16]. That idea extends the snake method by Kass and the context partial differential equation PDE driven evolving curves. A curve is like a 2D snake and is evolved by a PDE . The matching criterion is to find a suitable function so that the integral error $|I_1(m_1) - I_2(m_2)|^2$ is minimized. Their work generates high quality reconstruction and even when occlusion occurs. However, some areas of their work need improvement, which included bad complex surface performance, regularization biases and no guarantee on unconstrained camera configurations and arbitrarily-shaped scenes.

2.5 Volumes from Silhouettes

Besides feature-based structure from motion, some researchers studied the use of silhouette information of an image to recover the model information.

A lot of work in this area assumes that several views of the silhouette of an object are found. An early work on this topic was proposed by Giblin and Weiss[18]. The model is developed from the envelope of its tangents. However, Giblin and Weiss's work is on simulation stage only. A practical work is done by Ronen and Ullman[4]. They showed that it is possible to predict a new appearance of a viewer-centered object from any given viewpoint by using the magnitude of the image curvature of an object's silhouette. They made an assumption that the projection of the camera had to be orthographic. Blake and Cipolla [7][8] assumed that a camera moves continuously with a known motion, they parameterized the surface with respect to the arc length along the rim and the time. Their result is quite sensitive to the noise of the boundary. Instead of using the curvature of the boundary, they computed the difference between two radii of curvatures, or the differential curvature. Besides, Lim and Binford [32] described an object's surface by using two views of the object. Régis and Faugeras [44] studied the occluding contours problem of an image. They identified the occluding contours from triplets of images and modeled the object as the envelope of its tangent plane.

More recent work about structure from motion using silhouette information has been done. Mendonça, Wong and Cipolla[38] addressed the problem on using the profiles information to develop the model of an object rotating on a turntable in front of a single camera. The main improvement of their work is that it does not depend on point and line correspondences. Another work was presented by Wong and Cipolla[62] which

overcomes enhancement the problems occlusion when adding new views as the turntable rotates since part of the structure is invisible under circular motion.

Up to now we already studied the development of the field of structure from motion.

From the review we found that many of the algorithms for solving the model acquisition and pose estimation simultaneously were complex and slow. The approach from Chang and Wong [61] provides a more efficient method by using interleave bundle adjustment for generating both two results at once. However the generated model is easily distorted because of the feature mismatch in input images sequence. Therefore we developed a silhouette clipping method to enhance the structure reconstruction result of the interleave bundle adjustment algorithm.

Chapter 3: Initial model reconstruct the problem with mismatch noise

In the last chapter, we have a brief revision about the research field of structure from motion. In this chapter, we will discuss the interleave bundle adjustment algorithm and the weakness of this algorithm.

A short summary about the interleave bundle adjustment algorithm will be described here first. Figure 3.1 shows the flow diagram of the basic idea. A set of features was extracted from each image and an initial model (e.g. a plane model) is assumed. The feature sets and model will be used to estimate the pose of each frame. The pose information will then be used to predict a better model by using Newton's method with minimal residual image error. The pose estimation and model estimation processes are iterated until the total 2D re-projection error is small enough or too many iterations have been executed

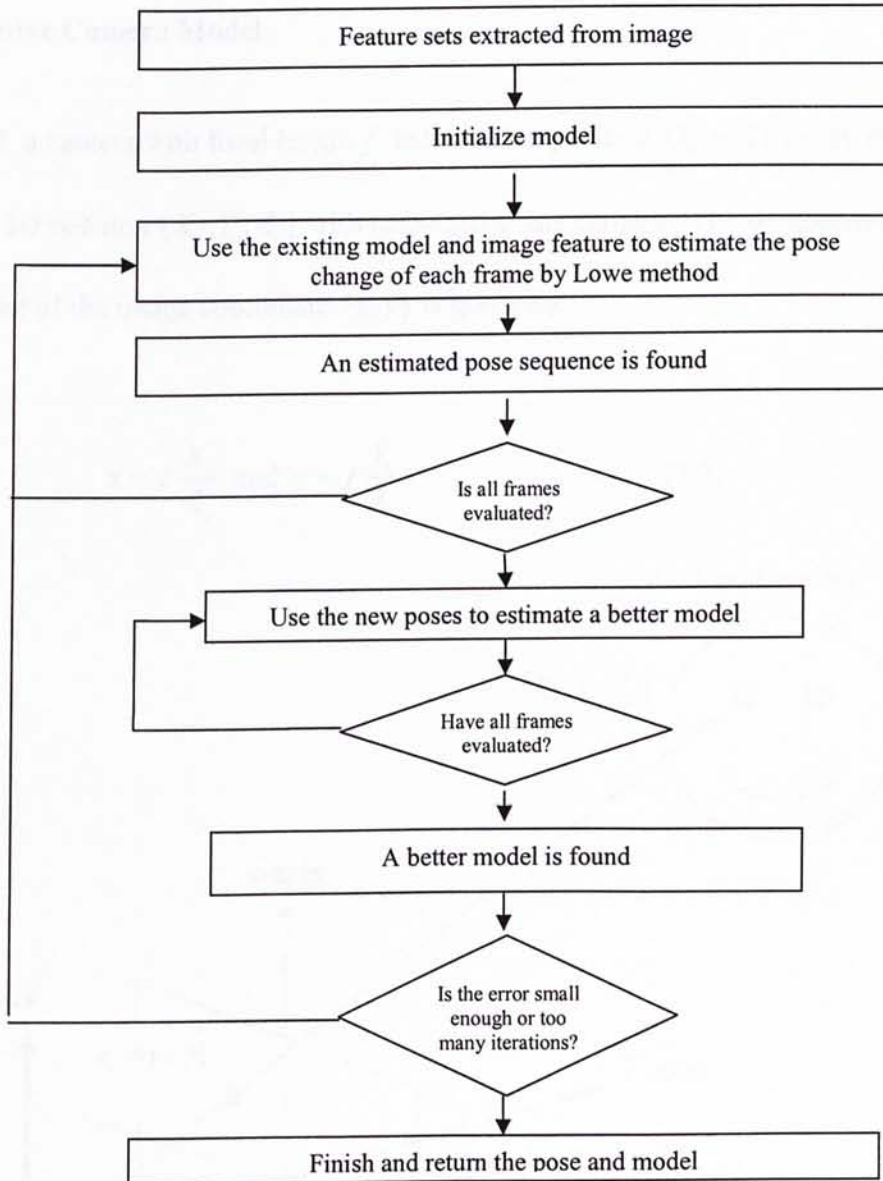


Figure 3.1 – Flow diagram of the interleave bundle adjustment method

3.1 Perspective Camera Model

In figure 3.2, a camera with focal length f is located at position O_w and a point P is located at a 3D position $[X_c, Y_c, Z_c]$ with respect to the camera. The perspective camera model of the image coordinate (x, y) is given by:

$$x = f \frac{X_c}{Z_c} \text{ and } y = f \frac{Y_c}{Z_c} \quad (3.1)$$

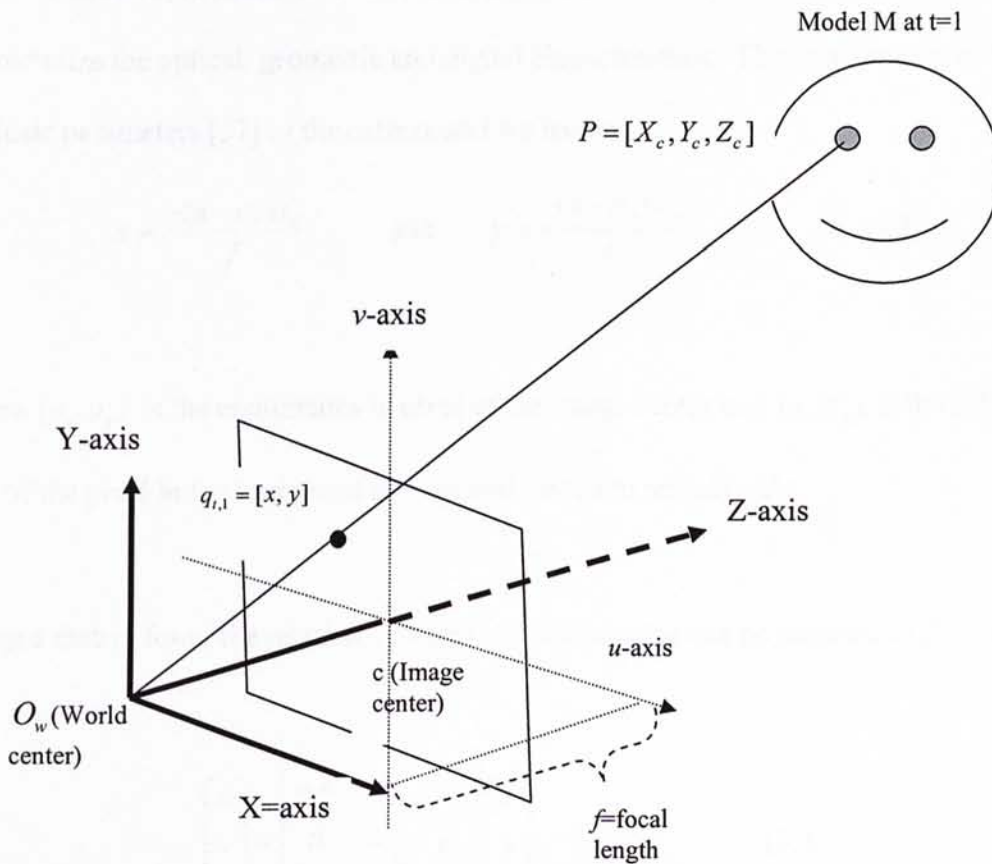


Figure 3.2 - Perspective projection of an object onto an image

3.2 Intrinsic parameters, Extrinsic parameters and camera motion

3.2.1 Intrinsic parameters

For a CCD camera, an image point (u, v) that is captured from the computer does not directly represent the physical coordinate in the retinal plane. The value of this image point depends on the size and shape of the pixels and the position of the CCD chip in the camera. Figure 3.3 shows the relationship between pixel size and the value of an image point. Therefore, in addition to the focal length, there are a set of parameters to characterize the optical, geometric and digital characteristics. They are defined as the intrinsic parameters [57] of the camera and we have:

$$x = \frac{-(u - o_x)s_x}{f} \quad \text{and} \quad y = \frac{-(v - o_y)s_y}{f} \quad (3.2)$$

where (o_x, o_y) is the coordinates in pixel of the image center and (s_x, s_y) is the effective size of the pixel in the horizontal and vertical direction respectively.

Using a matrix form, the relation between two coordinates can be denoted as follow:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{f}{s_x} & 0 & o_x \\ 0 & -\frac{f}{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.3)$$

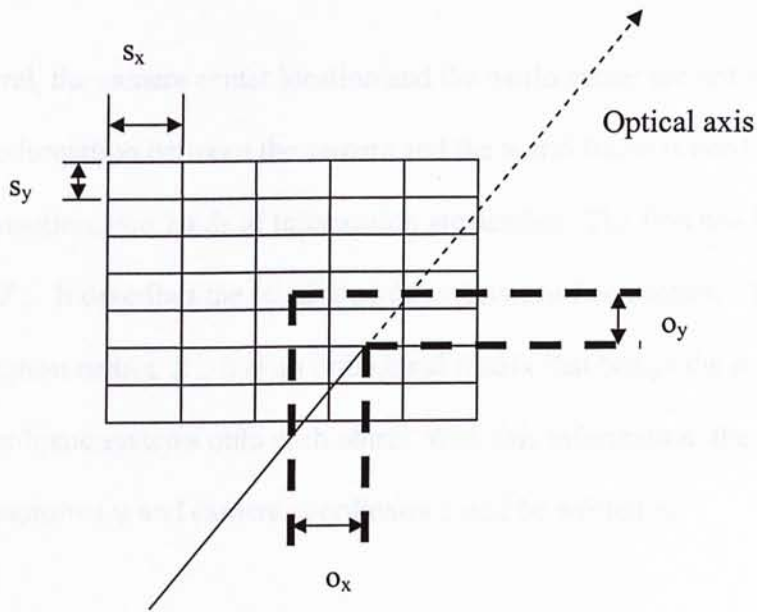


Figure 3.3 - The relationship between pixel size and the value of an image point

3.2.2 Extrinsic parameter and camera motion

In general, the camera center location and the world center are not the same. Therefore, the transformation between the camera and the world frame is needed. To describe this transformation, two kinds of information are needed. The first one is a 3-D translation vector T . It describes the relative position between two centers. The second one is a 3 by 3 rotation matrix R . It is an orthogonal matrix that brings the corresponding axes of two coordinate systems onto each other. With this information, the relation between world coordinates and camera coordinates could be written as:

$$P_c = R(P_w - T) \quad (3.4)$$

where P_w is the world coordinate.

This is also called the extrinsic parameters [57] of a camera. Then we obtain:

$$\begin{aligned} -(u - o_x)s_x &= f \frac{R_1^T (P_w - T)}{R_3^T (P_w - T)} \\ -(v - o_y)s_y &= f \frac{R_2^T (P_w - T)}{R_3^T (P_w - T)} \end{aligned}$$

where $R_i, i = 1, 2, 3$ is a 3-D vector formed by the i -th row of the matrix R . By using the matrix form, the equation can be rewritten as:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} R & -R^T T \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

which is simplified to

$$P_c = M_{ext} \begin{bmatrix} P_w \\ 1 \end{bmatrix}$$

This equation also describes a camera motion from one point P_1 to another point P_2 so

that we have

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \begin{bmatrix} R & -R^T T \end{bmatrix} \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} \quad (3.5)$$

Plugging into equation (2.3), we rewrite the equation as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\frac{f}{s_x} & 0 & o_x \\ 0 & -\frac{f}{s_x} & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & -R^T T \end{bmatrix} \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} \quad (3.6)$$

or, it is simplified to

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = M_{\text{int}} M_{\text{ext}} \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix}$$

and we have

$$u = \frac{x}{z} \quad \text{and} \quad v = \frac{y}{z}$$

3.3 Lowe's method

The basic perspective projection equations illustrates the relation between the position of corresponding features in two images and the pose change of a three dimensional model. However, it still has not provided the solution of the pose change calculation. An early work was done by Lowe [33]. It demonstrates that the locations of all projected model features in an image have to be consistent with the projection from a single viewpoint.

Assumed that the pose change parameters of the model between two images are denoted as $(\phi_x, \phi_y, \phi_z, x, y, z)$. The partial derivatives of u and v with respect to each of the pose change parameters are able to be calculated. The Newton method is used to calculate the optimum correction rotations $\Delta\phi_x, \Delta\phi_y, \Delta\phi_z$ about the camera-centered axes and correction translation $\Delta D_x, \Delta D_y, \Delta D_z$. Since have:

$$u = \frac{fx}{z + D_z} + D_z$$

so

$$\frac{\delta u}{\delta D_z} = 1$$

Also

$$\frac{\delta u}{\delta \phi_y} = \frac{f}{z + D_z} \frac{\delta x}{\delta \phi_y} - \frac{fx}{(z + D_z)^2} \frac{\delta z}{\delta \phi_y}$$

but we know that

$$\frac{\delta x}{\delta \phi_y} = z \quad \text{and} \quad \frac{\delta z}{\delta \phi_y} = -x$$

for simplicity, a constant c is substituted to:

$$c = \frac{1}{z + D_z}$$

giving,

$$\frac{\delta u}{\delta \phi_y} = fcz + fc^2 x^2 = fc(z + cx^2)$$

Similarly,

$$\frac{\delta u}{\delta \phi_z} = \frac{f}{z + D_z} \frac{\delta x}{\delta \phi_z} = -fcy$$

The goal of the multi-dimensional Newton convergence is to solve the vector of corrections

$$h = [\Delta D_x, \Delta D_y, \Delta D_z, \Delta \phi_x, \Delta \phi_y, \Delta \phi_z]$$

The error of u , v components is used independently to create separate linear constraints.

The error E_u and E_v of u and v is generated by the sum of the projects of its partial derivatives times the unknown error-correction values:

$$E_u = \frac{\delta u}{\delta D_x} \Delta D_x + \frac{\delta u}{\delta D_y} \Delta D_y + \frac{\delta u}{\delta D_z} \Delta D_z + \frac{\delta u}{\delta \phi_x} \Delta \phi_x + \frac{\delta u}{\delta \phi_y} \Delta \phi_y + \frac{\delta u}{\delta \phi_z} \Delta \phi_z \quad 3.7$$

$$E_v = \frac{\delta v}{\delta D_x} \Delta D_x + \frac{\delta v}{\delta D_y} \Delta D_y + \frac{\delta v}{\delta D_z} \Delta D_z + \frac{\delta v}{\delta \phi_x} \Delta \phi_x + \frac{\delta v}{\delta \phi_y} \Delta \phi_y + \frac{\delta v}{\delta \phi_z} \Delta \phi_z \quad 3.8$$

where the partial derivatives of u and v with respect to each of the camera viewpoint parameters are:

	u	v
D_x	1	0
D_y	0	1
D_z	$-fc^2x$	$-fc^2y$
ϕ_x	fc^2xy	$fc(z+cy^2)$
ϕ_y	$fc(z+cx^2)$	fc^2xy
ϕ_z	$-fcy$	fcx

Table 3.1 Residual Error with respect to pose parameters

In general the more correspondences between model and image the better the result, one way to merge correspondences is to use the Gauss least-squares method. The matrix equation is rewritten to:

$$Jh = \hat{e}$$

where J is the Jacobian matrix containing the partial derivatives, h is the vector of unknown corrections for pose change parameters that we are solving, and \hat{e} is the vector of total 2D error measured in the image. When the iteration converges to a correct solution, a least squares fit of the error is able to be performed by solving the corresponding normal equation:

$$J^T J h = J^T \hat{e}$$

By solving the vector h , we are able to get the pose change of the object between the images.

3.4 Interleave bundle adjustment for structure and motion recovery from multiple images

The original Lowe's pose estimation is designed to solve the pose estimation problem with a known structure in a single image. It is also an advantage of this method.

However, this method assumes that the model structure \vec{P} is known. Generally, in real application, this assumption does not hold. Chang and Wong [61] implemented an

interleave bundle adjustment algorithm to solve it by using a sequence of Γ images to find the model \vec{P} and the pose sequence $\vec{\theta} = \{\theta_1, \theta_2, \dots, \theta_t, \dots, \theta_\Gamma\}$.

Assumed an object M with N observable feature points $\vec{P} = (P_1, P_2, \dots, P_i, \dots, P_N)$ where $P_i = [X_i, Y_i, Z_i]$ is a 3D position (Figure 3.2). At time $t = 1$, the object is placed at an arbitrary location. At time $t = 2, 3, \dots, \Gamma$, the object is moved to a new position of rotational R_t and translation T_t transformations. A set of 2D feature point $\vec{q} = \{q_{1,t}, q_{2,t}, \dots, q_{n,t}\}$ is taken at time t .

The projected 2D feature $q_{i,t}$ is expressed as the projection result of the rotation and translation of model point P_i . Assumed the rotation and the translation of the model at

time t is $R_t = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}_t$ and $T_t = ([T_1, T_2, T_3]^T)_t$, respectively. The 2D projection

result is $q_{i,t} = (u_{i,t}, v_{i,t}) = g(R_t P_i + T_t)$, where $g()$ is a projective function, $T_t = ([T_1, T_2, T_3]^T)_t$,

is the translation vector and $R_t = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}_t$ is the rotational matrix. Specifically, $g()$ is

$$\begin{aligned} u &= f \frac{r_{11}X + r_{12}Y + r_{13}Z + T_1}{r_{31}X + r_{32}Y + r_{33}Z + T_3} \\ v &= f \frac{r_{21}X + r_{22}Y + r_{23}Z + T_2}{r_{31}X + r_{32}Y + r_{33}Z + T_3} \end{aligned} \tag{3.9}$$

Let $\hat{\theta}$ be the initial estimate of the pose and θ be the true pose, so that $\theta = \hat{\theta} + \delta\theta$, where $\delta\theta$ is the estimation error. By using expanding $g(\theta, P)$ into a series, we have

$$q_i = g(\theta, P_i) = g(\hat{\theta}, P_i) + \frac{dg(\hat{\theta}, P_i)}{d\hat{\theta}} \delta\hat{\theta} \quad (3.10)$$

where q_i is the actual 2D feature point we measured, and $g(\hat{\theta}, P_i)$ is the predicted feature point based on the known model point P_i and the estimated pose $\hat{\theta}$. The residual error between the measured feature point position and the predicted position is

$$\delta q_i = \frac{dg(\hat{\theta}, P_i)}{d\hat{\theta}} \delta\theta$$

Because both δq_i and $g(\hat{\theta}, P_i)$ are measurable or calculable, $\delta\theta$ is able to be found. By decreasing the residual error δq_i , the estimated pose is able to gradually approach to the true pose by repeating the process iteratively until the residual error is small enough.

