# Automatic 3D Object Reconstruction from Multiple 2D Image Views

**ZHANG Wenbo**

**School of Electrical & Electronic Engineering**

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirements for the degree of
Doctor of Philosophy

**2011**

# Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

Jan 28, 2011
Date

Zhang Wenbo
ZHANG Wenbo

*To my family.*

# Acknowledgement

I would like to take this opportunity to express my appreciation and gratitude to several individuals, without whom I would not have enjoyed my study and completed this thesis. First of all, I would like to thank my supervisor, Associate Professor Eric Sung, for his inspiring encouragement, insightful suggestions and valuable advice. His guide helped me throughout my research work. I would also like to express my appreciation to Associate Professor Chua Chin Seng and Professor Wang Dan Wei for their invaluable support.

For plentiful discussions and exchanges of ideas throughout my study period, I would like to express my gratitude to Mrs. Zhu Yan, Dr. Kong Hui, Dr. Yang Xulei, Dr. Gao Xinting and Mr. Wang Zhimin for their kind assistance. Credits should also be given to the technicians in the computer vision laboratory Mr.Sow Peck Heng and Mr. Yuen Sien Huan for their logistic support.

Lastly, special appreciation to my wife, Dr. Zhang Jing, the thesis can not be made without her encourage and love.

# Abstract

The 3D reconstruction problem in computer vision is to acquire the third dimension information of a target object from its multiple 2D images. It is a naturally ill-posed problem. As a result, although intensive researching works have been done on this field, the 3D reconstruction is still an open problem.

The 3D reconstruction problem normally contains a series of steps including camera calibration, image matching and reconstruction. There are specific problems in each step, however, the results of previous step will also have strong effect on the ones in late stages. For example, the whole reconstruction system accuracy performance is depending on the step with worst error. This leads to the idea of considering the steps jointly. Moreover, most of existing image matching and reconstruction methods work on 2D images because they are the only information available. However, this approach will have difficulty to deal with the variations due to changes of viewpoint. So the task to overcome the variations and obtain dense and accurate reconstruction is another key problem for us. In addition, our research is also trying to develop algorithms for reconstruction with less user intervention, and a fully automatic one will be the ultimate goal.

In this thesis, we will firstly study the cooperation between feature detection and feature matching in order to improve the overall performance. The feature selected is the corner. To meet this objective, we redesign the standard corner detector to improve its accuracy and robustness. Also, we redesign the core function of corner

matching method by incorporating more information, and based on which we formulate our own energy function. In this way, we reduce the overall computational complexity and still obtain equal to or better results. Secondly, we propose an adaptive 3D correlation window which works in 3D space instead of in 2D image space. We formulate a scheme to integrate the adaptive 3D window searching with layered depth images. In this scheme, the information is acquired under different assumptions step by step until a general assumption that the object is pair-wise smooth. Our method, which can better cope with the variations of perspective distortion, pose variation and occlusion, yields results in dense and accurate reconstruction.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

A digitized 3D object model is essential to many fields of applications such as virtual reality, object recognition, human computer interface, interactive media, filmmaking and computer games. Acquiring 3D information of real world object from 2D images is a recurring problem in computer vision. This task can be accomplished by either active or passive approaches. The active approaches use an active source to project onto the 3D scene and use a camera or other sensors to receive the reflected source and enables the 3D surface to be reconstructed. The important advantage with active source is the avoidance of the stereo correspondence provide that is plaqueing the passive stereovision approach. The structured lighting and laser range imaging techniques are examples of active methods. Sabata [1] and Aggarwal [2] gave a good literature survey on recovering structure from active range sensors. The active approaches can produce accurate 3D face model but the user need to buy a special equipment. We are seeking a cheaper way by passive methods.