Let the initial estimated pose $\hat{\theta} = [T_1, T_2, T_3, \phi_1, \phi_2, \phi_3]$, where ϕ_1, ϕ_2, ϕ_3 denotes the roll, pitch and yaw (RPY) angles. By using Eq. 3.7 and Eq. 3.8, we have

$$\begin{aligned} \delta u_i &= \sum_{j=1}^3 \left[\frac{\partial u_i}{\partial T_j} \Delta T_j + \frac{\partial u_i}{\partial \phi_j} \Delta \phi_j \right], \\ \delta v_i &= \sum_{j=1}^3 \left[\frac{\partial v_i}{\partial T_j} \Delta T_j + \frac{\partial v_i}{\partial \phi_j} \Delta \phi_j \right]. \end{aligned}$$

To compute the residual error, the i -th point of the 3D model is rotated and translated.

By using Eq. 3.9, the rotated model (X_i^R, Y_i^R, Z_i^R) is given by

$$\begin{aligned} X_i^R &= r_{11}X_i + r_{12}Y_i + r_{13}Z_i \\ Y_i^R &= r_{21}X_i + r_{22}Y_i + r_{23}Z_i \\ Z_i^R &= r_{31}X_i + r_{32}Y_i + r_{33}Z_i \end{aligned}$$

and the translated model (X'_i, Y'_i, Z'_i) is

$$\begin{aligned} X'_i &= X_i^R + T_1 \\ Y'_i &= Y_i^R + T_2 \\ Z'_i &= Z_i^R + T_3 \end{aligned}$$

By using the result of table 3.1, we have the residual error component with respect to T:

$$\frac{\partial u_i}{\partial T_1} = \frac{f}{Z_i}, \quad \frac{\partial u_i}{\partial T_2} = 0 \quad \text{and} \quad \frac{\partial u_i}{\partial T_3} = -f \frac{X_i}{Z_i^2},$$

and

$$\frac{\partial v_i}{\partial T_1} = 0, \quad \frac{\partial v_i}{\partial T_2} = \frac{f}{Z_i} \quad \text{and} \quad \frac{\partial v_i}{\partial T_3} = -f \frac{Y_i}{Z_i^2},$$

The residual error components with respect to ϕ are written as:

$$\frac{\partial u_i}{\partial \phi_1} = -f \frac{X_i Y_i^R}{Z_i^2}, \quad \frac{\partial u_i}{\partial \phi_2} = f \frac{X_i X_i^R + Z_i Z_i^R}{Z_i^2}, \quad \frac{\partial u_i}{\partial \phi_3} = -f \frac{Y_i^R}{Z_i}$$

and

$$\frac{\partial v_i}{\partial \phi_1} = -f \frac{Y_i Y_i^R + Z_i Z_i^R}{Z_i^2}, \quad \frac{\partial v_i}{\partial \phi_2} = -f \frac{X_i Y_i^R}{Z_i^2}, \quad \frac{\partial v_i}{\partial \phi_3} = -f \frac{Y_i^R}{Z_i}$$

Now we have six unknown parameters $(\Delta T_1, \Delta T_2, \Delta T_3, \Delta \phi_1, \Delta \phi_2, \Delta \phi_3)$ in two equations for each feature point. When we have N feature points in a model, we have $2N$ equations for the six unknown parameters. The equations are then solved by using standard least

square methods [50]. After ΔT_i and $\Delta \phi_i$ have been solved, they are converted back to matrix representation for further calculation.

Now the model estimation component is added into the calculation process. By extending the equation Eq. 3.10, the estimated model error δP_i is included in the calculation of residual error. The Eq. 3.10 is then expanded to

$$q_{i,t} = g(\hat{\theta}_i, \hat{P}_i) + \frac{dg(\hat{\theta}_i, \hat{P}_i)}{d\hat{\theta}_i} \delta\theta_i + \frac{dg(\hat{\theta}_i, \hat{P}_i)}{d\hat{P}_i} \delta P_i$$

where \hat{P} represents the current state estimation of the model points. Then the model is improved by updating $P_i = \hat{P}_i + \delta P_i$. The model and pose change development are separated into two phases. The first phase estimated the pose sequence

$\vec{\hat{\theta}} = \{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_t, \dots, \hat{\theta}_\Gamma\}$ by an estimated model \hat{P} . The second phase then used the

estimated $\vec{\hat{\theta}}$ result in phase one to update the estimated model \hat{P} . The two phases iterate until the overall residual error becomes a minimum.

The general description of Wong and Chang's algorithm is shown as follow:

Step 1: Feature sets $\{q_{i,t}\}$ are extracted from images for $i = 1, 2, \dots, N$ and $t = 1, 2, \dots, \Gamma$.

Step 2: Initialize model structure \hat{P} , e.g. a plane with some noise

Step 3: First pass:

For $t = 1$ to Γ

Using \hat{P} and image feature $\{q_{1,t}, q_{2,t}, \dots, q_{N,t}\}$ of the t -th image to find $\hat{\theta}_t$ by Lowe's method.

End

An estimated pose sequence $\vec{\hat{\theta}} = \{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_\Gamma\}$ is found.

Step 4: Second pass:

For $i = 1$ to N

Using $\vec{\hat{\theta}}$ and the i -th image features of the all images $\{q_{i,1}, q_{i,2}, \dots, q_{i,\Gamma}\}$ find a better-predicted model structure \hat{P} using Newton's method by minimizing the residual image error.

End

An estimated structure pose \hat{P} is found.

Step 5: Stop if the 2D total re-projection error $\sum_{all\ i,t} (\delta q_{i,t})^2$ is small enough or too many

iterations have been executed. Otherwise loop back to step 3.

3.5 Feature points mismatch analysis

One of the significant causes of incorrect model recovery is the error of feature points.

Those errors of feature points mainly come from three sources:

1. The extraction error from the feature extraction program
2. The noise from the original picture
3. The missing points from the motion of the object

The effect of noise is explained. Assumed the noise of a feature point i at frame t to be

$\xi u_{i,t}$ and $\xi v_{i,t}$, then the feature point location becomes:

$$\delta u'_{i,t} = \delta u + \xi u_{i,t},$$

$$\delta v'_{i,t} = \delta v + \xi v_{i,t},$$

The noise error term causes a deformation of the model. Because it is bounded by an

initial base image, the deviation of δX_{e_i} and δY_{e_i} will not be greater than

$$\delta X e_i = \frac{\delta Z e_i * \xi u_{i,t}}{f}$$

$$\delta Y e_i = \frac{\delta Z' e_i * \xi v_{i,t}}{f}$$

where f is the focal length of the camera

Since we have,

$$\delta u_{i,t} = a'_{11} \delta X_i + a'_{12} \delta Y_i + a'_{13} \delta Z_i$$

$$\delta v_{i,t} = a'_{21} \delta X_i + a'_{22} \delta Y_i + a'_{23} \delta Z_i$$

Therefore, the effect of $\xi u_{i,t}$ and $\xi v_{i,t}$ are rewritten to

$$\xi u_{i,t} = a'_{11} \delta X e_i + a'_{12} \delta Y e_i + a'_{13} \delta Z e_i$$

$$\xi v_{i,t} = a'_{21} \delta X e_i + a'_{22} \delta Y e_i + a'_{23} \delta Z' e_i$$

where $Z e_i$ is the effect of Z change due to ξu_i and $Z' e_i$ is the effect of Z change due to ξv_i .

By substituting the value of $\delta X e_i$ and $\delta Y e_i$, we have

$$\xi u_{i,t} = a'_{11} \frac{\delta Z e_i * \xi u_{i,t}}{f} + a'_{12} \frac{\delta Z' e_i * \xi v_{i,t}}{f} + a'_{13} \delta Z e_i$$

$$\xi v_{i,t} = a'_{21} \frac{\delta Z e_i * \xi u_{i,t}}{f} + a'_{22} \frac{\delta Z' e_i * \xi v_{i,t}}{f} + a'_{23} \delta Z' e_i$$

Then we have the total noise effect of Z_e from both the horizontal and vertical pixel noise:

$$\delta Z_{e_i} + \delta Z'_{e_i} = \frac{f(a'_{21}\xi u_{i,t}^2 + a'_{12}\xi v_{i,t}^2 - a'_{11}\xi u_{i,t}\xi v_{i,t} - a'_{13}\xi v_{i,t}f - a'_{22}\xi u_{i,t}\xi v_{i,t} - a'_{23}\xi u_{i,t}f)}{(a'_{11}\xi u_{i,t} + a'_{13}f)(a'_{22}\xi v_{i,t} + a'_{23}f) - a'_{12}a'_{21}\xi u_{i,t}\xi v_{i,t}}$$

From the above result, we found that point mismatch causes a serious model distortion during reconstruction. In spite of many improvements in feature points tracking algorithms, point correspondence are still a problem in this topic especially for real images. Therefore, we need a mechanism to pick out the mismatch points. Chang and Wong [61] suggest two methods to solve the problem.

The first method is to hard code the boundary of the feature point by using a rectangular boundary. An important source of mismatched feature points comes from the background of the picture. This method filters out the unwanted background feature points. The advantage of this method is fast and simple. However, the weakness of this method is not robust enough and it is not able to filter out some background feature points if the object under test is an irregular object. Besides, this method is also not able to fix the mismatch feature points belonging to the object.

The second method is to delete those fast moving feature points. If the movement of a feature point is found abnormally large between two pictures, it will be considered as a noisy feature point and it will be deleted during the model recovery. This method is able

to filter some noisy features. However, if the image of object under test is moving fast, this method will delete some actual feature points by wrongly considering those feature points as noise.

The feature point mismatch effect is demonstrated by a simulation result using a sphere as follows. The simulation is processed by the following condition.

Step 1: The set of 3D sphere raw data is generated by Matlab software

Step 2: The 3D sphere is then rotated by a small step sequentially

Step 3: An image is captured for each pose-changed sphere

Step 4: Those images are feed to the alternative bundle algorithm and a 3D model is generated

Step 5: Some noise is added to those captured images

Step 6: Those images are feed to alternative bundle adjustment algorithm and a 3D model is generated again

The model output from step 4 for the images set without noise is shown in figure 3.4a.

The model output from step 6 for the image set with noise is shown in figure 3.4b. From the model result, serious distortion is found due to the effect of the mismatch noise at the input image sequence. More detail about the synthetic data will be discussed in chapter 5 again.

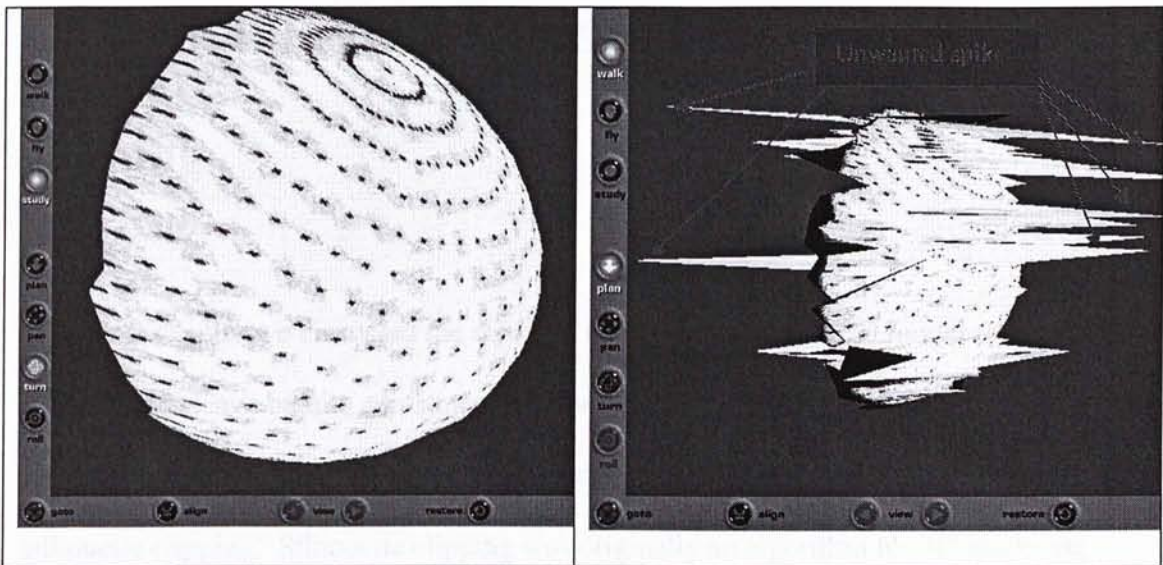


Figure 3.4a (Left) The simulation sphere result from the images without noise. Figure 3.4b(Right) the simulation sphere result from the image with noise

In this chapter, we have a revision on the perspective camera model and discussed the details of an interleave bundle adjustment algorithm. We have also studied the effect of feature mismatch to that algorithm. In the next chapter, we will suggest a method to improve the noise effect problem.

Chapter 4: Feature selection by using silhouette clipping

In the last chapter we discussed the effect of feature mismatch in 3D model recovery by using an interleave bundle adjustment method. In this chapter, we shall propose a solution to improve the problem. Our proposal is inspired by a technique called silhouette clipping. Silhouette clipping was originally an algorithm for 3D rendering which was introduced by Sander, Hoppe and Snyder[45]. In the rest of this chapter, we have a brief introduction about the silhouette clipping technique and how to enhance the result of the interleave bundle adjustment algorithm.

4.1 Introduction to silhouette clipping

In a rendering process, a set of triangles to describe the surface models is required. However, to process those triangles is also the bottleneck in geometry processing. The output quality and the calculation efficiency in a rendering process are contradicted to each other because the increase in the number of triangles provides a better result visually but causes longer calculation time. Much work has been done for how to generate a surface model in using coarse meshes [35] [11] [10]. Those techniques give a pretty

good approximate result by using fewer triangles; however, it is not a true solution to the problem.

Sander, Hoppe and Snyder[45] introduced a silhouette chipping technique to improve the performance. The basic idea of their algorithm is shown as follows:

Preprocess - Give a dense original mesh:

- Build a progressive hull representation of the original mesh and extract from it a coarse mesh, which has the property that it encloses the original, allowing proper clipping.
- Sample the color and/or normal field of the original mesh, construct a texture map and/or normal map over each face of the coarse mesh.
- Enter the edges of the original mesh into a search tree for efficient runtime extraction of silhouette edges.

Runtime – for a given viewpoint:

- Extract the silhouette edges from the search tree
- Create a mask in the stencil buffer by drawing the silhouette edges as triangle fans.
- Render the coarse mesh with its associated texture/normal maps, but clipped an anti-aliased using the stencil.

4.2 Silhouette clipping for 3D model

Out of bound feature

Silhouette clipping extracts the silhouette edges to set a stencil and clips away the out of bound data. It is able to filter out those feature data that do not belong to the observed object. Figure 4.1 shows an example of out of bound feature with an imaginary sphere face. During the feature capture phase, some out of bound features are captured for model recovery. Those bad feature points distort output model and are needed to be removed. By checking whether a feature point lies outside the silhouette of the object, it is not difficult to separate out those bad feature points from good ones.

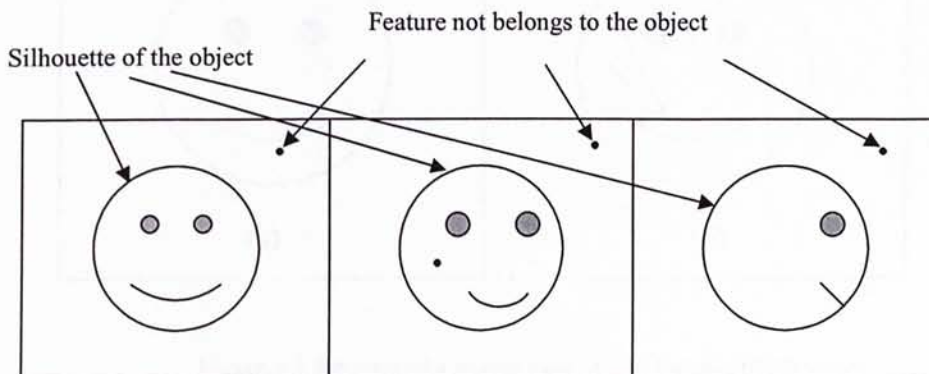


Figure 4.1 Raw pictures of sphere face

Out of bound spike

As we mentioned in the last chapter, the feature noise or mismatch causes the distortion of the model and generates many spikes on the output model. When those mismatch feature points are removed, those spikes will be disappeared as well. Let's use the sphere face example again. When a mismatch feature point exists on the sphere face picture, the 3D output model will generate an unwanted spike similar to that in figure 4.2. Figure 4.2a shows a perfect sphere face. Figure 4.2b shows a sphere face with a spike from the same point of view.

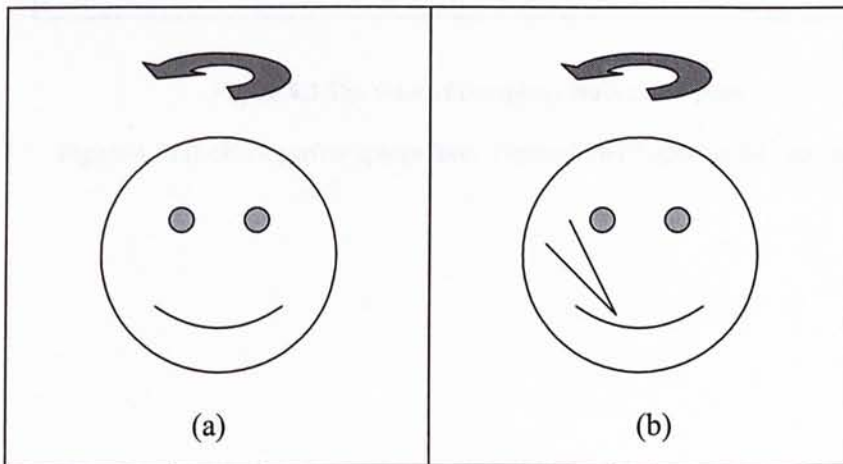


Figure 4.2 the view of a sphere face from reference frame view.

Figure 4.2a (Left), a perfect sphere face. Figure 4.2b (Right) the face has a spike

It is difficult to pick out the mismatch point from the view of figure 4.2b. However, when the camera view is changed to another position, for instance, rotates the object by 90 degrees, those spikes will be easier to be found out, at least by human eyes. For

examples, in figure 4.3b, the spike is easier to be figured out when the camera view is changed. Figure 4.3a shows the perfect model view for the same pose.

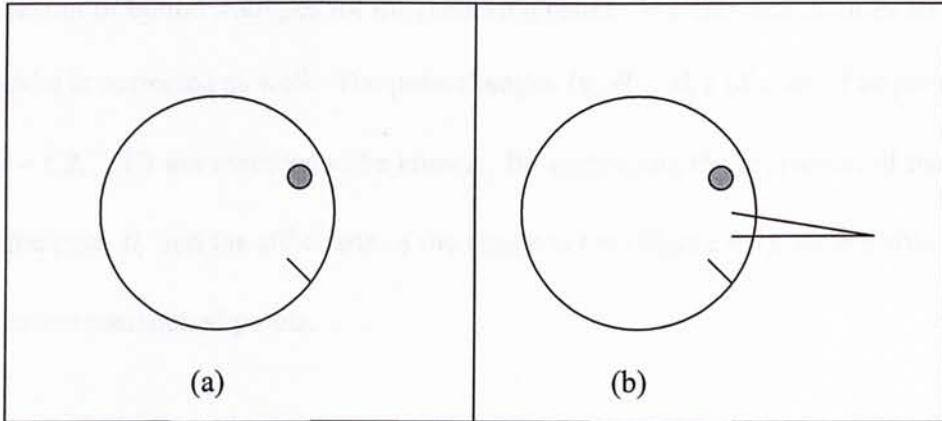


Figure 4.3 The view of the sphere from a new pose.

Figure 4.3a (Left), a perfect sphere face. Figure 4.3b (Right) the face has a spike

The spike in figure 4.3b is easier to be detected by human eyes because all spike points shall not exceed the silhouette boundary in all pose projection view. It is quite similar to the idea of silhouette clipping. However, the original silhouette clipping method only corrects the out of bound triangles for the rendering result. We enhance the idea so that the 3D model is corrected as well. The pose changes $(\theta_1, \theta_2, \dots, \theta_\Gamma)$ of a set of sequential picture $(t = 1, 2, \dots, \Gamma)$ are assumed to be known. By comparing the projection of the model at the pose θ_i and the silhouette of the image at $t=i$ (Figure 4.4), we are able to pick out those mismatched points.

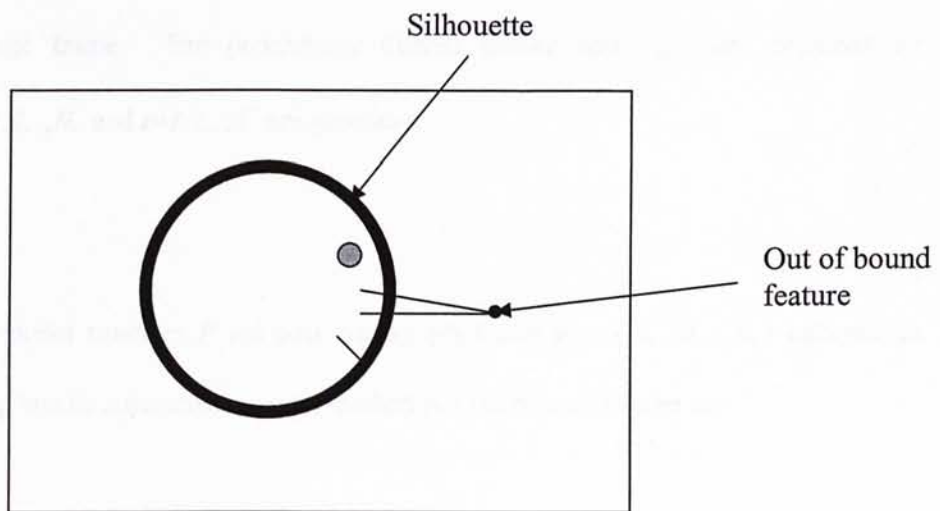


Figure 4.4 Use the silhouette to filter the out of bound feature

The procedure of the boundary filter is shown as follows:

Step 1: Raw feature sets $\{q_{i,t}\}$ are extracted from images for $i=1,2,\dots,N$, and $t=1,2,\dots,\Gamma$.

Step 2: Boundary sets $\{b_{j,t}\}$ are extracted from image for $j=1,2,\dots,M$, and $t=1,2,\dots,\Gamma$ by using the snake algorithm.

Step 3: First pass:

For $t=1$ to Γ

By using the boundary outline $\{b_{j,t}\}$, take out all the out of bound feature in each image frame. The preliminary filtered feature sets $\{q_{i,t}\}$ are extracted for $i=1,2,\dots,N$, and $t=1,2,\dots,\Gamma$ are generated

End

Step 4: Initialize model structure \hat{P} and pose change per frame $\hat{\theta} = \{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_\Gamma\}$ information per frame by using bundle adjustment Lowe's method and the filtered feature sets.

Step 5: Second pass:

For $t=1$ to Γ

For $\hat{\theta} = \hat{\theta}_1$ to $\hat{\theta}_{\frac{\Gamma}{4}}$

1. Transfer the model \hat{P} to pose $\hat{\theta}$
2. Project the transferred model to corresponding image view.

3. By using the boundary outline $\{b_{j,t}\}$, take out all the out of bound features in each image frame. The filtered feature sets $\{q_{i,t}\}$ are extracted for $i=1,2,\dots,N'$, and $t=1,2,\dots,\Gamma$ are generated.

End

End

Step 6: Iterate the model structure \hat{P} and pose change per frame $\vec{\hat{\theta}} = \{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_\Gamma\}$ by using bundle adjustment Lowe's method and the filtered feature sets.

Step 7: Loop back to step 5 until either deletion of no further point or exit the maximum iteration number.

Figure 4.5 shows the flow chart of our algorithm. Step1 prepares the raw features for model recovery and step 2 prepares the silhouette boundary for boundary filter. Step 3 filters the useless background features which do not belong to the object. Step 4 generates the preliminary model and the pose change per frame. However, the model in this stage is still distorted by noise and mismatch points.

Step 5 mainly has two purposes. The first purpose is to move the model to the poses of the future frames and to generate the projected view of the model at those poses. The second purpose is to delete the model points if those corresponding feature points are out of bound in the projected view.

Step 6 regenerates the model again. Because the model and the pose that were generated in step 5 are distorted, we need to regenerate the model and pose after the feature points are updated. The new result will be more correct.

Step 7 reiterate the boundary filter and model generation until the stop criteria is fulfill.

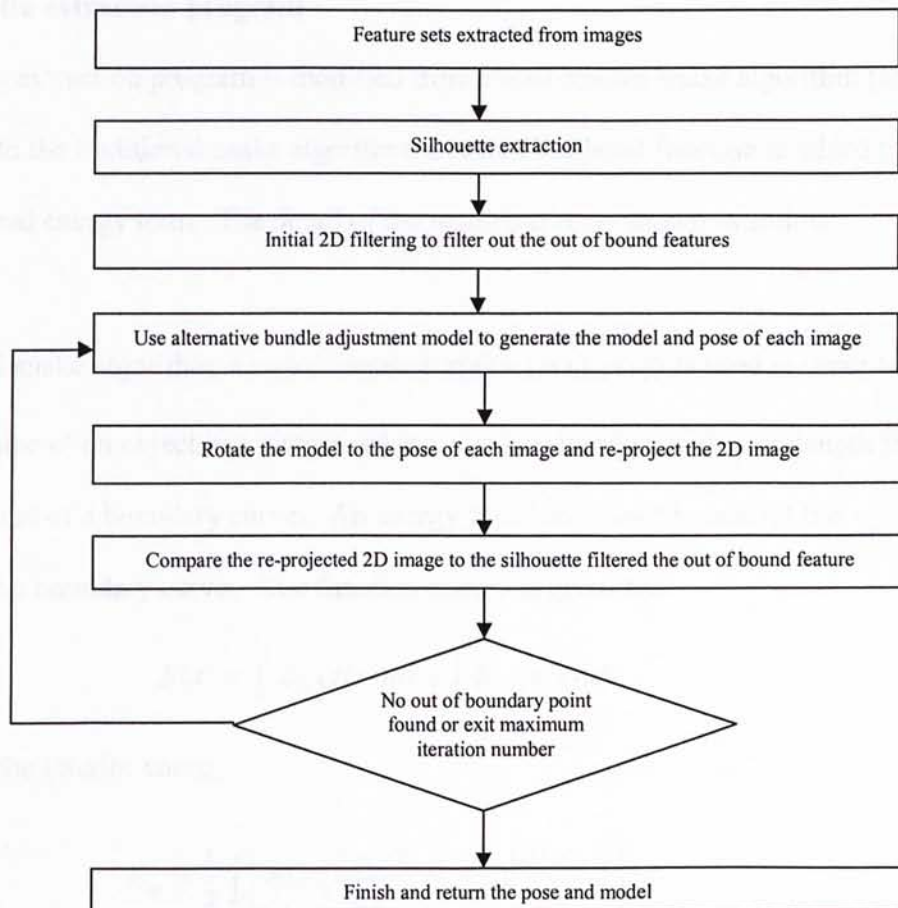


Figure 4.5 – Flowchart of feature selection by silhouette clipping

4.3 Implementation

The implementation of our work is divided to two parts. The first part is a silhouette extraction program. The second part is a feature filter which is merged into the algorithm of interleave bundle adjustment algorithm.

4.3.1 Silhouette extraction program

Our silhouette extraction program is modified from a well-known snake algorithm [69]. In addition to the traditional snake algorithm, a color likelihood function is added to the original external energy term. The detail of the modification is shown as follow:

In the original snake algorithm, a curve function $z(s) = (x(s), y(s))$ is used to describe the boundary outline of an object in a picture, where s is measured from the arc-length from the starting point of a boundary curve. An energy function is used to control the evolution of the boundary curve. The function energy is given by:

$$E(z) = \int_0^l E_{\text{int}}(z(s))ds + \int_0^l E_{\text{ext}}(z(s))ds$$

where E_{int} is the interior energy

$$E_{\text{int}} = \frac{1}{2} \int_0^l \left(\alpha(s) \left| \frac{\delta z(s)}{\delta s} \right|^2 + \beta(s) \left| \frac{\delta^2 z(s)}{\delta s^2} \right| \right) ds$$

and $\alpha(s)$ is the snake tension and $\beta(s)$ is the snake rigidity.

The external energy E_{ext} typically uses the spatial gradient $-\|\nabla I\|$ of the gray-level intensity image I , computes at each snake point. To minimize the energy function, the Euler equation $F_{int} + F_{ext}$ has to be equal to zero. F_{int} is the interior force from internal energy E_{int} and F_{ext} is the external force from external energy E_{ext} .

When the snake program is used to capture the silhouette outline in real images in our application, the outline is wrongly captured when the background is noisy. Therefore, a minor modification is made to add a color analysis on top of the snake algorithm.

A color histogram of the target object is a pre-requisite of our color analysis. The target object is assumed to be located at the center of the image. The sample color histogram is obtained from a 10 by 10 square box around the center point of the picture. The color histogram is then normalized and translated to probability function $p(t | h)$.

$p(t | h)$ represents the probability that a color measurement t belongs to the target object based on the color histogram h of the target object.

An initial boundary curve is set in an arbitrary area on the image that covers the target object. In addition to the original external force in the snake algorithm, a color likelihood external force $-\nabla E_{col}$ is added so that the external force is then changed to

$F_{ext} = -\nabla E_{ext} - \nabla E_{col}$. This color external force is proportional to the color likelihood of the control point to the target object. It is then formulated to:

$$\nabla E_{col} = \frac{1}{p(t|h)} \text{ if } p(t|h) \leq p_{threshold} \quad \text{and,}$$
$$\nabla E_{col} = 0 \text{ if } p(t|h) > p_{threshold}$$

With this color external force term, the snake curve is able to overcome the local minimum attraction from the noisy and complex background and is able to catch the target object.

The program was implemented in Matlab environment. The original images are shown in Figure 4.6. Figure 4.6a, 4.6b, 4.6c and 4.6d shows the objects that need to be tracked for their outline respectively. The objects under test were put around the center area of the images.

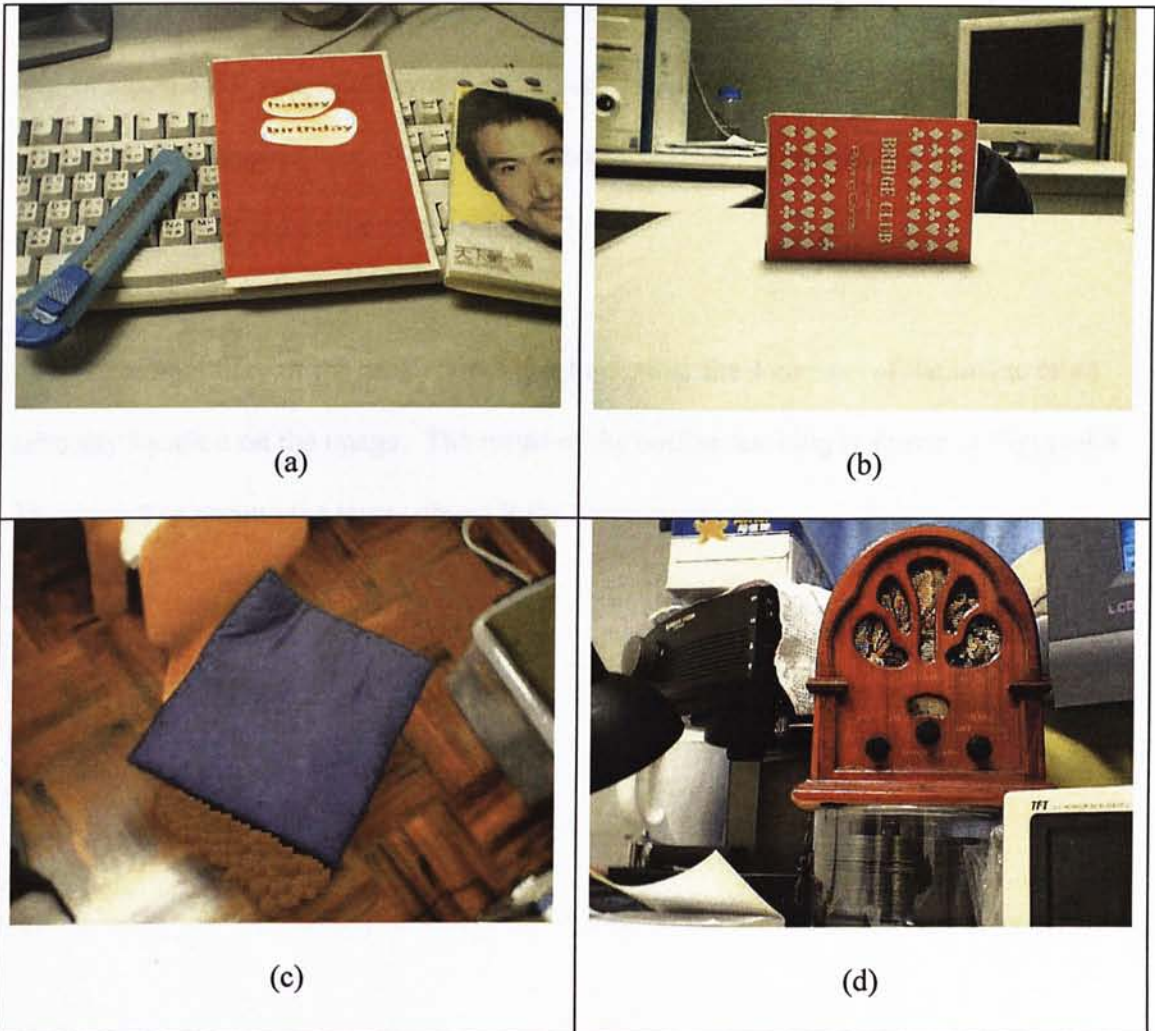


Figure 4.6 Examples of the input picture for the snake algorithm

Then the color histograms of the target object were captured. Figure 4.7a, 4.7b, 4.7c and 4.7d show the color histograms of the target object respectively. In Figure 4.7, the hue and saturation color space are divided to 51 partitions and was laid horizontally on the x-axis and y-axis respectively. The vertical axis represents the normalized frequency of a particular color found on the target object.

The initial boundary of the snake curve is set on either the 4 corners of the image or an arbitrary location on the image. The result of the outline tracking is shown at Figure 4.8. The thick line around the target object is the tracking result.



FIGURE 4.8. The result of snake curve tracking.

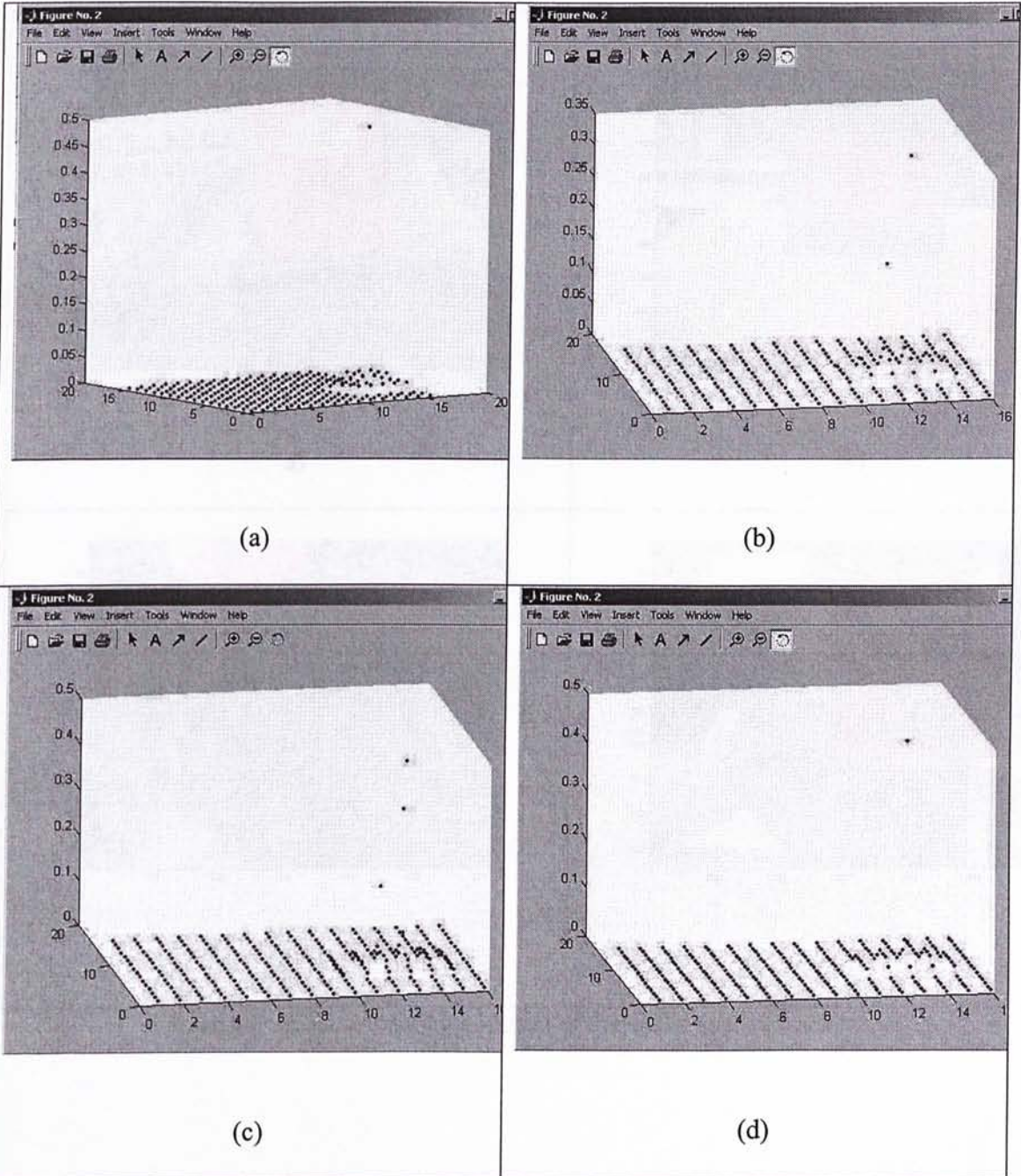


Figure 4.7. The color histogram of the target object in Figure 4.6

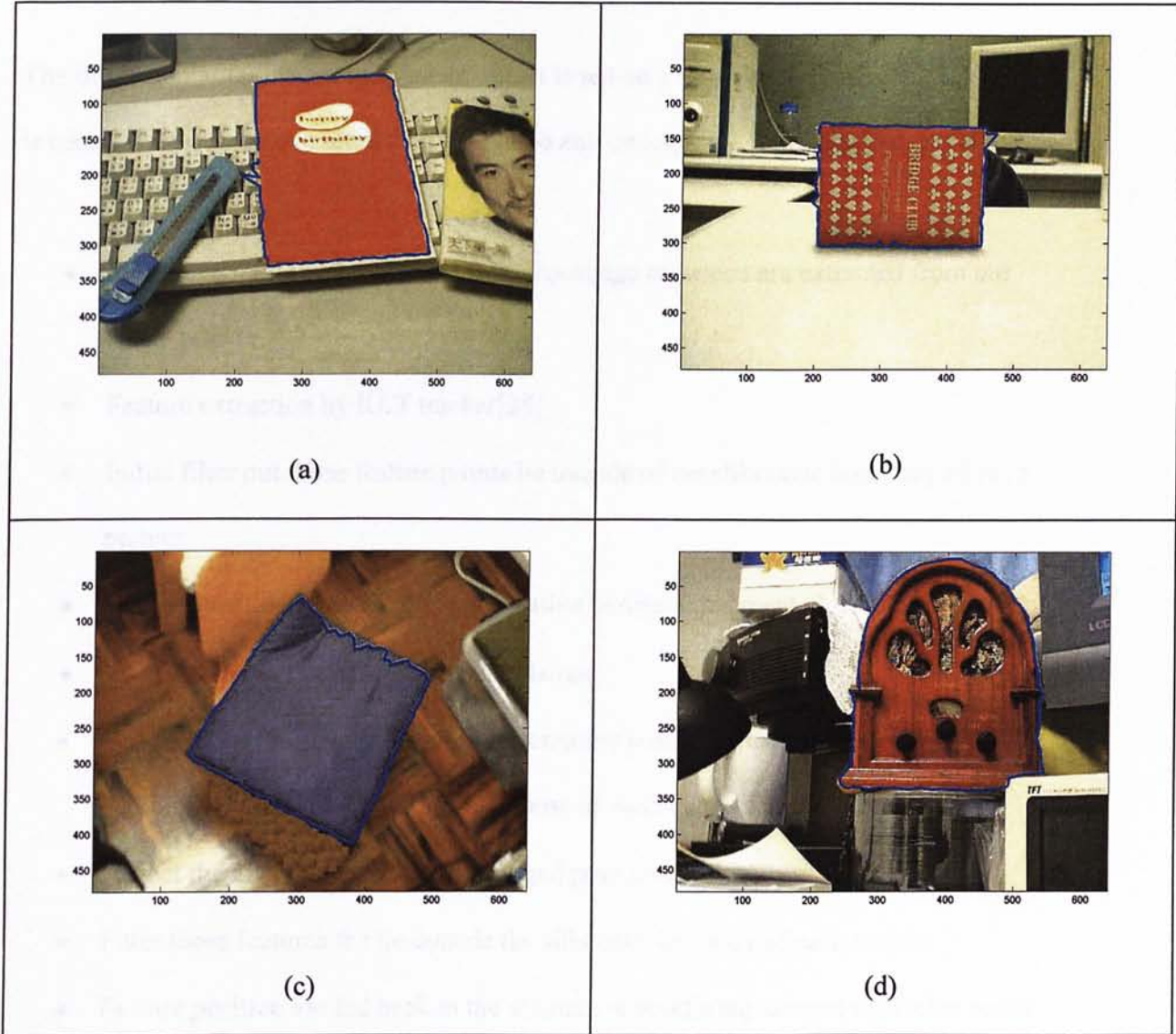


Figure 4.8 The result (in blue traces) of the snake outline tracker.