As for the passive approach, many methods have been proposed to obtain the 3D

shape from 2D images. One group is called structure-from-motion. They track features through video frames, then recover the 3d shape of the object. Nonlinear methods for recovering 3D information from tracked features over two or more frames have been proposed [3], [4]. A key idea, known as the eight-point algorithm [5], makes possible the recovering of motion and shape information by a linear method using eight feature points tracked over two frames. This eight-point algorithm, however, requires the knowledge of the full perspective camera calibration. Work by others make simplifying assumptions such as orthographic [6] [7] or affine assumptions [8] [9] [10]. Findings reported good performances provided the situations are close to the assumptions. Work in [11] shows the success of this kind of method. At the same time, the structure-from-motion method also suffer from a main inherent difficulty. On one side, to find the corresponding points across frames with big pose variation, they need to detect some reliable features and track them; on the other side, to recover a 3D model, the dense matching across images is needed.

Another group, known as stereo vision [12] [13], has been quite successful in 3D reconstruction under specific condition. The stereo cameras are fully calibrated first, then by finding the correspondence between two images, the 3D shape can be recovered. However, one disadvantage of stereo vision [14] is the tradeoff between the length of baseline and the stereo vision sensitivity to noise. For a short-baseline stereo, the correspondence is relatively easy to find due to the small difference between two images. But at the same time, the short-baseline will make the triangulation unstable. In another words, a small disparity error will result in a large error in triangulation. On the other hand, when a wide-baseline stereo is used, the correspondence will be hard to find although the triangulation becomes more stable. This is in fact the same problem that SFM faces. Actually, stereovision can be viewed as structure from motion (SFM) with the camera moving but with a fine difference. In SFM, the two views are taken at separate times with the same camera whilst in

stereovision, in theory, the two views could be simultaneously taken. This difference between image is important for reconstruct the 3rd dimensional information.

The multi-baseline stereo vision [15] can resolve the above baseline-matching dilemma. One could view this approach as a bridging of multiple views. Those views closer to each other allow easier matching of features. Hence by bridging from view to view one could find feature matches of views far from each other. This enables accurate computations of surfaces.

The 3D face reconstruction is an ill-posed problem by itself in computer vision. Due to this nature, any reconstruction method should use as much information as possible. Thus the advantage of Multi-view reconstruction is the potential for dense surface recovery. From the above discussion, there are still some points that need investigation on the multi-view based 3D reconstruction, such as how to establish the relationship of multiple views and how to recover the target object surface when there are perspective distortions among the views.

## 1.2 Objective

The main objective of this thesis is to research and develop new ideas for multi-view based 3D reconstruction, and to improve the computational efficiency and performance. We also try to understand and explain the 3D reconstruction problem in a novel way. Normally, to achieve the 3D shape from 2D images, researchers need to fully calibrate the cameras to obtain both internal and external parameters, then establish the correspondences between images by finding feature or dense matching, then triangulate the 3D shape in 3D space.

We will build a system that only needs to know the cameras' intrinsic parameters, capture the object simultaneously, then obtain the 3D shape of the object by de-

signed algorithms. To establish the relationship between images, we take advantage of point features like corners, and designed improved algorithm to match them. Many corner detectors have been proposed based on different principles. Also, different methods have been proposed for matching the point features in the past three decades. The review of corner detectors and matching methods will be given in chapter 3. They are quite successful by themselves, e.g. designing corner detector by assuming a perfect matching method, or designing matching methods by assuming a perfect corner detector. However, because the feature matching methods are building the correspondences based on the corner detector, the accuracy, stability and other attributes of the corner detector will greatly affect the final feature correspondence. Consquently, the external camera parameters will also be affected.

Our first objective is to consider the corner detection and corner matching simultaneously, and to build up a wrapper scheme to improve the final feature matching accuracy. Our target is to establish more accurate matches between images, and that will pave the way for an accurate extrinsic camera parameters recovery. We will also improve the computational efficiency in the process.

With a camera system calibrated, the next step is to recover the 3D shape with the fully calibrated camera. Most methods obtain the 3D shape by finding dense matching on images, and then triangulate the 3D shape. The projection from 3D world to 2D image is straightforward, but, the inverse process is complicated. The image of the same object can be distorted by pose variation and illumination variation. The result is that, in practice, a dense matching is difficult to acquire, especially when the pose variation is large.