4.3.2 Feature filter for alternative bundle adjustment algorithm

The implementation is assumed that an object is put on a turntable and a video sequence is taken. Pictures are extracted from the video and undergo the following process steps:

- Silhouette outline of the object from the image sequence are extracted from our snake tracker.
- Feature extraction by KLT tracker[28]
- Initial filter out those feature points lie outside of the silhouette boundary of each picture
- Feature positions are fed to the alternative bundle adjustment algorithm
- 3D structure and pose information obtained
- Use the pose information from the alternative bundle adjustment algorithm, rotate and translate the 3D model from the pose of each image accordingly.
- Project the 2D pictures for each changed pose model
- Filter those features the lie outside the silhouette bounady of each picture
- Feature position are fed back to the alternative bundle adjustment algorithm again until the feature is good enough
- From the 3D structure and the first picture of the sequence, produce textured VRML files for display.

In this chapter, we described the idea of using silhouette clipping in feature points filtering. We also discussed about the details of the implementation work of our algorithm. The test result of our algorithm will be demonstrated in the next chapter.

Chapter 5 - Experimental data

5.1 Simulation

A synthetic data simulation has been used for testing the algorithm. A virtual sphere was generated by a Matlab program. The sphere had many feature points on its surface. The sphere was then placed before a simulation camera. The sphere was rotated and pictures were taken for each few degrees of rotation. Those pictures were then input to our algorithm for testing.

5.1.1 Input of simulation

The details of the simulation are described as follows:

A synthetic sphere was generated and about 1200 feature points in evenly distribution by scattered on the surface of the sphere. The synthetic sphere was then rotated before a virtual camera and a picture was taken for every 3 degrees of rotation. Totally 120 frames were taken. Figure 5.1 shows a few sample pictures of the original synthetic sphere. Because the rotation angle between each picture is not large, each sample picture looks quite similar and not much different.

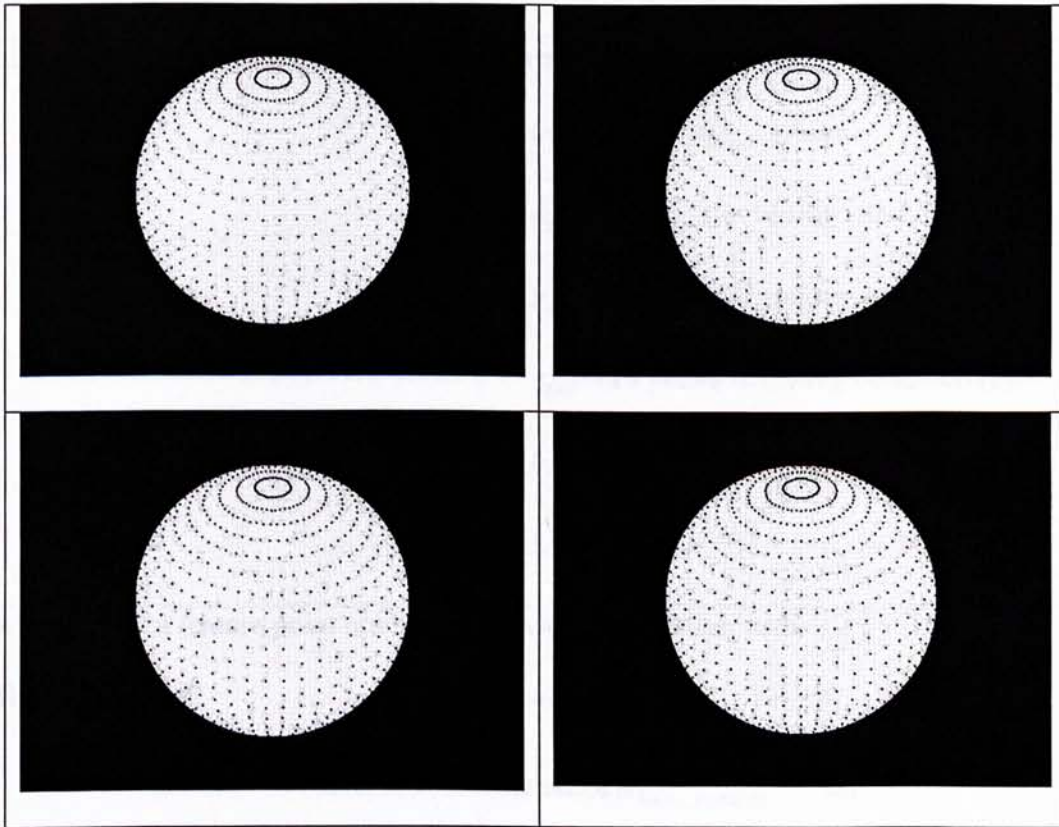


Figure 5.1 Sample pictures of original sphere

After the synthetic sphere pictures were generated, noise was injected into those pictures. To demonstrate the effect of noise, noise was injected in different magnitude in several tests so that the effect is easier to be compared. Noise was injected with the following conditions:

Noise magnitude distribution	<p>normal distribution with:</p> <p>mean (μ_{noise}) = n_{max}, where</p> <p>$n_{max} = 5$ pixels in testing condition (a).</p> <p>$n_{max} = 10$ pixels in testing condition (b).</p> <p>$n_{max} = 15$ pixels in testing condition (c).</p> <p>$n_{max} = 20$ pixels in testing condition (d).</p> <p>and, variance (σ_{noise}^2) = 5</p>
Probability of a feature point are inflected by noise	<p>normal distribution with:</p> <p>mean ($\mu_{Noise_probability}$) = 0.3,</p> <p>variance($\sigma_{Noise_probability}^2$) = 0.1</p>

Figure 5.2 and figure 5.3 show the captured feature points. Figure 5.2 shows the original feature points without noise. Figure 5.3a, 5.3b, 5.3c and 5.3d show the feature points with maximum noise magnitude (n_{max}) equals to the testing condition (a), (b), (c) and (d), or, 5, 10, 15 and 20 pixels respectively.

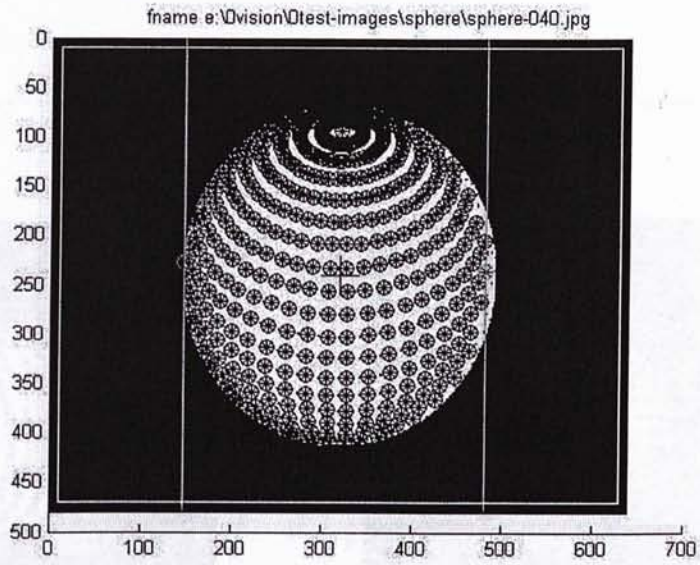


Figure 5.2 - Feature extraction for no noise condition

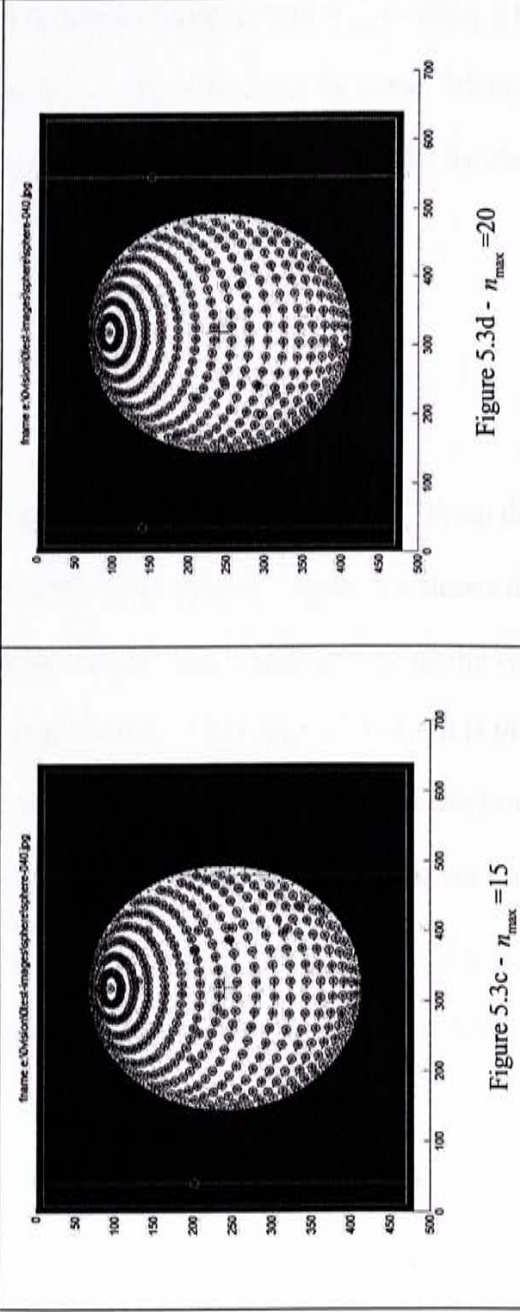
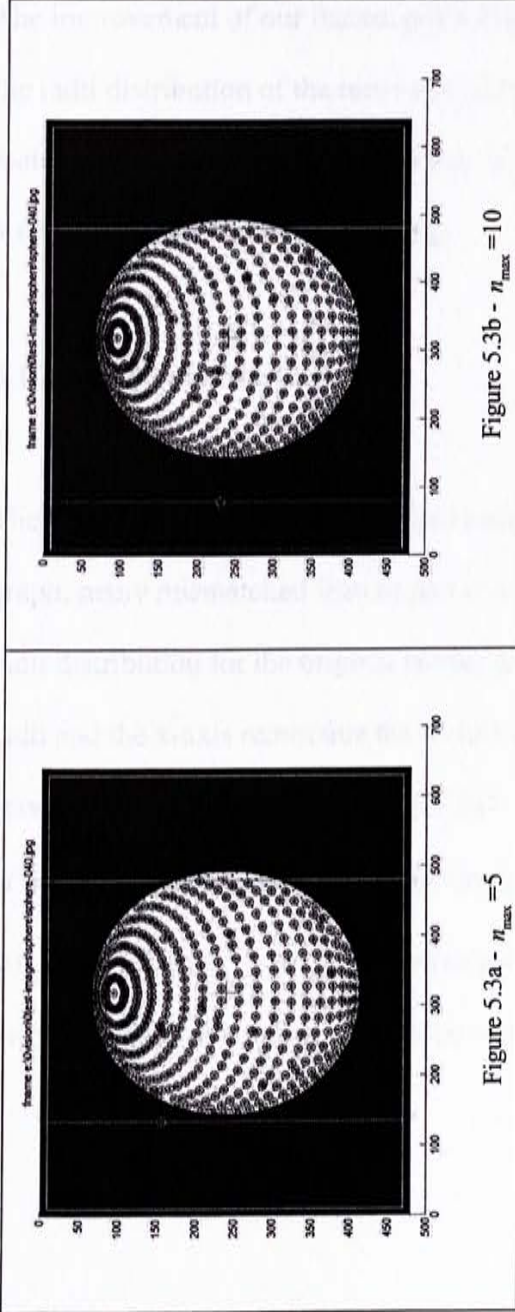
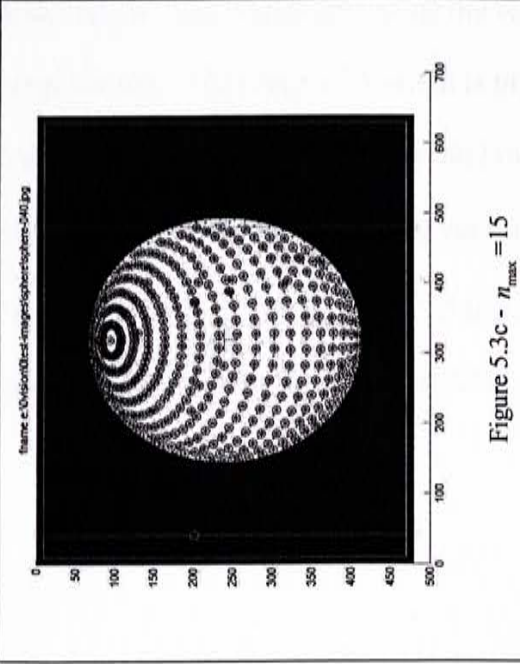
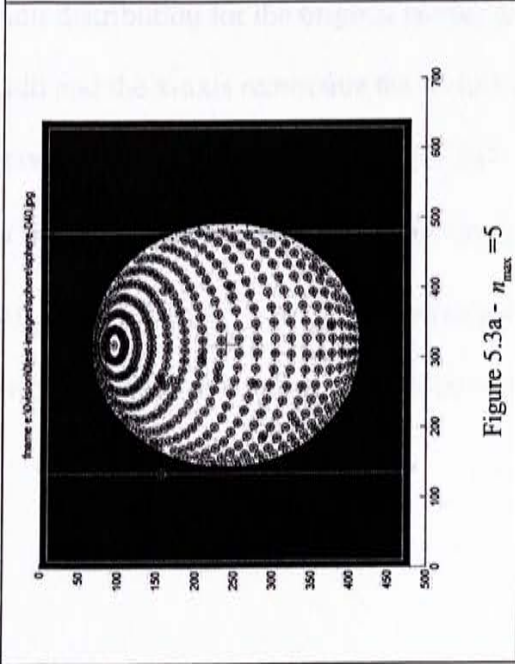


Figure 5.3 - Feature point with noise



5.1.2 Output of the simulation

The improvement of our feature point filter is demonstrated in three ways - comparing the radii distribution of the recovered 3D model points, comparing the actual 3D model feature points plotting. The third way is by plotting the recovered 3D model by using a VRML display with the texture map.

5.1.2.1 Radii distribution

The radius of each recovered sphere model feature point is plot in a graph. From the graph, many mismatched feature points are found to be filtered. Figure 5.4 shows the radii distribution for the original sphere without noise. The y-axis represents the scaled radii and the x-axis represents the feature point number. The range of the radii is in between 12×10^3 to 15×10^3 . Figure 5.6a1, 5.6b1, 5.6c1 and 5.6d1 show the scaled radius of the recovered model from the picture with noise magnitude n_{\max} in testing condition (a), (b), (c) and (d) respectively before using the feature filter. Figure 5.6a2, 5.6b2, 5.6c2 and 5.6d2 show the scaled radii of the recovered model after using the feature filter.

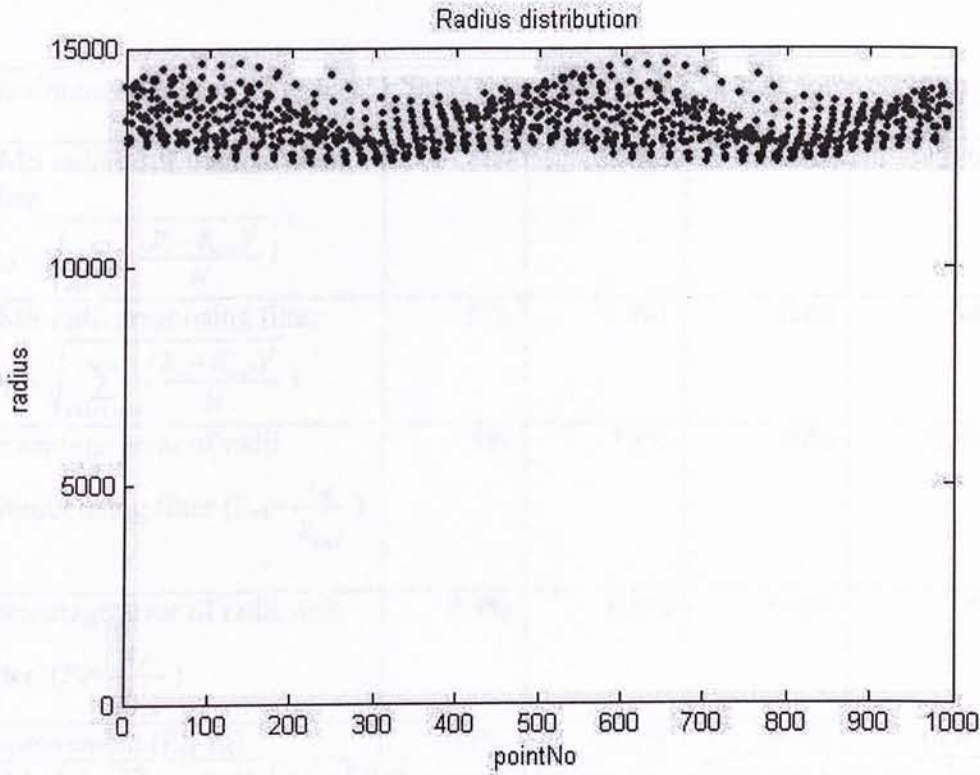


Figure 5.4 - The radius of the recovered 3D sphere feature distribution

When we compare the difference between the recovered model error before and after using the feature filter in figure 5.5, we find that the radii distribution with the filter is narrower than that without using the filter. Then the root means square (RMS) of the radii errors are calculated to demonstrate the improvement. Before the filter is used, the

Maximum Noise (n_{max}) / pixel	Small noise condition		Large noise condition	
	5	10	15	20
RMS radii error without using filter $(e_{nf} = \sqrt{\sum_{i=1}^N \frac{(R_i - R_{real})^2}{N}})$	669	993	2751	2894
RMS radii error using filter $(e_f = \sqrt{\sum_{i=1}^N \frac{(R_i - R_{real})^2}{N}})$	596	921	1268	947
Percentage error of radii without using filter ($E_{nf} = \frac{e_{nf}}{R_{real}}$)	5%	7.4%	20%	21.6%
Percentage error of radii with filter ($E_f = \frac{e_f}{R_{real}}$)	4.5%	6.9%	9.4%	7%
Improvement ($E_{nf} - E_f$)	0.6%	0.5%	10.6%	14.6%

Table 5.1 - The comparison of radius error between using filter and without using filter

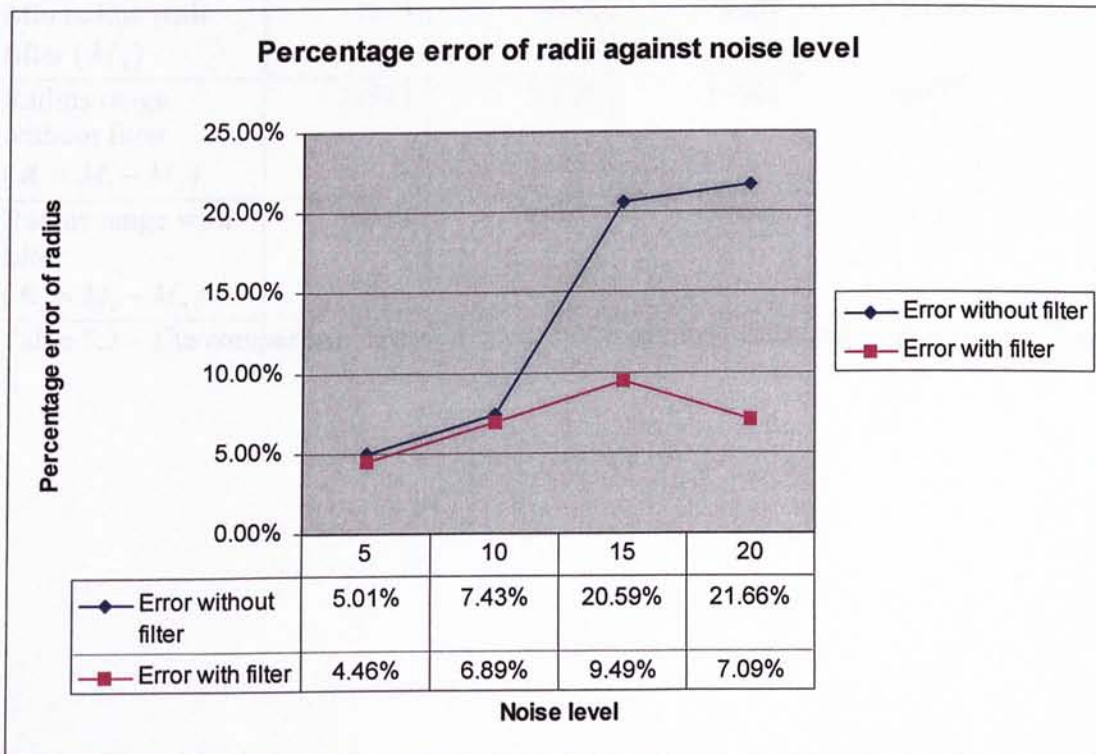


Figure 5.5 – The comparison of RMS radii error before and after using filter

The next analysis is done by comparing the radii range. In Figure 5.6a1, the radii range of the sphere model lies between 7×10^3 and 20×10^3 . After the feature filter is applied, the radii range is narrowed down from 7×10^3 to 15×10^3 . The radii range deviation of the model has about 51% improvement. In other testing conditions (figure 5.6b, 5.6c and 5.6d), similar improvement is also able to be found. The detail radii range improvement is shown in table 5.2.