Our second objective is to design an algorithm to recover the 3D object shape. We will take the pose variation into account and avoid establishing dense matching that is only based on 2D images. We formulate our reconstruction method as an information exploring problem, and try to take into account as much information

as we can. To accomplish this objective, we design our algorithm to operate in 3D space. And our aim is to find the best solution that is consistent with the input images.

## 1.3 Main Contributions

As we discussed in the objective section, we mainly focus on two steps of 3D reconstruction, firstly, how to obtain the external parameters of cameras, and furthermore, how to get reliable fundamental matrix; secondly, how to recover the 3rd dimensional surface of the targeted object. We formulate the 3D reconstruction as an information exploration problem. Therefore, our main contributions can be summarized as followed.

1. **An improved SUSAN corner detector**

   In the literature review later, we can find that quite a number of corner detectors have been proposed and compared with other detectors. And also, many are proposed based on the assumption that corners have been detected accurately and robustly. This assumption is somewhat ideal because current corner detectors can hardly satisfy all possible assumptions. Based on different principles, some corner detectors have the advantage on stability and robustness, and some are more advanced on the localization accuracy. Unfortunately, there is no ideal corner detector that can perform perfectly on every aspect such as robustness and accuracy, especially the ones that more critical for the later matching algorithm. Our first contribution is to specially design a wrapper corner detector concatenated with the corner matching method. Our corner detector identify the qualified point by analyzing the local image. The proposed corner detector will be shown to be an improvement over the well known SUSAN corner detector. As a minimum 8 pairs of matches are enough

to recover the Fundamental matrix between two images, our contribution is attributable to find out the true corners that are stable and with good localization accuracy. Our target is not a power general corner detector, instead, we design our corner detector to best cooperate with the corner matching method by giving a small amount of correct corners.

2. **A robust feature matching method based on a new core function**

   After the corners are detected in images, a feature matching algorithm can be used to find the corresponding corners between or across images. Let us take two images as example. The popular idea is to evaluate similarity between the local image areas that are surrounding the corners in each image. If the similarity is higher than some threshold, then the two corners are accepted as a pair of candidate of matching points. However, a straightforward application of this method will result in many false matches. As a result, a number of methods have been proposed to robustly establish the correspondence. In the work of Zhang [16], a robust matching based on the definition of matching strength was proposed that used the sum of the matching strength as an energy function. The robust matches are established when the energy function is minimized. The strength of match is defined based on affine assumption between images. As Zhang's method did not take into account the possibility that the two images are slightly rotated from each other, our contribution is made to re-define the strength of match to incorporate rotation transformation. By this way, we tried to get stronger invariance by a group of corners. A method based on median filter, which in turn applied to the local transformations, is designed to evaluate the weight of a pair of corners from two images.

3. **An integrated matching scheme that consider the corner detection and corner matching results simultaneously**

Here our contribution is to propose an integrated matching scheme. The idea is inspired by the concatenated steps of 3D reconstruction. The former step's output will have a great effect on the input of the later step. For example, the accuracy and stability of detected corners will directly influence the final calculated extrinsic camera parameters. If the stability is perfect, but the accuracy of localization has more than two pixel distance error, the epipolar geometry found based on that will have significant errors. With this observation, we start to consider the corner detection and corner matching as a whole. And because it requires only a minimum number of 7 pairs correct and accurate corresponding corners to recover the epipolar geometry, our idea is use only the most accurate and stable corners, for our corner matching algorithm to obtain the best group of corner correspondence. By this way, we minimize the data size of corner correspondence, and also minimize the percentage of noise of the data size. Both improvements lead to the simplification of the non-linear optimization applied when searching for the best epipolar geometry.

4. **A reconstruction method based on 3D adaptive correlation window with calibrated cameras**

When the 3D world is projected to 2D images, the images of the same object will be different because of pose variation, occlusion, illumination variation, perspective distortion and sensor difference. By using simultaneous multi-camera capture, we actually minimize the influence by the illumination variations. However, it will still not be a trivial problem to densely match the images. Our idea is inspired by the projection process that relating the 3D world space to 2D image space. We formulate 3D reconstruction as an information exploration problem, and start from the strictest assumption to obtain more information. After that, we step by step relax our assumption and find new constraints until the 3D shape of the target object is recovered. Our proposed 3D reconstruction method operates in both 3D and 2D space

to find the dense matching instead of only in 2D image space in literature. A 3D adaptive correlation window is defined and proved to find the true 3D surface together with a dense matching. The occlusion is accounted by a layered depth image. An iteration method is then applied to refine the object surface until it converge.