Maximum Noise (n_{\max}) / pixel	Small noise condition		Large noise condition	
	5	10	15	20
Max radius without filter (M_1)	20276	23328	67349	107750
Min radius without filter (M_2)	7315	8337	5908	4813
Max radius with filter (M_3)	15865	19252	16454	17543
Min radius with filter (M_4)	7315	9075	5908	9188
Radius range without filter ($R_1 = M_1 - M_2$)	12961	14991	10546	102937
Radius range with filter ($R_2 = M_3 - M_4$)	8550	10195	10546	8355

Table 5.2 - The comparison of radius range between using filter and without using filter

Maximum Noise (n_{max}) = 5 pixels

Without filter

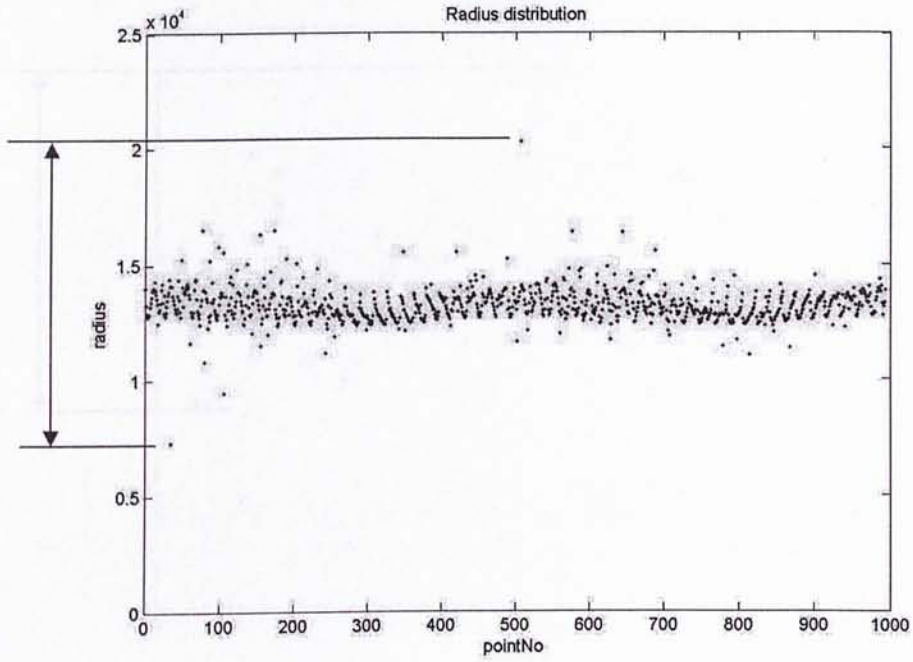


Figure 5.6a1

With filter

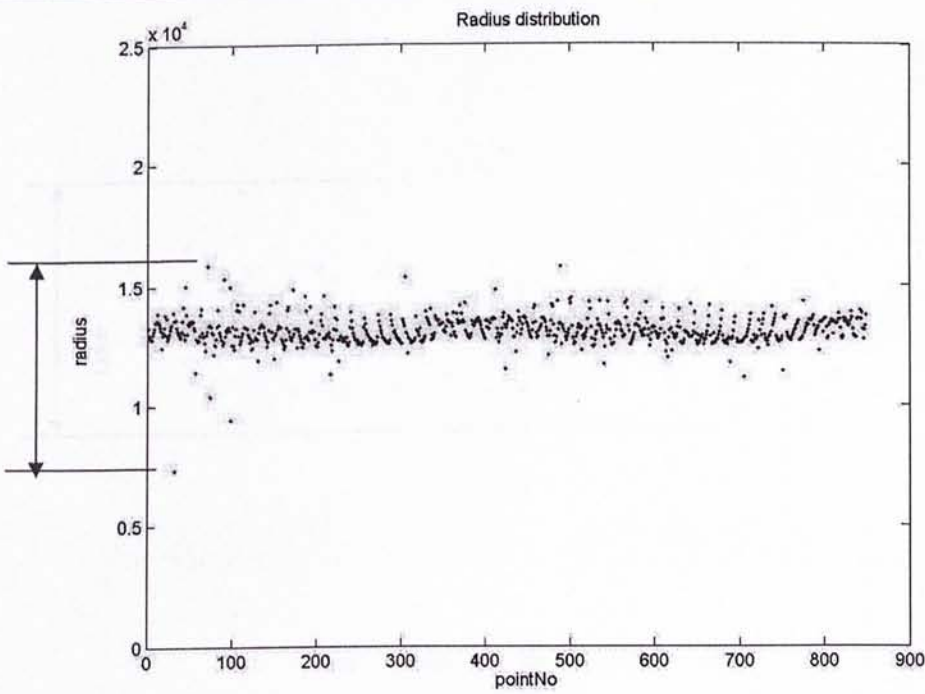


Figure 5.6a2

Figure 5.6a - Radius distribution with noise

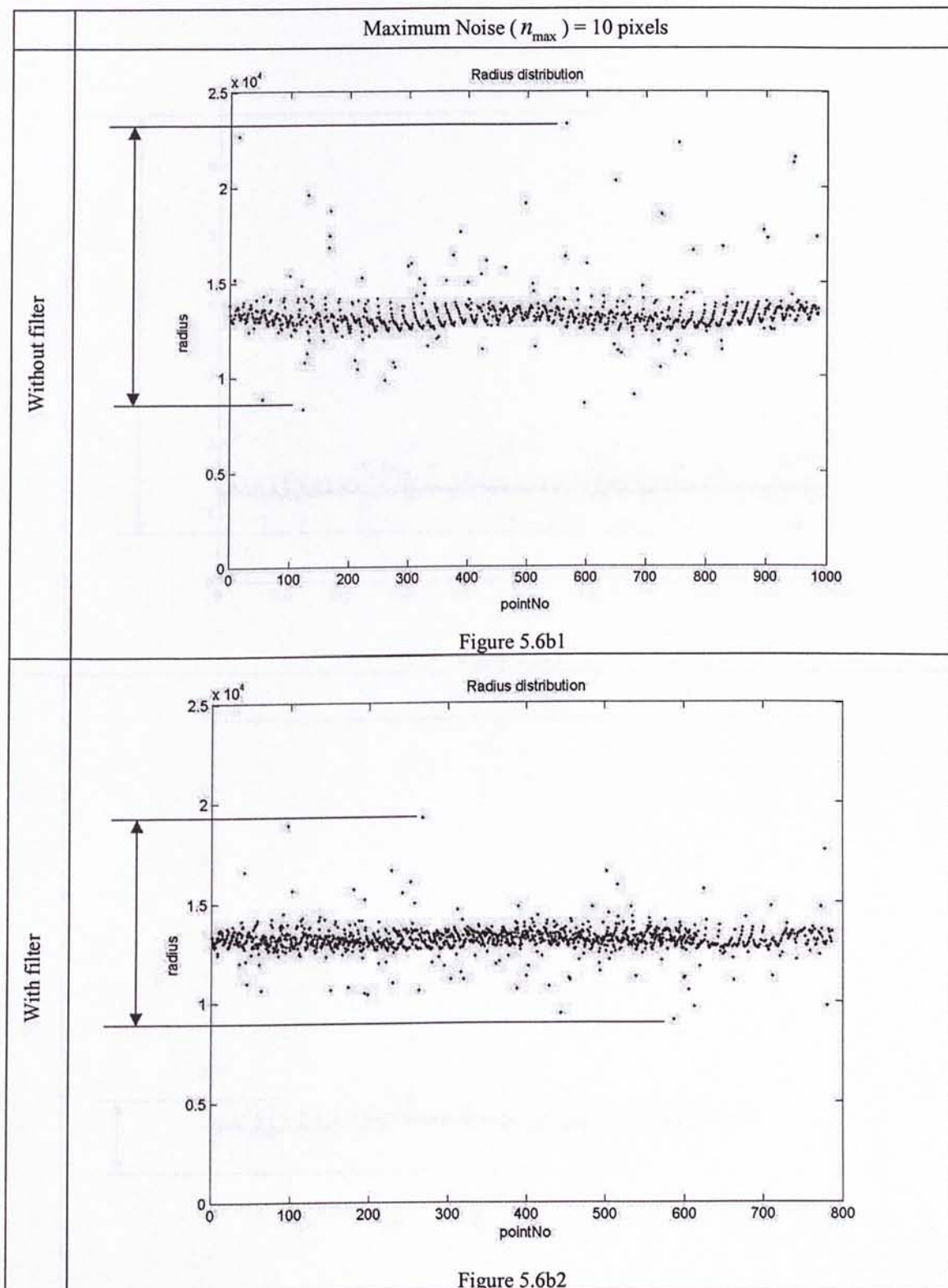


Figure 5.6b1

Figure 5.6b2

Figure 5.6b - Radius distribution with noise

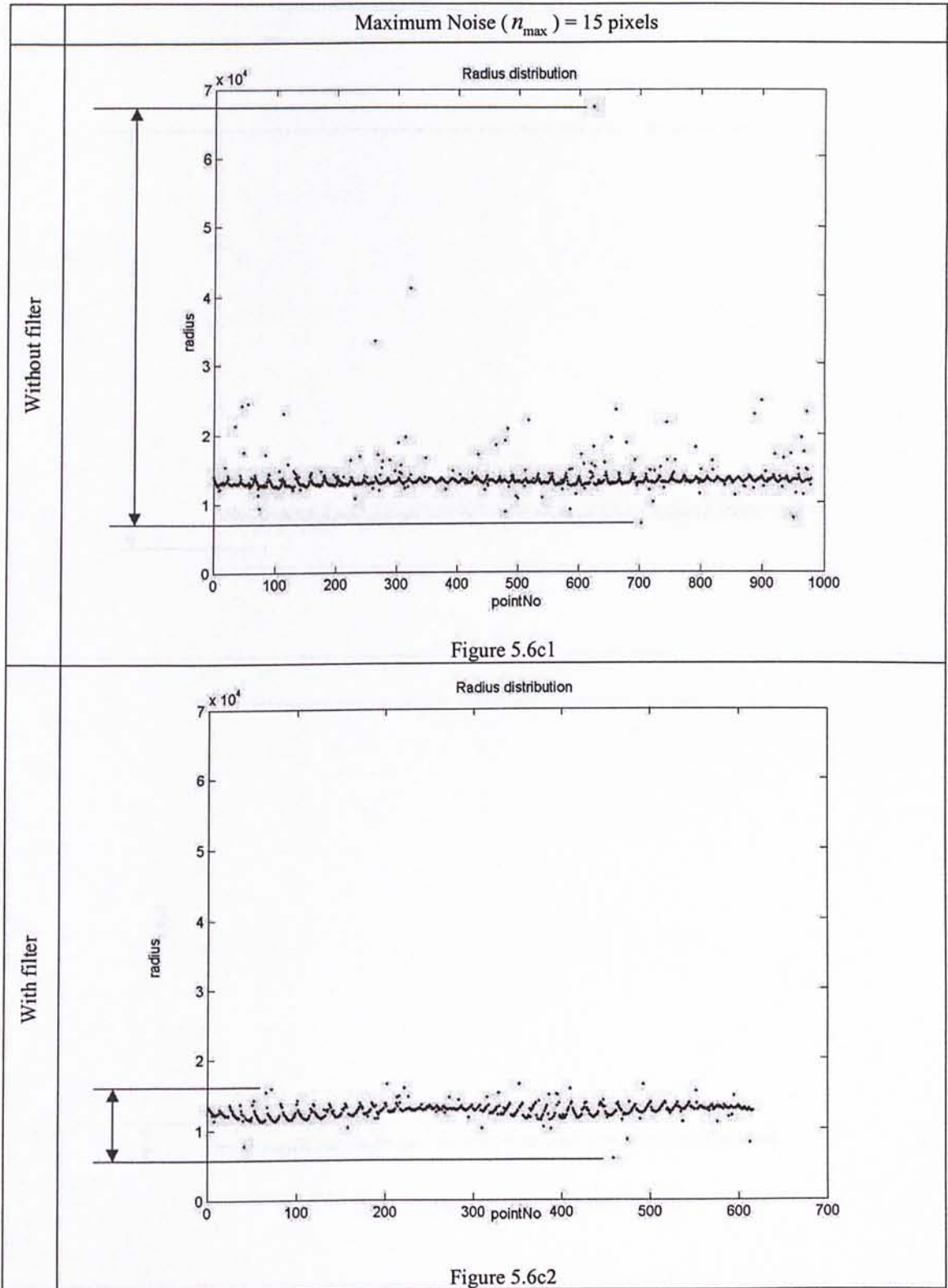


Figure 5.6c - Radius distribution with noise

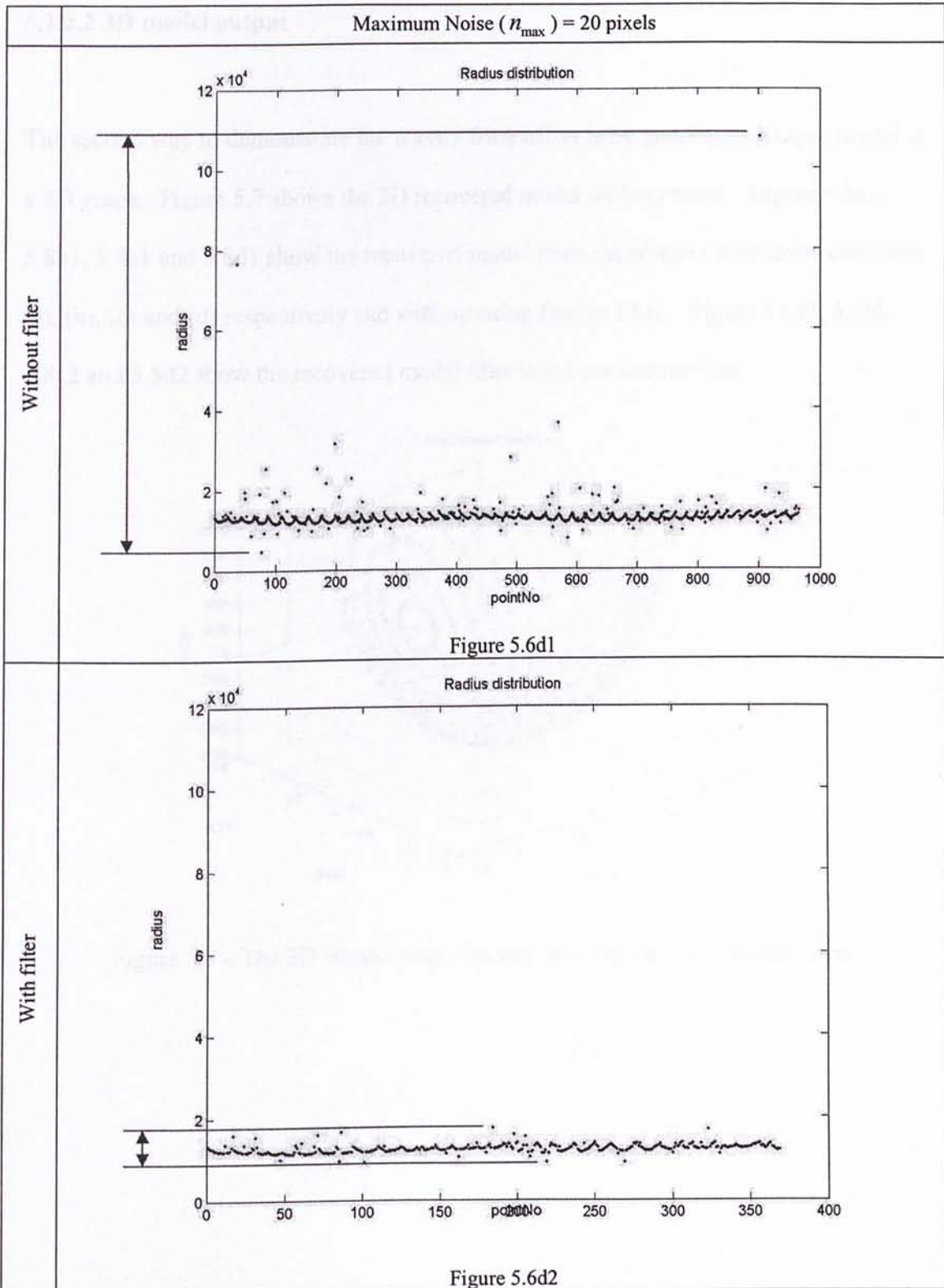


Figure 5.6d - Radius distribution with noise

5.1.2.2 3D model output

The second way to demonstrate the feature filter effect is by plotting the output model in a 3D graph. Figure 5.7 shows the 3D recovered model without noise. Figure 5.8a1, 5.8b1, 5.8c1 and 5.8d1 show the recovered model from the pictures with noise condition (a), (b), (c) and (d) respectively and without using feature filter. Figure 5.8a2, 5.8b2, 5.8c2 and 5.8d2 show the recovered model after using the feature filter.