## 1.4 Thesis Organization

Chapter 2 introduces the important concept of projective geometry that is related to the rest of thesis. Firstly the projective geometry and transformations are introduced as the natural base of 3D reconstruction. After that we describe the pin-hole camera model and single view geometry. Finally, we introduce some knowledge on two view and multi view geometry which relates images of the same scene.

Chapter 3 give a detailed literature review of our work. We first review the corner detector literature and identify their advantages and shortcomings. Then we give an introduction for the matching methods that can find correspondence between two images to find the epipolar geometry. After that, a detailed review will be given on recovering the 3D shape from a calibrated camera system.

Chapter 4 proposes our improved SUSAN corner detector. We also demonstrate the influence of the corner detector to the final matching results.

Chapter 5 proposes a feature matching scheme with new definition of matching strength. The final results of feature correspondence are also showed in this chapter, and compared with the existing methods.

Chapter 6 proposes a novel 3D reconstruction method. A reconstruction scheme is built up which incorporates 3D adaptive window and layered depth image.

Chapter 7 concludes the thesis and contains discussions on the future work.

# Chapter 2

# Camera Projective Geometry

This chapter describes the theoretical foundation of our research on 3D reconstruction. We first introduce the projective geometry as the theoretical basis of 3D reconstruction. The projective space is a superset of the classic Euclidean space, and its algebraic expression of projection from 3D world to 2D image is to linearize the expression in the otherwise nonlinear expression in Euclidean space. We also introduce the various levels of stratification that the 3D space can be described by projective, affine, similarity and Euclidean geometry, each being a subset of the former one. Then we will review the camera model and single view geometry. Different camera models are introduced and the associated projective matrix are compared. Also, a special case that maps a 3D plane to 2D image is studied. After that, the two-view geometry is introduced. The epipolar geometry and fundamental matrix, which describing the relationship between two images of the same 3D object, are studied in detail. We also extend the plane object case from one view to two views, and the relationship between the two images of the same world plane homography is investigated. This homographic model will be used in our work.

The organization of this chapter is as follows. Section 2.1 introduces the basic projective geometry. Section 2.2 describes the camera model and projective matrix.

In section 2.3, two-views geometry is introduced. Lastly in section 2.4, a discussion is given.

## 2.1 3D Projective Geometry and The Stratification

One important aspect of projective geometry is the study of invariance under projective transformation. It is the mathematical framework to view computer vision in general, especially when relating images. The camera used in computer vision field captures a perspective image of the real world. This perspective image could be described by the classic Euclidean geometry in a fairly complicated manner. However, it will be really hard to formulate in a neat and convenient manner if two or more images are to be related. This relationship could be expressed by a combination of two perspective projection, which will be a projective transformation with closed form. There are two levels in between, affine and similarity, from projective geometry to Euclidean geometry. Each is a subset of the former one in the order of projective, namely affine, similarity and Euclidean. And each strata is related to a group of transformations and invariance. At the same time, although a more flexible transformation group can facilitate the formulating of relationship between images, it also means less invariants and fewer preserved measures that are disadvantageous to real world descriptions. Both the projective geometry concepts and the relationship between stratums will be studied in detail with reference to [12] [17] [13] [18] [19].