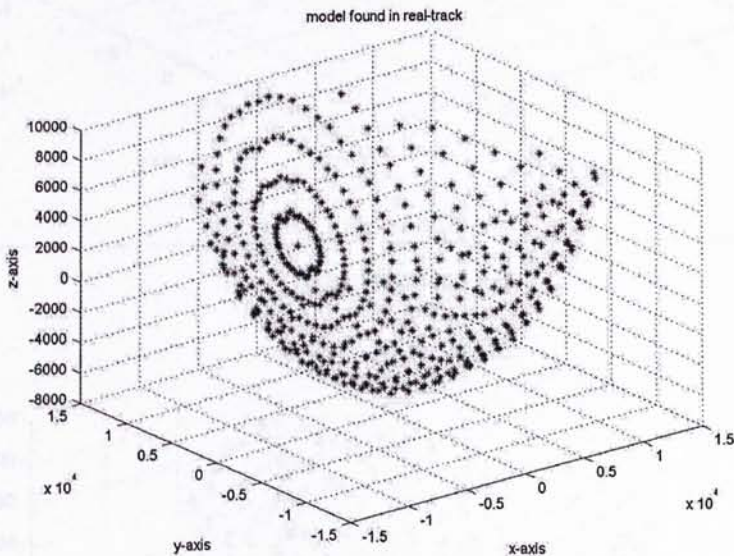


Figure 5.7 - The 3D feature graph for the recovered model without noise

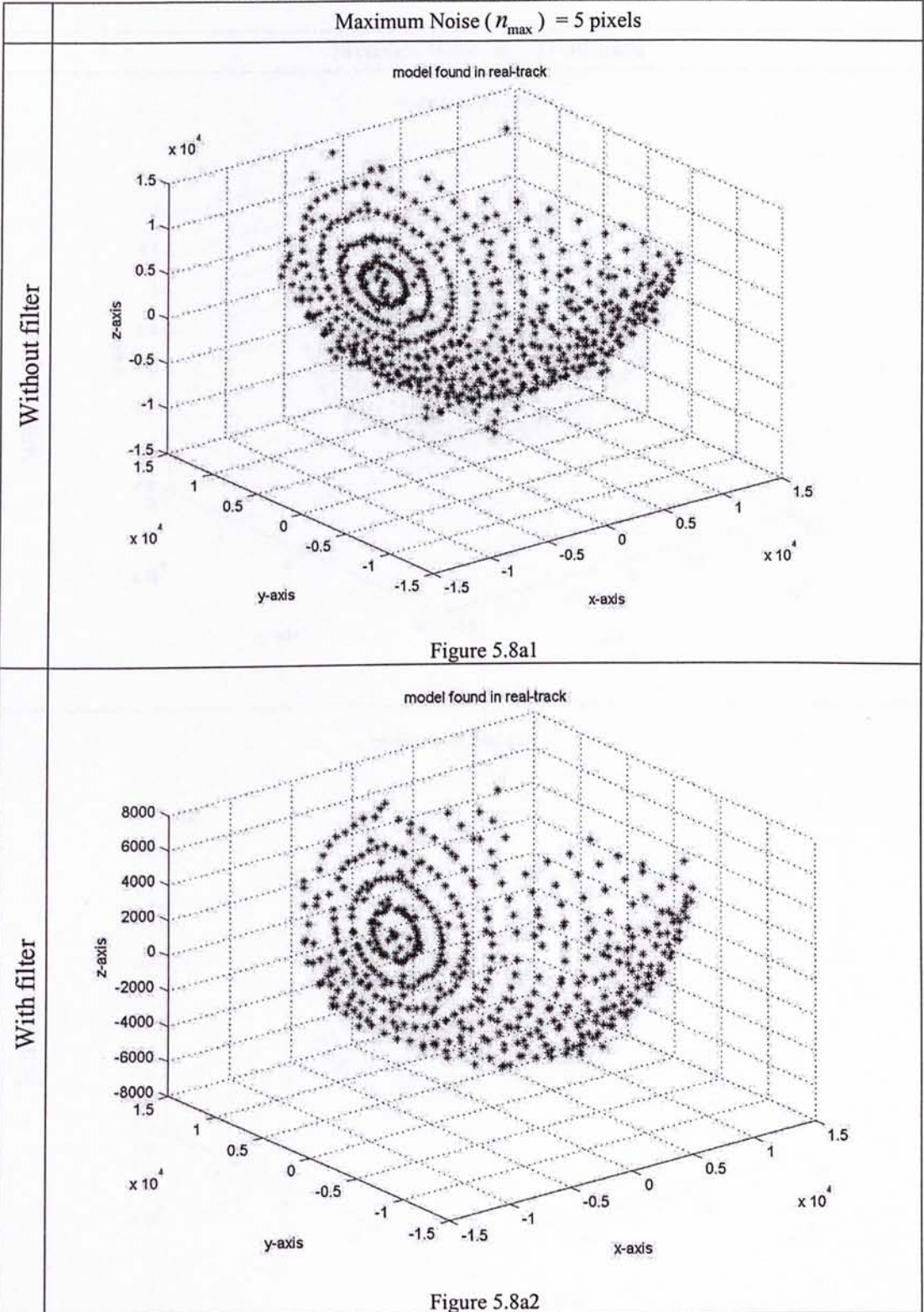


Figure 5.8a - The 3D features of the recovered model with noise

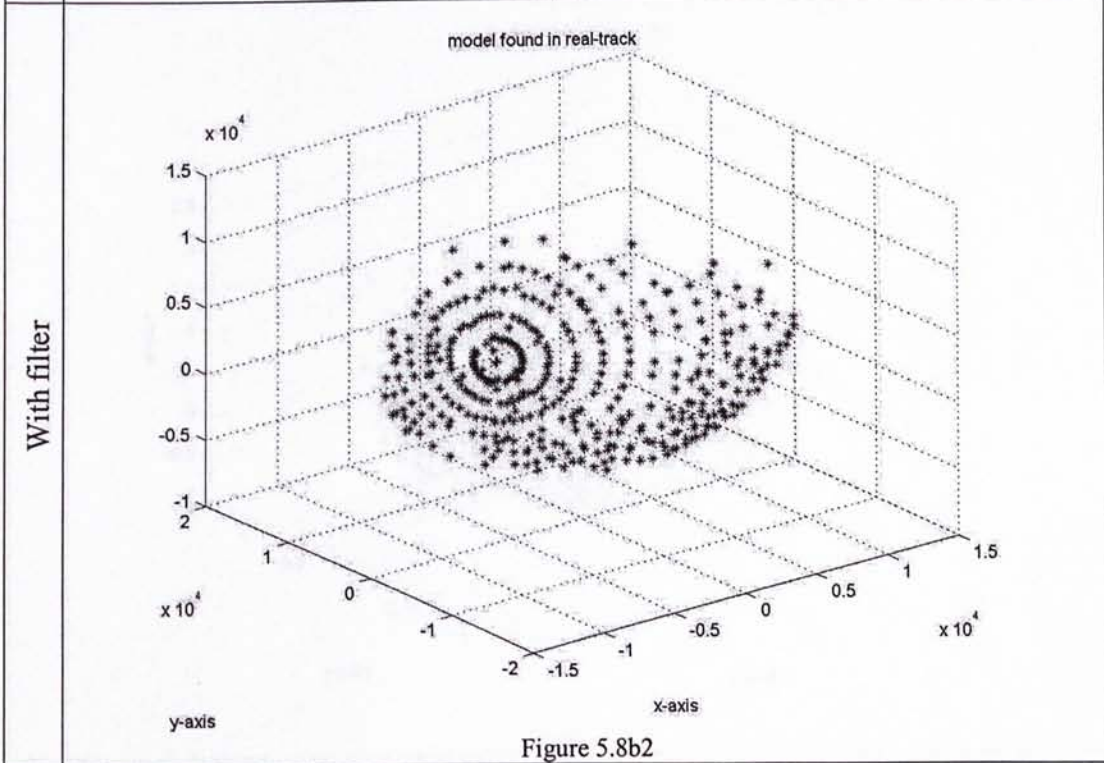
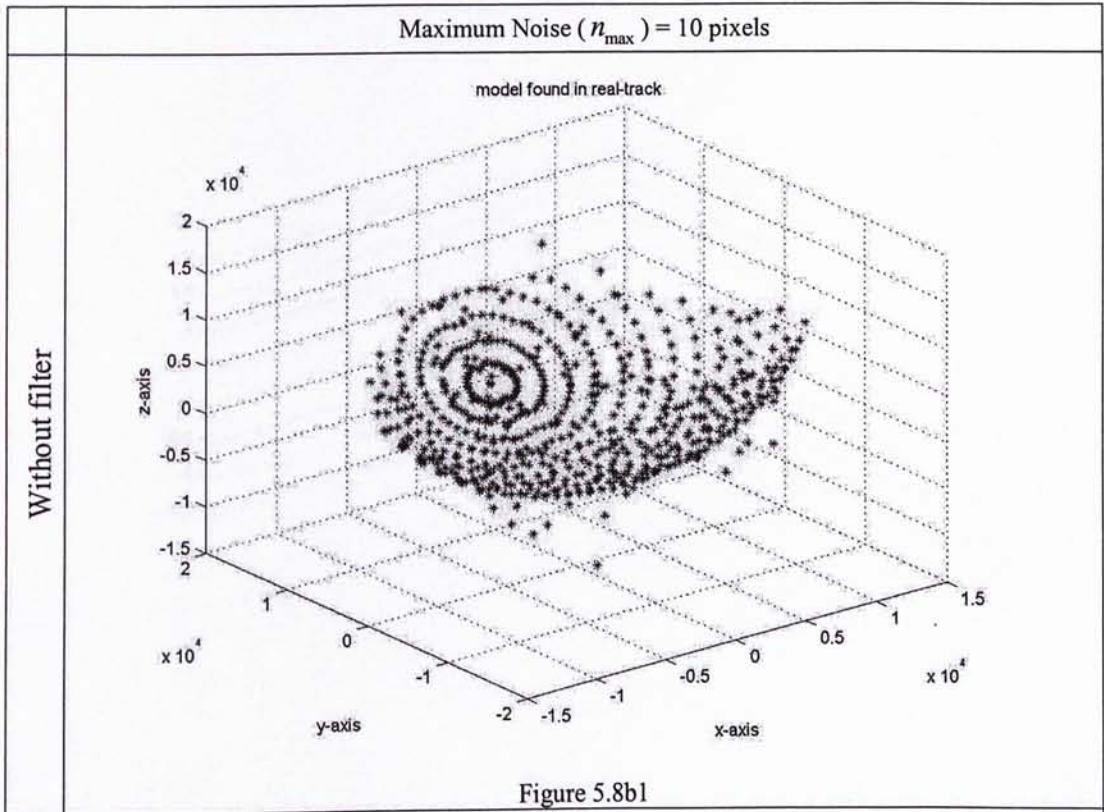


Figure 5.8b - The 3D features of the recovered model with noise

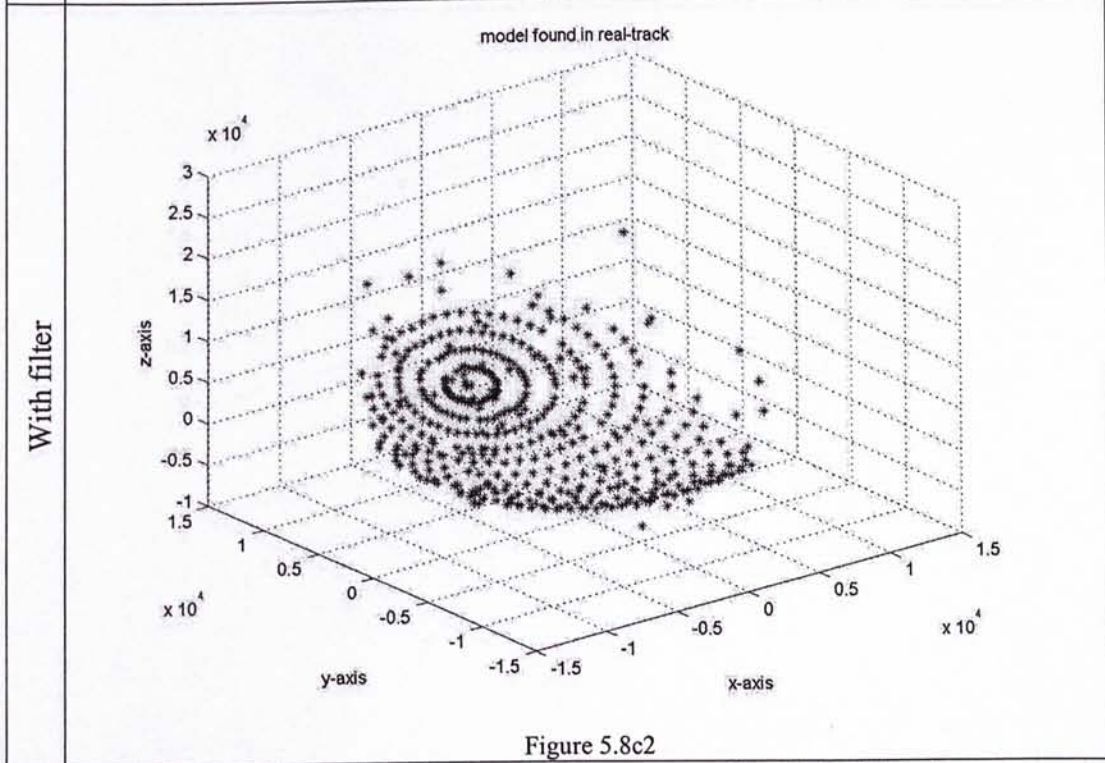
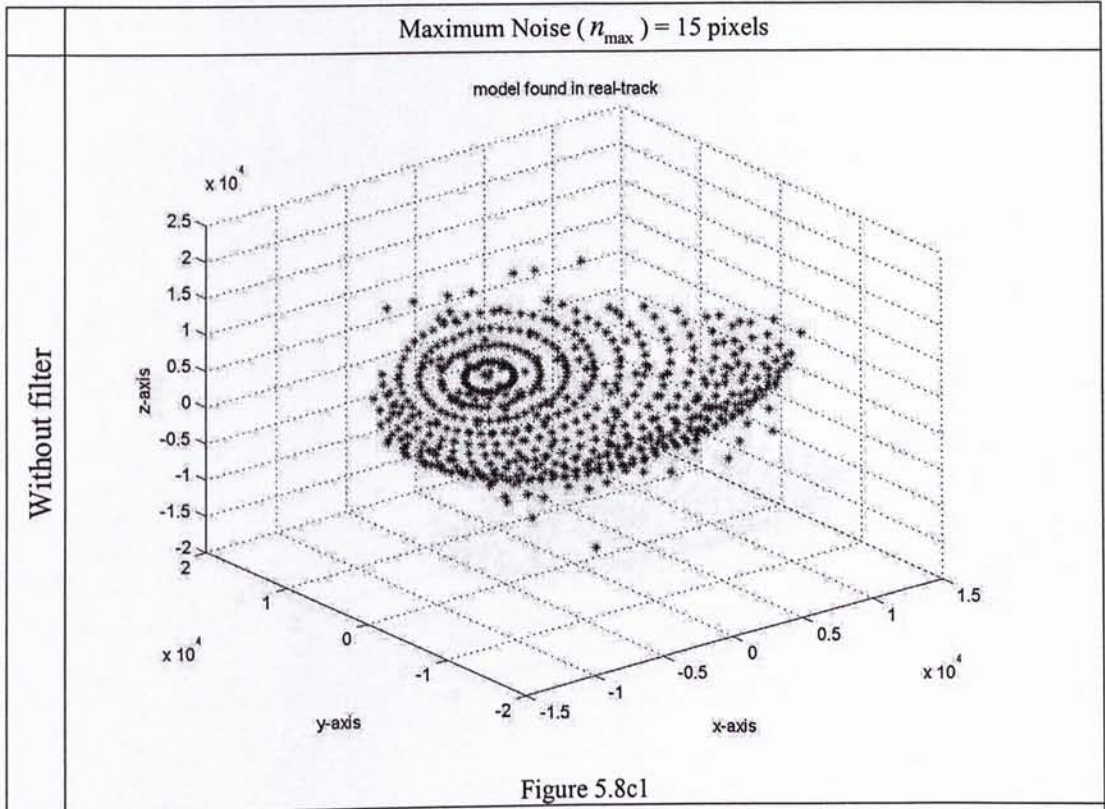


Figure 5.8c - The 3D features of the recovered model with noise

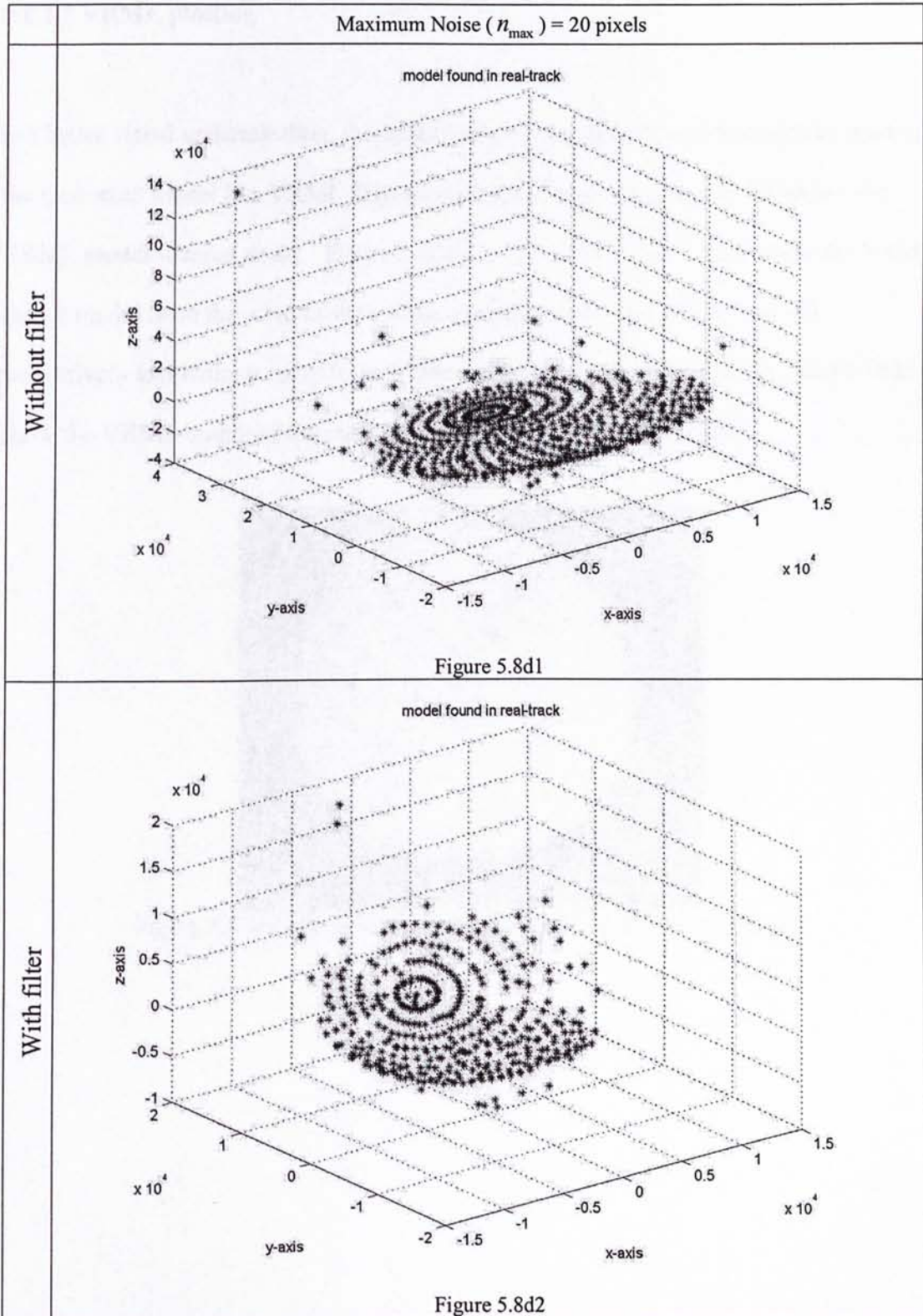


Figure 5.8d - The 3D features of the recovered model with noise

5.1.2.3 VRML plotting

For better visual understanding, the improvement of the filter is demonstrated by plotting the recovered model in a VRML diagram with the texture map. Figure 5.9 shows the VRML model without noise. Figure 5.10a1, 5.10b1, 5.10c1 and 5.10d1 show the VRML output model from the pictures with noise testing condition (a), (b), (c) and (d) respectively and without using feature filter. Figure 5.10a2, 5.10b2, 5.10c2 and 5.10d2 show the VRML output of the recovered model after using feature filter.

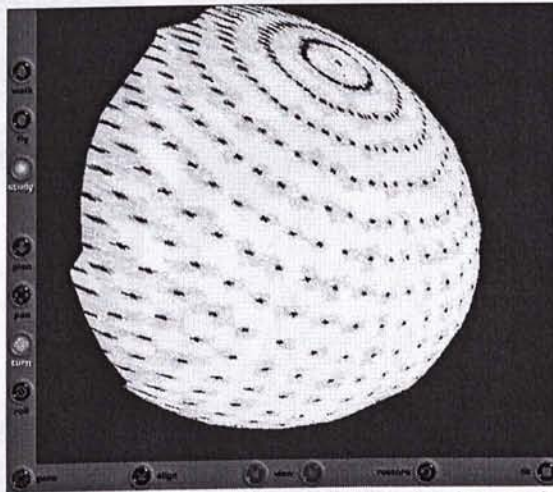


Figure 5.9 VRML plotting of the recovered sphere without noise

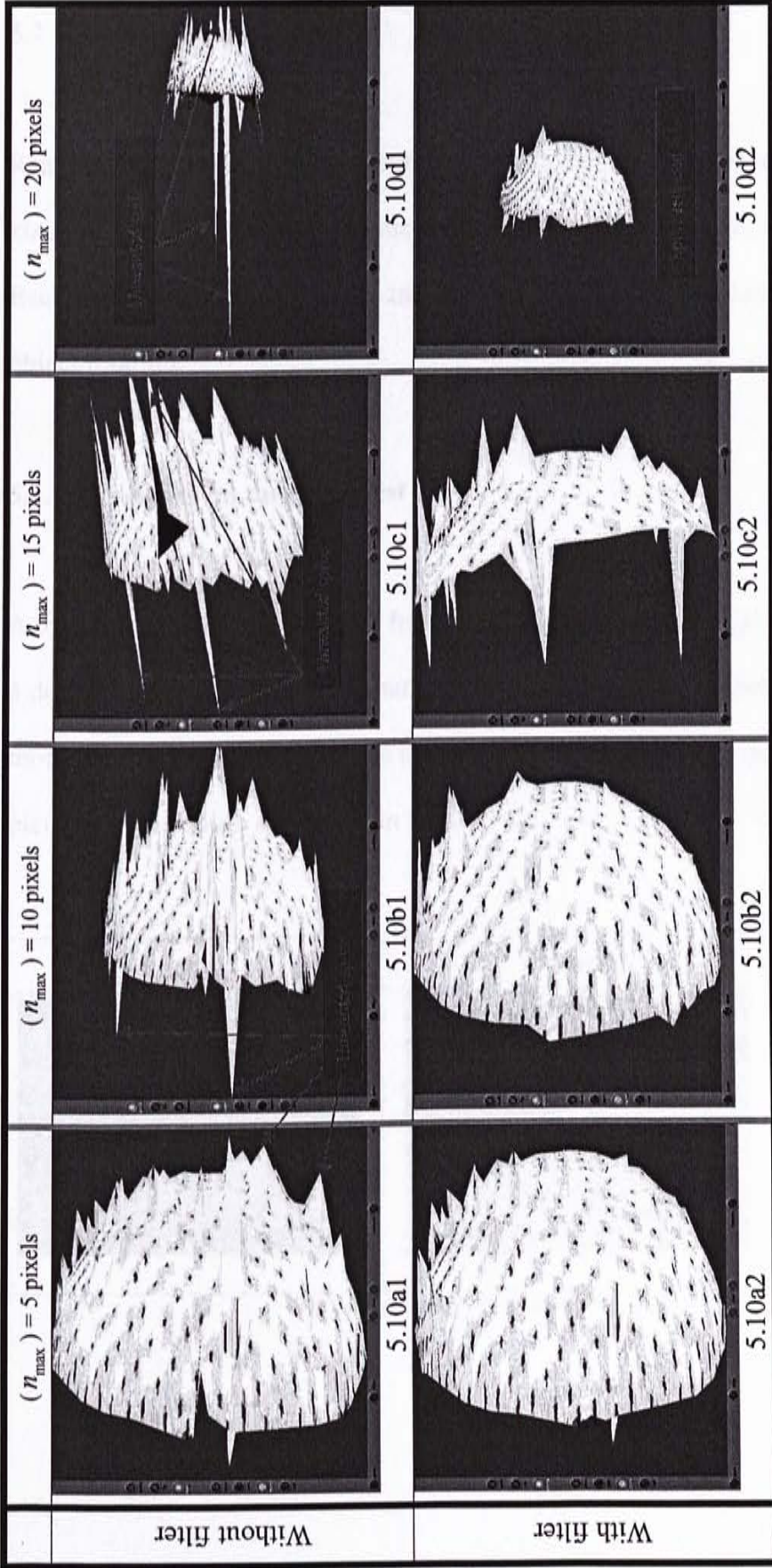


Figure 5.10 – Comparison of recovered model in VRML plot

5.2 Real Image testing

Real images are used to test the silhouette filter algorithm. The tests include different kinds of objects. Those objects include a toy robot, toy house, flask and toy tomb. Each object is put on a turntable and pictures are captured by a camera during the object rotation.

5.2.1 Toy house on turntable test

A toy house is put on a turntable for testing. A camera captures pictures in every 2 to 3 degrees during the turntable rotation. A comparison between the recovered 3D model results before and after the use of the filter is performed. Some samples of the pictures of the house are shown in Figure 5.11.

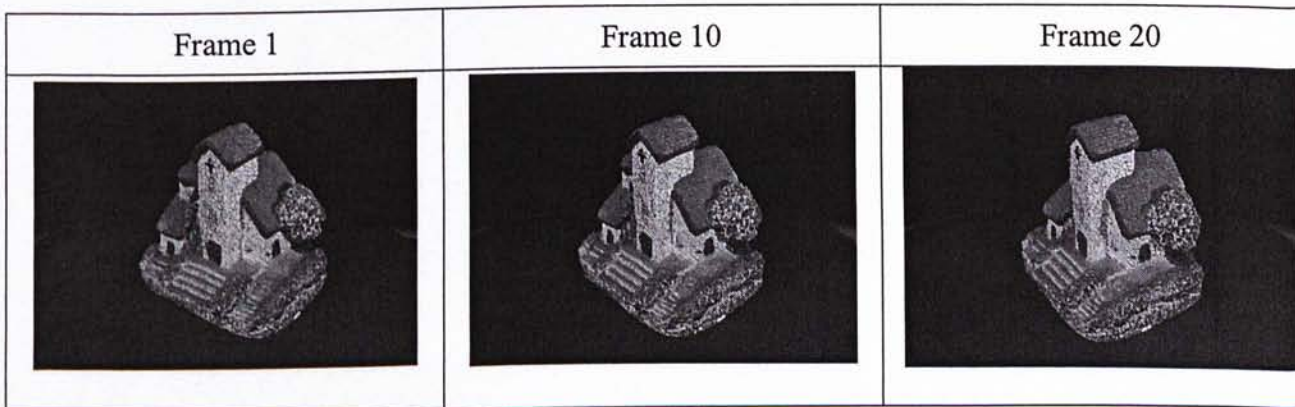
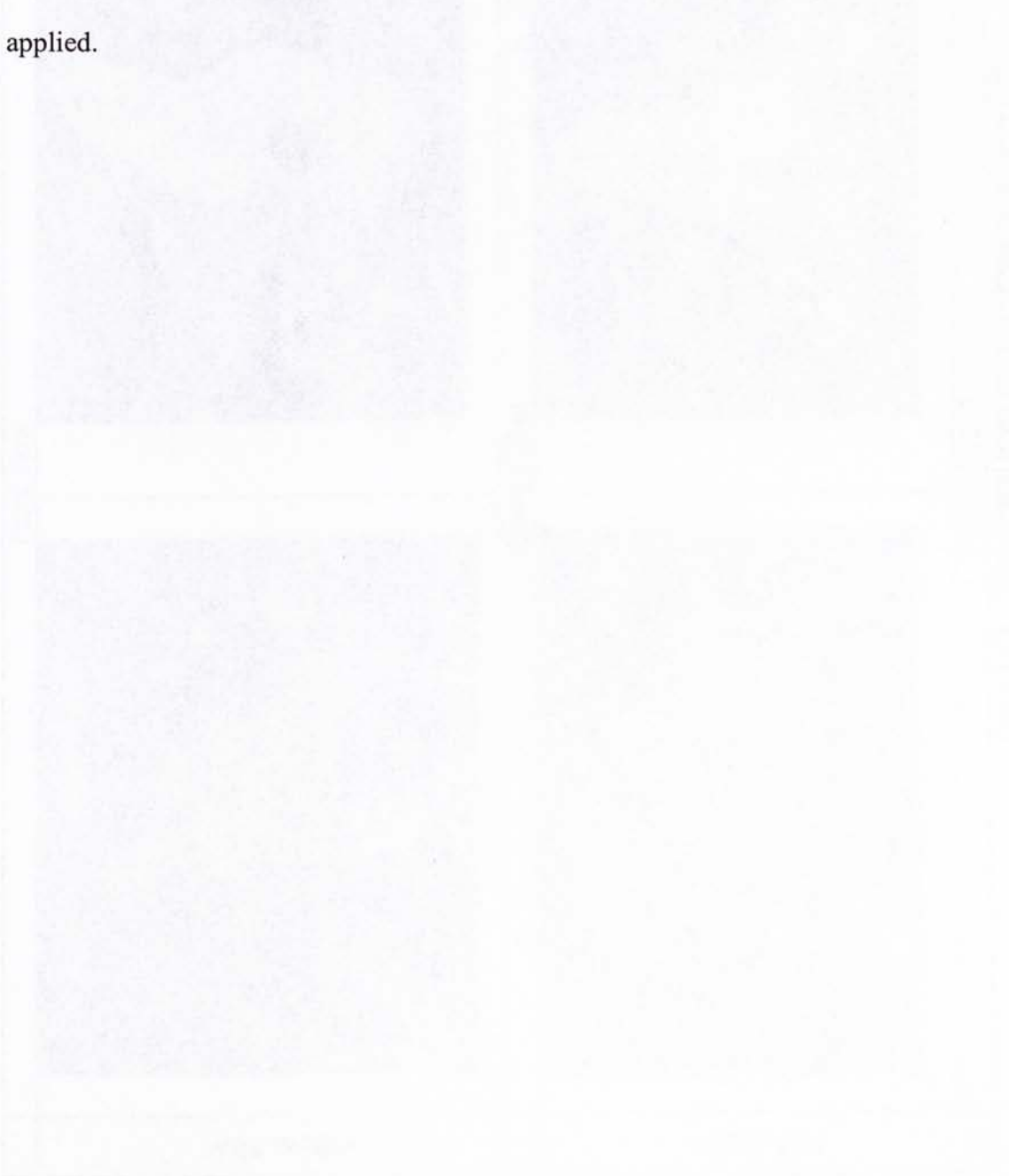


Figure 5.11 - Sample pictures of house test

The pictures are then feed into the interleave bundle adjustment algorithm. The results of the 3D model before and after using our filter are shown in figure 5.12.

Figure 5.12a1 and 5.12b1 display the texture mapped VRML results of the recovered model without using our feature point selection filter. Figure 5.12a2 and 5.12b2 display the result after applied the feature filter. It is clearly shown that a spike is located at the top and the bottom of the house and it is removed after the filter is applied.



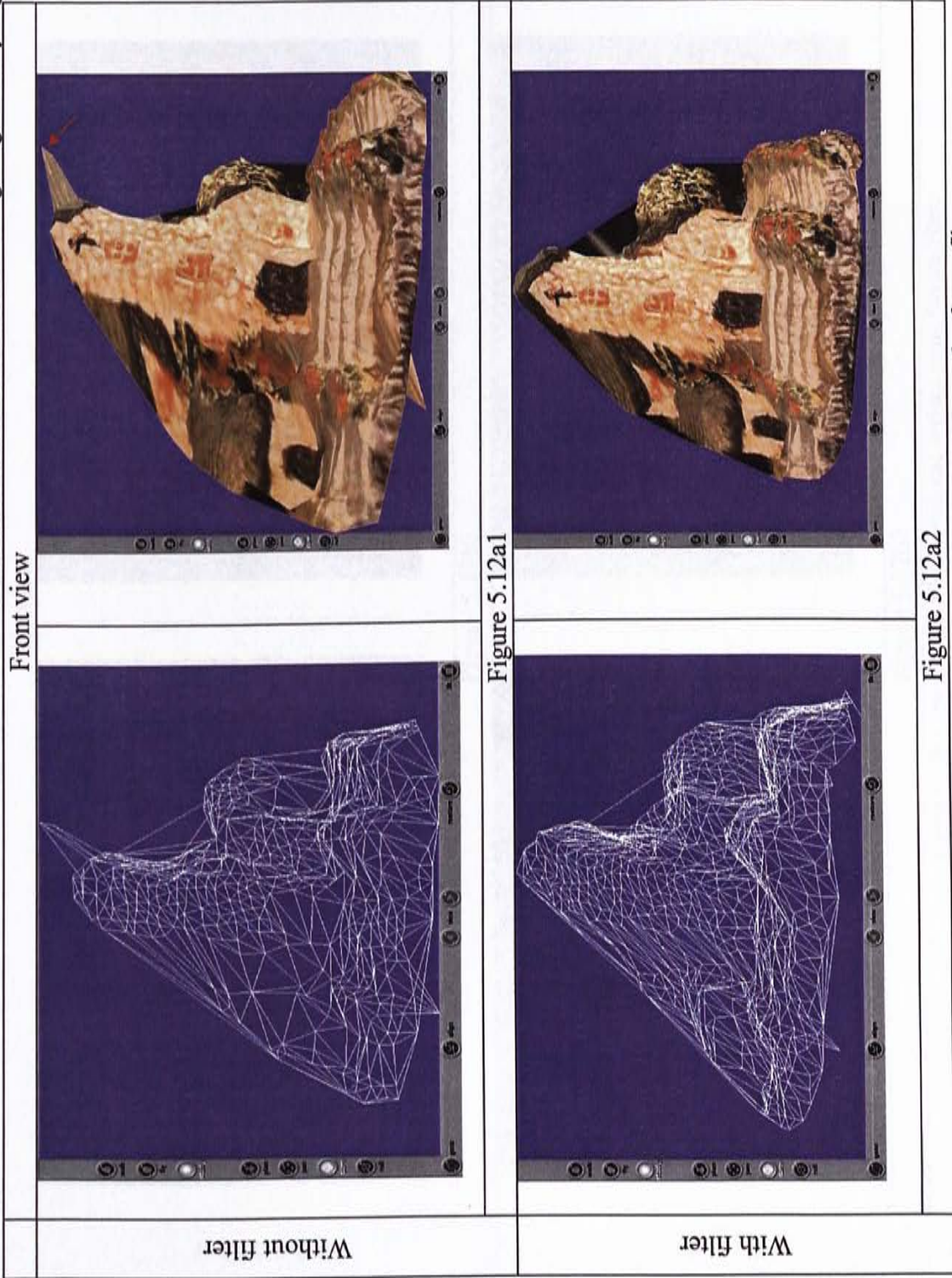
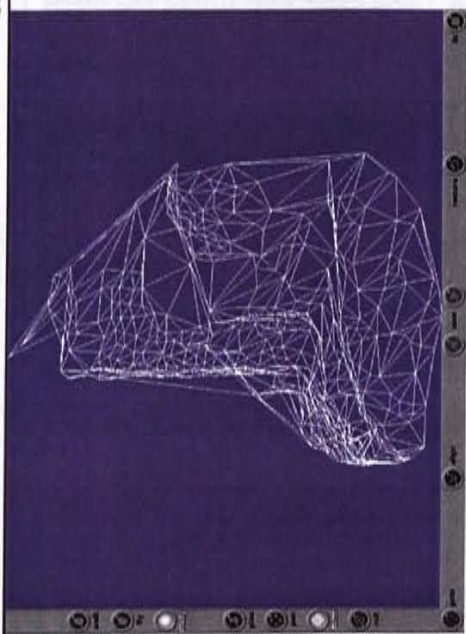


Figure 5.12a2

Figure 5.12a - The recovered house with and without using feature filter.

Side view



Without filter

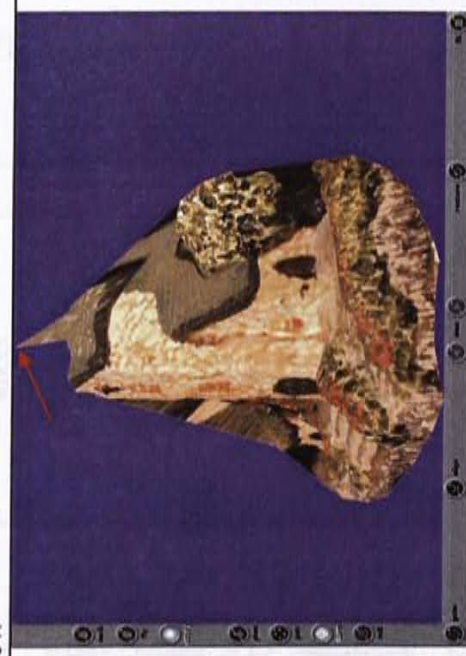
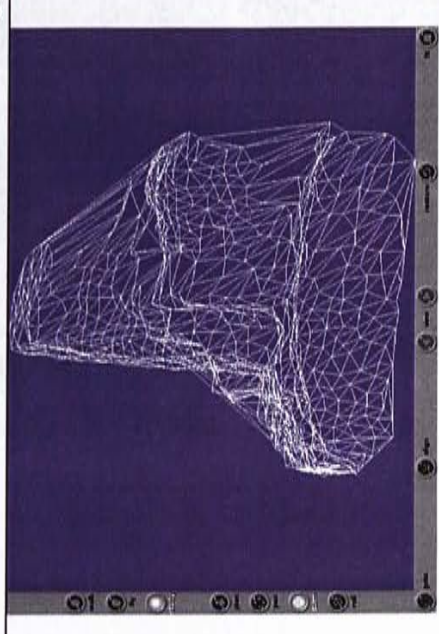


Figure 5.12b1



With filter

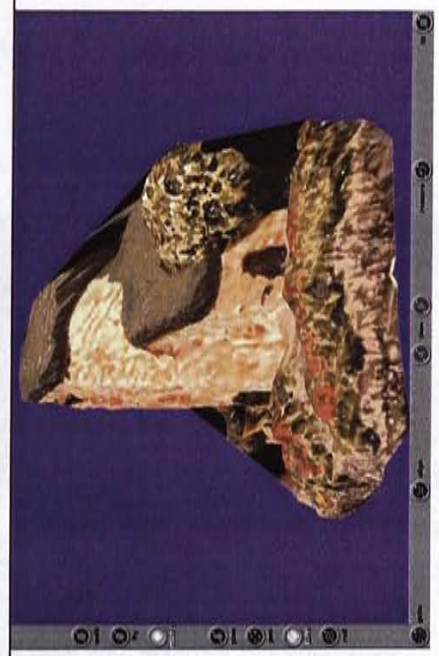


Figure 5.12b2

Figure 5.12b - The recovered house with and without using feature filter.

5.2.2 Other tests on turntable

To test the flexibility of the algorithm, different objects are using for testing. Figure 5.13 shows the sample original pictures of a toy robot, flask and tomb respectively. The comparison of the result is shown in Figure 5.14 and Figure 5.15.