### 2.1.1 The 3D Projective Geometry

#### The Homogeneous Coordinates

The use of homogeneous coordinates allow the projective transformation to be computed in terms of matrix calculations. An $n$ dimensional projective point can be denoted as $[x_1, ..., x_i, ..., x_{n+1}]^T$ where not all $X_i$ are equal to zero. In homogeneous coordinates, two proportional sets of coordinates denote the same point of projective space:

$$[x_1, ..., x_i, ..., x_{n+1}]^T \sim \lambda[x_1, ..., x_i, ..., x_{n+1}]^T \qquad (\lambda \neq 0, \lambda \in IR) \qquad (2.1)$$

### The General Projective Space

A general n-dimensional projective space, $IP^n$, is defined in homogeneous coordinates as any (n+1)-vector, that has

$$[x_1, ..., x_i, ..., x_{n+1}]^T \sim \lambda[x_1, ..., x_i, ..., x_{n+1}]^T \qquad (\lambda \neq 0, \lambda \in IR) \qquad (2.2)$$

where at least one of $x_i$ is different from zero. When $x_{n+1} \neq 0$, the vector $[x_1/x_{n+1}, ..., x_n/x_{n+1}]^T$ is just the Euclidean space $IR^n$, and the points that $x_{n+1} = 0$ are denoted as **points at infinity**. The coordinate $x_{n+1}$ is usually normalised to 1 for convenience because the scalar does not matter.

### The 3D Projective Space and Transformation

The 3D projective space, $IP^3$, can be defined as any non-zero 4-vector $M = [X, Y, Z, W]^T$ that satisfy $M \sim \lambda M$ for any non-zero $\lambda$. Point $M$ is corresponding to the point $[X/W, Y/W, Z/W]$ in $IR^3$ when $W \neq 0$. The points of $W = 0$ form a plane known as the **plane at infinity**. The plane at infinity is actually exactly the same as any other plane in projective space, it is identified because of the stratification concept and the speciality in 3D reconstruction.

The corresponding projective transformation is a linear transformation on homogeneous 4-vectors that maps one projective point to another: $X' = HX$. It is

represented by $4 \times 4$ non-singular matrix $T_P$:

$$T_P \sim \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & 1 \end{bmatrix} \tag{2.3}$$

$T_P$ has 15 degree of freedom because it is defined up to a non-zero scalar.

### 2.1.2 The Stratifications of 3D Spaces

The 3D world space is not only described by Euclidean structure, but also projective, affine and similarity ones. These spaces or strata are overlaid on and related to each other.

#### The Affine Stratum

The 3D affine space is a subset of 3D projective space with identifying the **plane at infinity**. The associated affine transformation will not change the **plane at infinity** and can be denoted as $T_A$:

$$T_A \sim \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

$T_A$ has 12 degree of freedom whereas the **plane at infinity** account for 3 degree of freedom. A note here is that the **plane at infinity** can be any plane in projective space because all projective planes are exactly same.

#### The Similarity Stratum

The 3D similarity space is a subset of 3D affine space, hence the subset of 3D projective space, with identifying the absolute conic:

$$\left.\begin{matrix} x_1^2 + x_2^2 + x_3^2 \\ x_4 \end{matrix}\right\} = 0 \tag{2.5}$$

which contains 5 degree of freedom. The 3D similarity transformation could be denoted as $T_S$:

$$T_S \sim \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ s_{31} & s_{32} & s_{33} & s_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = s \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} = s \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \tag{2.6}$$

And $T_S$ has 7 degree of freedom: one for scaling $s$, three for rotation $R$ and three for translation $t$.

**The Euclidean Stratum**

The 3D Euclidean space is a subset of 3D similarity space with fixing the scale. The associated Euclidean transformation is represented as $T_E$:

$$T_E \sim \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \tag{2.7}$$

$T_E$ has 6 degree of freedom with regarding to three for $R$, and three for $t$. However, it will be impossible to identify the scale without an extra real world information, without which the 3D reconstruction will then be recovering the object structure in similarity space.

The relationship of basic transformations and invariants corresponding to each stratum is illustrated in Fig.2.1 and Fig. 2.2:

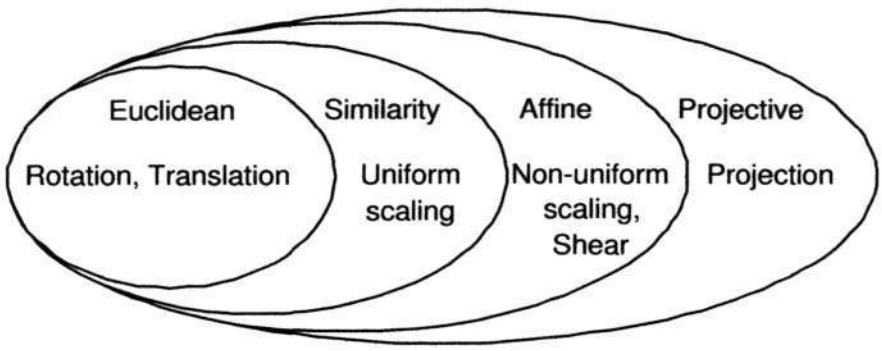



Figure 2.1: The Hierarchy of Transformations from Euclidean Geometry to Projective Geometry

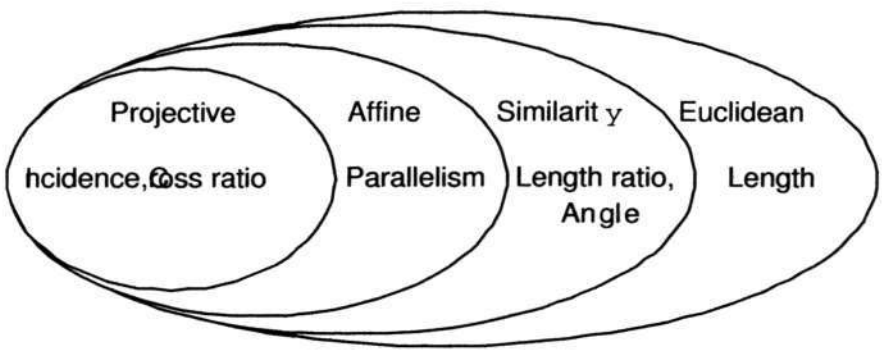



Figure 2.2: The Hierarchy of Geometric Invariants from Projective Geometry to Euclidean Geometry

## 2.2 Camera Model and Single View Geometry

The camera model of interest in this section is central projection. There are two major class of camera models: the models with a finite center and the ones with center 'at infinity'. In computer vision, a pin-hole camera is the usual model adopted for analysis. Here the pin-hole camera model has a finite center and points can be

modelled as a linear projection from 3D space into 2D image. Then, the affine camera model, as one type of camera model with projection center 'at infinity' due to its parallel projection,is illustrated.

### 2.2.1 Finite Projective Camera

We define three coordinate systems: the world coordinate system, the camera coordinate system and the image coordinate system as shown in Fig. 2.3. The origin of

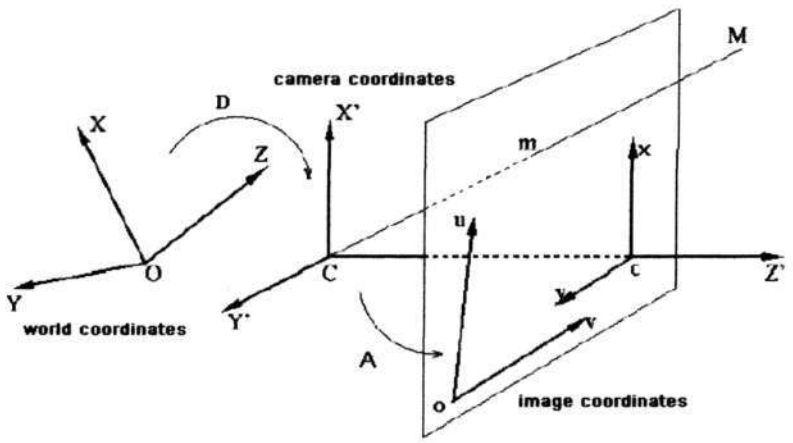



Figure 2.3: Pin-hole Camera Model [12]

camera coordinates system coincides with the centre of projection, or *camera center*. The line that is perpendicular to the image plane and passing through camera center is the *principal line*, and the intersection point between principal line and image plane is the *principal point*. Also, the plane that is parallel to image plane and contains camera center is denoted as principal plane.

Given a 3D space point $\mathbf{M} = [X, Y, Z, 1]^T$ and its image $\mathbf{m} = [u, v, 1]^T$, the projection can be represented as:

$$\mathbf{m} = \mathbf{PM} \tag{2.8}$$

where $\mathbf{P}$ is a $3 \times 4$ perspective projection matrix, which can be further decomposed as:

$$\mathbf{P} = \mathbf{KQD} \tag{2.9}$$

where

$$\mathbf{K} = \begin{bmatrix} f_u & f_u \cot\theta & u_0 \\ 0 & f_v / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

are the intrinsic parameters of the camera and it maps the camera coordinates to the image coordinates. The matrix

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.11}$$

is the projection matrix and

$$\mathbf{D} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0^T & 1 \end{bmatrix} \tag{2.12}$$

is 3D rigid transformation which includes rotation $\mathbf{R}$ and translation $\mathbf{t}$ to align the world coordinates to the camera coordinates.

The general form of matrix $\mathbf{K}$ contains five independent variables that describe the intrinsic parameters of a camera. $\alpha_u$, $\alpha_v$ represent for the focal length in horizontal and vertical pixels, $\gamma$ gives the skew or non-orthogonality between the axes and usually assume to be zero, and $u_0$, $v_0$ represent the coordinates of the principal point. Matrix $\mathbf{D}$ is a $4 \times 4$ extrinsic matrix which has six degrees of freedom: three for rotation and three for translation. So a finite projective camera has 11 degrees of freedom.

### 2.2.2 Infinite Affine Camera

When the camera center is lying on the plane at infinity, the top left $3 \times 3$ block of $\mathbf{P}$ will be singular. This characteristic will change the camera model performance greatly. Furthermore, when the principal plane is coincident with the plane at infinity, it becomes an affine camera. The model of the camera at infinity will not be studied in this thesis.

The affine camera is one that has a camera matrix $\mathbf{P}$ in which the last row of it is of the form $(0, 0, 0, 1)$. It is called an affine camera because points at infinity are mapped to points at infinity. Compared with equation 2.9, the camera matrix $\mathbf{P}_A$ can be decomposed as:

$$\mathbf{P}_A = \begin{bmatrix} \mathbf{K}_{2\times 2} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.13}$$

Affine cameras can be categorized into orthographic projection, weak perspective projection and para-perspective projection. Figure 2.4 is a profile of them compared with perspective projection.

The orthographic projection, shown as $X_{orth}$ in Figure 2.4, maps a 3-space point $(X, Y, Z, 1)$ to the image point $(X, Y, 1)$. A general orthographic projection can be represented by a matrix of the form:

$$\mathbf{P}_{orth} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.14}$$

An orthographic camera has five degrees of freedom: three for rotation $\mathbf{R}$ and two for translation $\mathbf{t}$, noting that information along the of $\mathbf{Z}$ axis is dropped.