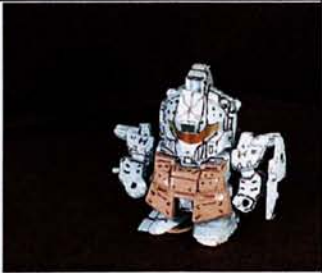
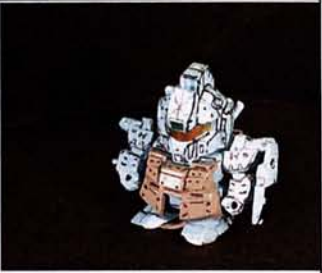



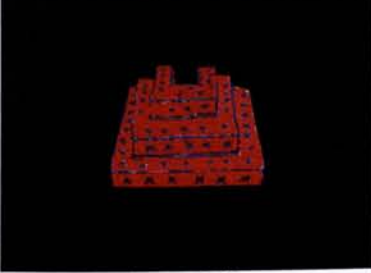
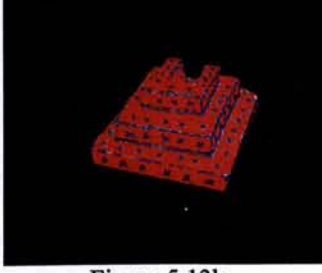
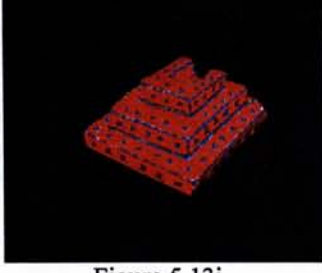
	Frame 1	Frame 10	Frame 20
Robot	 <p>Figure 5.13a</p>	 <p>Figure 5.13b</p>	 <p>Figure 5.13c</p>
Flask	 <p>Figure 5.13d</p>	 <p>Figure 5.13e</p>	 <p>Figure 5.13f</p>
Tomb	 <p>Figure 5.13g</p>	 <p>Figure 5.13h</p>	 <p>Figure 5.13i</p>

Figure 5.13 - Original sample pictures for toy robot, flask and tomb respectively

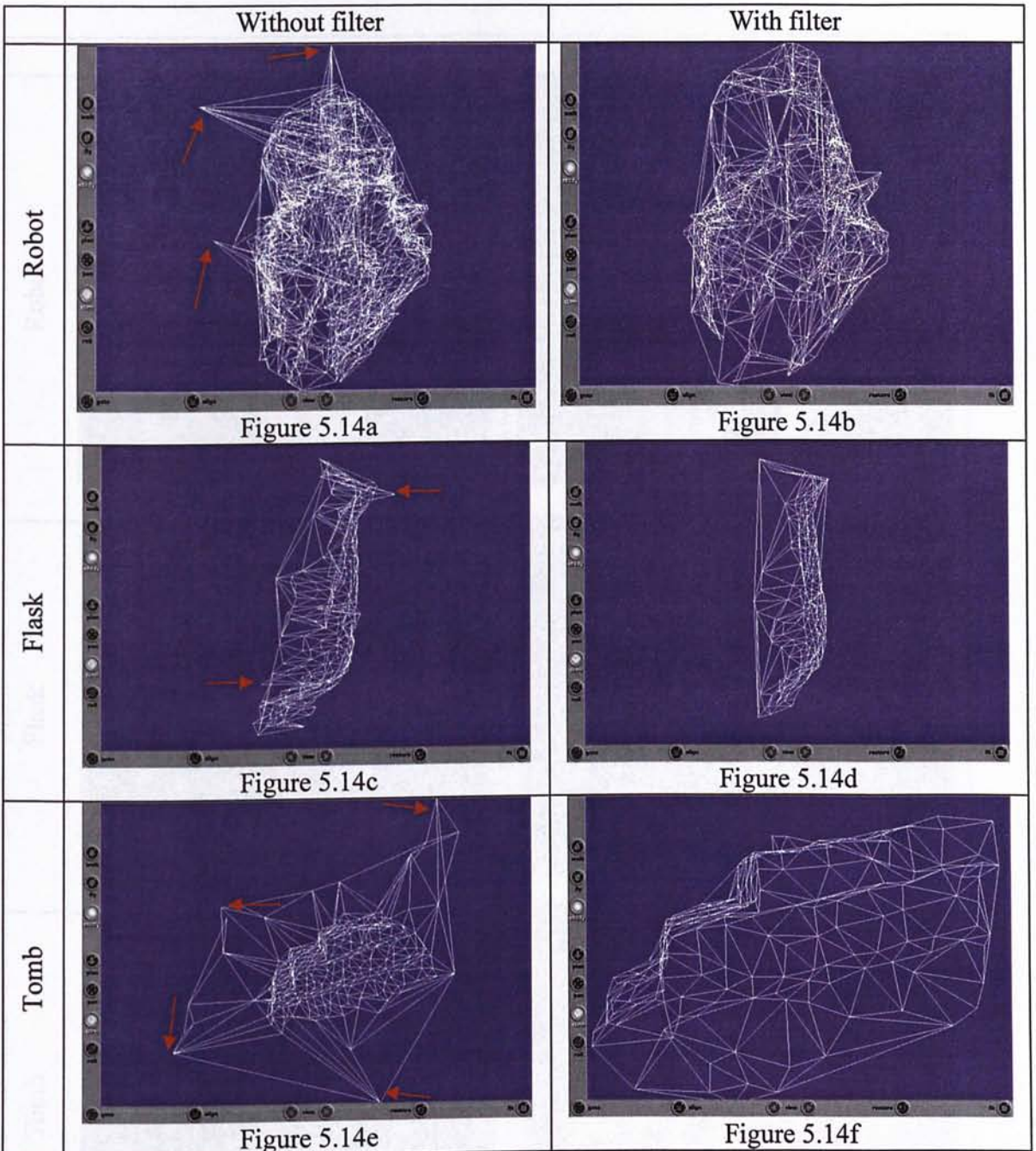


Figure 5.14 - The result of the toy house, flask and tomb with and without using feature filter (wire frame)

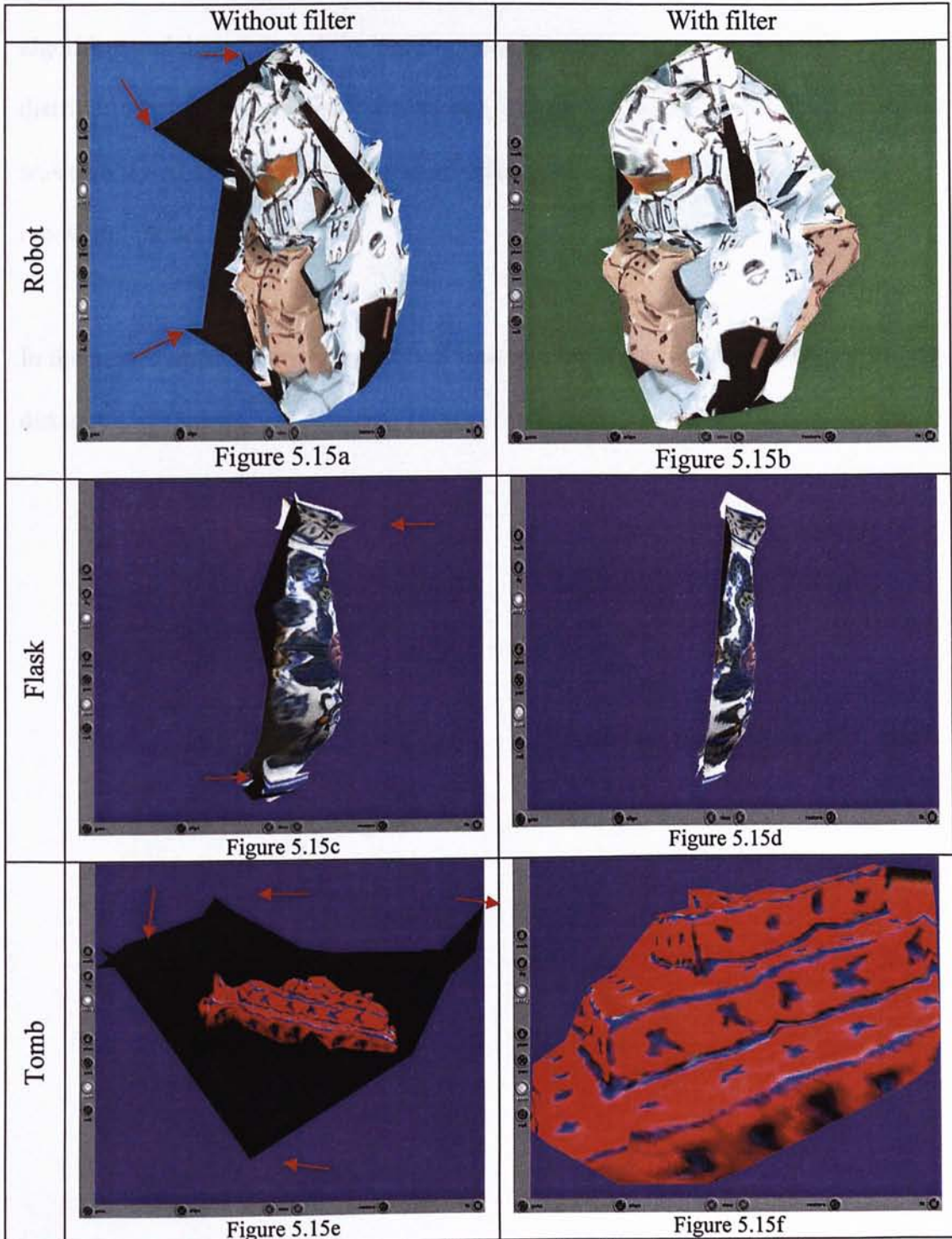


Figure 5.15 - The result of the toy robot, flask and tomb with and without using feature filter

In this chapter, we discussed a synthetic data simulation result of the feature filter algorithm and demonstrated the improvement of the filter by using the radii distribution of the sphere and the 3D texture mapped VRML diagram. The algorithm was also tested in sequential real pictures and significant improvement on the 3D recovered model was found.

In the next chapter, a conclusion of our work and the future working direction will be discussed.

Chapter 6 – Conclusion and discussion

The work presented in this thesis dealt with a feature point selection problem for a realistic 3D model recovery algorithm for sequential images. A feature filter that is able to filter out the mismatch feature points is developed for the use of the interleave bundle adjustment algorithm. We implemented the algorithm and applied successfully to synthetic data simulation and real images.

In this thesis,

- We discussed the effect of the mismatch feature points to the alternative bundle adjustment algorithm. We found that the mismatch points would generate some spikes on the recovered output model.
- We proposed a feature point selection algorithm by applying the silhouette clipping technique. A brief description of our proposal is shown as follows:
 1. A set of feature point is extracted for each image.
 2. Use the silhouette of the target object to pick out the out of bound feature points in each image.

6. Use the silhouette of the target object to filter out the out of bound feature points in each image again.
 7. Loop back to step 3 and reiterate the process until no out of bound feature is found or the maximum number iteration is exceeded.
- In our study, we found that our algorithm still has some areas for improvement.
 - Feature over-pruning – because the noisy feature affects both the estimated model and pose, if the pose result is incorrect good feature points are also pruned incorrectly.
 - Silhouette clipping sensitivity – because of the above over feature pruning problem, we need to add some tolerance to the silhouette boundary to avoid over pruning. However, how to set the tolerance parameters to optimize the performance needs future study.

Future directions:

Selection of correct feature points for a corresponding object in Structure from motion (SFM) is my main research interest. Many structure-from-motion algorithms nowadays are able to solve the problem of a single moving object only. However, in

moving objects situation. The main reason is that many existing approaches found the pose and model in separate processes. However, by using the interleave bundle adjustment algorithm, it would be possible to find a reasonable solution.

In the interleave bundle adjustment algorithm, because the pose and model information can be found in a same process, some clustering techniques in the process is able to be applied so that multi-moving objects problem is possible to be dealt with. The idea for multiple objects structure from motion is described as follows:

During the pose estimation phase of the alternative bundle adjustment algorithm, instead of estimating the pose angles by using a standard least square method, we can use a clustering tool, for example K-mean clustering, to separate the feature points to two or more groups by using the residual error of each model point error. After the feature groups are formed, we can apply a standard least square method to find the $\hat{\theta}_i$ value of each group. Then we continue the original interleave bundle adjustment algorithm to find the model and pose of each feature group.

Although the theory of the above feature selection method is quite simple, it still

Reference:

- [1] M. Armstrong, A. Zisserman and P. Beardsley, "Euclidean structure from uncalibration images, *Proc. British Machine Vision Conference* 1994.
- [2] M. Armstrong, A. Zisserman and R. Hartley, "Euclidean Reconstruction from Image Triplets", *Computer Vision – ECCV'96, Lecture Notes in Computer Science, Vol. 1064, Springer-Verlag, pp. 3-16, 1996.*
- [3] A. Azerbayejani, B. Horowitz and A. Pentland, "Recursive estimation of structure from motion using relative orientation constraints", *Proceedings of the International Conference for Computer Vision and Pattern Recognition, IEEE Computer Society Press, pp. 294-299, June 1993.*
- [4] B. Ronen and S. Ullman, "The alignment of objects with smooth surfaces," in *Proc. Second Int. Conf. Comput. Vision*, Dec. 1988.
- [5] P. Beardsley, A. Zisserman and D. Murray, "Sequential Updating of Projective and Affine Structure from Motion", *International Journal of Computer vision (23), No. 3, pp.235-259, Jun-Jul 1997.*
- [6] P. Beardsley, P. Torr and A. Zisserman "3D Model Acquisition from Extended Image Sequences", *Computer Vision – ECCV'96, Lecture Notes in Computer Science, Vol. 1065, Springer-Verlag, pp 683-695, 1996.*
- [7] A. Blake and R. Cipolla, "Robust estimation of surface curvature from deformation of apparent contours," in *Proc. First Euro. Conf. Comput. Vision*, 1990.
- [8] R. Cipolla and A. Blake, "The dynamic analysis of apparent contours," in *Proc. Third Int. Conf. Comput. Vision*, 1990.
- [9] B. Boufama and R. Mohr, "Epipole and fundamental matrix estimation using virtual parallax", *Proc. International Conference on Computer Vision, pp.1030-1036, 1995.*

pp. 222-229, Osaka, Japan, 1990.

[13] O. Faugeras, L. Quan and P. Sturm, "Self-Calibration of a 1D Projective Camera and Its Application to the Self-Calibration of a 2D Projective Camera", *Computer Vision – ECCV'98, vol.1, Lecture Notes in Computer Science, Vol. 1406, Springer-Verlag*, pp. 36-52, 1998.

[14] O. Faugeras and B. Mourrain, "on the geometry and algebra of point and line correspondence between n images", *Proc. International Conference on Computer Vision*, pp. 951-962, 1995.

[15] O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig", *Computer Vision – ECCV'92, Lecture Notes in Computer Science, Vol, 588, Springer-Verlag*, pp. 563-578, 1992.

[16] Faugeras, O. and R. Keriven, "Complete Dense Stereovision Using Level Set Methods". In: *Proc. 5th European Conf. On Computer Vision*. Pp. 379-393, 1998.

[17] A. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences", *Computer Vision – ECCV'98, vol. 1, Lecture Notes in Computer Science, Vol. 1406, Springer-Verlag 1998*, pp. 311-326, 1998.

[18] P. Giblin and R. Weiss, "Reconstruction of surface from profiles," in *Proc. First Int. Conf. Comput. Vision*, Dec. 1986.

[19] R. Hartley, "Cheirality invariants", *Proc. D.A.R.P.A Image Understanding Workshop*, pp. 743-753, 1993.

[20] R. Hartley, "Projective reconstruction from line correspondences", *Proc. IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Press*, 1994.

[21] R. Hartley, "A linear method for reconstruction from points and lines", *Proc. International Conference on Computer Vision*, pp. 882-887, 1995.

- [25] A. Heyden, *Geometry and Algebra of Multiple Project Transformations*, Ph.D. thesis, Lund University, 1995.
- [26] D. Jacobs, "Linear Fitting with Missing Data; Applications to Structure-from-Motion and to Characterizing Intensity Images", *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society press, pp. 206-212, 1997.
- [27] Jebara, T.; Azarbayejani, A.; Pentland, A. "3D structure from 2D motion", *IEEE Signal Processing Magazine*, Volume: 16 Issue: 3 , Page(s): 66 -84, May 1999 .
- [28] Jianbo Shi and Carlo Tomasi "Good Feature to Track" *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593-600, 1994.
- [29] Okutomi, M. and T. Kanade, "A multiple-baseline stereo". *IEEE Trans. Pattern Anal. Machine Intell.* 15(4), 353-363, 1993.
- [30] E. Kuppaa, "Zur ermittlung eines objektes aus zwei perspektiven mit innerer orientierung", *Sitz-Ber. Akad. Wiss., Wien math naturw. Abt. Iia* 122:1939-1948, 1913.
- [31] K.N. Kutulakos and S.M. Seitz , "What Do N Photographs Tell Us About 3D Shape?", *University of Rochester Computer Science Department Technical Report #680*, January 1988.
- [32] H.S. Lim and T.O. Binford, "Curved surface reconstruction using stereo correspondence," in *Proc. Image Understanding Workshop*, 1988, pp.809-819.
- [33] D. G. Lowe. "Three-dimensional object recognition from single two-dimensional images". *Artificial Intelligence*, 31:355--395, 1987.
- [34] Q.-T. Luong and O. Faugeras, "Self Calibration of a moving camera from point correspondences and fundamental matrices", *International Journal of Computer Vision*, vol. 22-3, 1997.

Geometry from Profiles under Circular Motion” *IEEE Transactions on pattern Analysis and Machine Intelligence*, Vol. 23, No 6, p. 604-616, June 2001.

[39] R. Mohr, F. Veillon and L. Quan, “Relative 3D reconstruction using multiple uncalibrated images”, *Proc. International Conference on Computer Vision, IEEE Computer Soc. Press*, pp.543-548, 1993.

[40] T. Moons, L. Van Gool, M. Van Diest, and E. Pauwels, “Affine reconstruction from perspective image pairs”, in: *J.L. Mundy, A. Zisserman, and D. Forsyth (eds.), Applications of Invariance in Computer Vision, Lecture Notes in Computer Science, Vol. 825, Springer-Verlag*, pp. 297-316, 1994.

[41] T. Moons, L. Van Gool, M. Proesmans and E. Pauwels, “Affine reconstruction from perspective image pairs with a relative object-camera translation in between”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 18, no. 1, pp. 77-83, Jan 1996.

[42] M. Mühlich and R. Mester, “The Role of Total Least Squares in Motion Analysis”, *Proc. ECCV’98*, pp.305-321, 1998.

[43] M. Pollefeys, Self-calibration and metric 3D reconstruction from uncalibrated image sequences, PhD. thesis, K.U.Leuven, 1999.

[44] Régis Vaillant and Olivier D. Faugeras, “Using External Boundaries for 3-D Object Modeling,” *IEEE Transactions on pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, p. 157-173, February 1992.

[45] P.V. Sander, X. Gu, S.J. Gortler, H. Hoppe, J. Snyder. “Silhouette Clipping” *In Computer Graphics, Proceedings of SIGGRAPH 2000*, pp. 327-334, 2000.

[46] A. Shashua and S. Avidan, The Rank 4 constraint in Multiple View Geometry, *Proc. Computer Vision-ECCV’ 96, Springer-Verlag*, April 1996.

[47] A. Shashua, “Trilinearity in visual recognition by alignment”, *Computer Vision-ECCV’9, Lecture Notes in Computer Science, Vol.801, Springer-Verlag*, pp.479-

- [51] R. Szeliski and S. Kang, "Recovering 3D shape and motion from image streams using non-linear least-squares", *DEC technical report 93/3, DEC*, 1993.
- [52] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization approach", *International Journal of Computer Vision*, 9(2):137-154, 1992.
- [53] P. Toor and A. Zisserman, "Robust parametrization and computation of the trifocal tensor", *Image and Vision Computing*, 15(1997) 591-605.
- [54] P. Torr, P. Beardsley and D. Murray, "Robust Vision", *Proc. British Machine Vision conference*, 1994.
- [55] P. Torr, A. Fitzgibbon and A. Zisserman, "Maintaining Multiple Motion Model Hypotheses Over Many Views to Recover Matching and Structure", *Proc. International Conference on computer Vision*, Narosa Publishing house, pp485-491, 1998.
- [56] B. Triggs, "The geometry of projective reconstruction I: Matching constraints and the joint image", *Proc. International Conference on Computer Vision, IEEE Computer Soc. Press*, pp. 338-343, 1995.
- [57] Trucco and Verri, *Introductory techniques for 3-D computer vision*, Prentice Hall, 1998.
- [58] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections", *Nature*, 293:133-135, 1981.
- [59] R. Tsai and T. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces", *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol.6, pp.13-27, Jan. 1984.
- [60] Weng, J.; Ahuja, N.; Huang, T.S., "Optimal motion and structure estimation Pattern Analysis and machine Intelligence", *IEEE Transactions on, Volume: 15 Issue: 9, Page(s): 864 -884*, Sept. 1993.

France, 1996.

[64] C. Zeller and O. Faugeras, "Camera self-calibration from video sequences: the Kruppa equations revisited" *INRIA, Sophia-Antipolis, France, Research Report 2793*, 1996.

[65] Z. Zhang and Y. Shan. "Incremental Motion Estimation through Local Bundle Adjustment." *Technical Report MSR-TR-01-54, Microsoft Research*, May 2001. (<http://research.microsoft.com/~zhang/publications.htm#sec-techreports>).

[66] Z.Zhang, R. Deriche, O. Faugeras and O.-T.Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry", *Artificial Intelligence Journal*, Vol78, pp. 87-119, October 1995.

[67] A. Zisserman, D. Liebowitz and M. Armstrong, "Resolving ambiguities in auto-calibration", *Phil. Trans. R. Soc. Lond., A(1998) 356, 1193-1211*.

[68] B. Triggs, "The Absolute Quadric", *Proc. 1997 Conference on Computer Vision and Pattern Recognition*, IEEE Computer Soc. Press, pp. 609-614, 1997.

[69] Kass et al 1988. Kass, M., Witkin, A. and Terzopoulos, D., 1988. "Snakes: Active contour models." *Internat. J. Comput. Vision* 1, pp. 321-331

CUHK Libraries



004146310