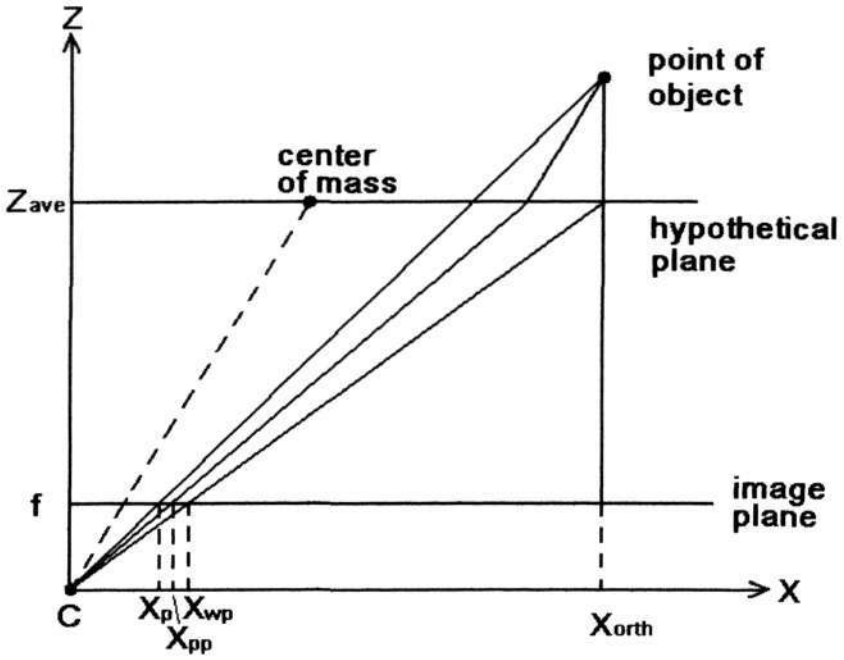



Figure 2.4: Affine Camera Model Categories

The weak perspective projection [20], shown as $X_{wp}$ in Figure 2.4 , is an orthographic projection followed by isotropic scaling. Thus, it is also known as scaled orthographic projection and its matrix may be written as:

$$\mathbf{P}_{weak} = \begin{bmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.15}$$

It has six degree of freedom, one more (the isotropic factor $k$) than orthographic projection. In this projection, the object points are orthographically projected onto a hypothetical image plane parallel to the actual image plane but passing through the object's center of mass $C_m$. Then this image is projected onto the image plane using perspective projection. The weak perspective projection can be used when the object remains centered in the image, and when the distance to the object is

large relative to the size of the object.

The para-perspective projection [21] [22], shown as $X_{pp}$ in Figure 2.4, is a closer approximation to perspective projection than weak perspective projection. It is of the form:

$$\mathbf{P}_{para} = \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.16}$$

It has seven degree of freedom, two more $(\alpha_x, \alpha_y)$ than orthographic projection. In para-perspective projection, firstly an object point is projected along the direction of the line connecting the focal point of the camera to the object's center of mass, onto a hypothetical image plane parallel to the real image plane and passing through the object's center of mass. Then, the point is projected onto the real image plane using perspective projection.

The camera model described above can be seen to be an affine camera satisfying additional constraints, thus the affine camera is an abstraction of this hierarchy. As mentioned in equation 2.13, the affine camera has eight degree of freedom, and may be thought of as the parallel projection version of the perspective projective camera.

### 2.2.3 The Plane Object

A special case is when the object is planar or a plane. For a point $\mathbf{X}$ on a world plane, its image coordinates could be easily obtained by $\mathbf{x} = P\mathbf{X}$, where P is the projection matrix. However, because the world coordinate system is in practice arbitrary, we can select it to make one axis perpendicular to the plane object and the origin is on the plane. Then $P$ can be computed as follows. Suppose the plane

is overlaid with the xz-plane of the world coordinates, then:

$$\mathbf{x} = P\mathbf{X} = [p_1, p_2, p_3, p_4] \begin{bmatrix} X \\ 0 \\ Z \\ 1 \end{bmatrix} = [p_1, p_3, p_4] \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} \tag{2.17}$$

Eq. 2.17 is then mapping a world plane to the image plane, and the transformation is a plane to plane projective transformation, or called *planar homography.*

If the camera is an affine one, Eq. 2.17 could be revised as:

$$\mathbf{x} = P\mathbf{X} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ 0 \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{13} & p_{14} \\ p_{21} & p_{23} & p_{24} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} \tag{2.18}$$

Eq. 2.18 means the mapping between object plane and image plane is a general plane to plane affine transformation.

## 2.3 Geometry of Two views

One dimension is lost when capturing a 3D world to 2D image. The task of 3D reconstruction is to recover the lost dimension information from 2D images. However, it will be impossible to do it by only one image without any other a-priori knowledge. Normally, two or more images are used for 3D reconstruction by finding their relationship. The two-views geometry is the basis of these methods. The relationship between two un-calibrated images is the epipolar geometry and further, described by fundamental matrix.

### 2.3.1 Epipolar Geometry

The epipolar geometry is an intrinsic projective geometry between two cameras. It is independent of scene structure, and only depends on the intrinsic parameters and relative pose of the cameras.

Suppose a point $\mathbf{M}$ in 3-space has its projections, $\mathbf{m}$ and $\mathbf{m}'$ in two images respectively. The images can be captured by two cameras at the same time instance or by one camera at different viewpoints and time instance under some condition. As is shown in Fig. 2.5.

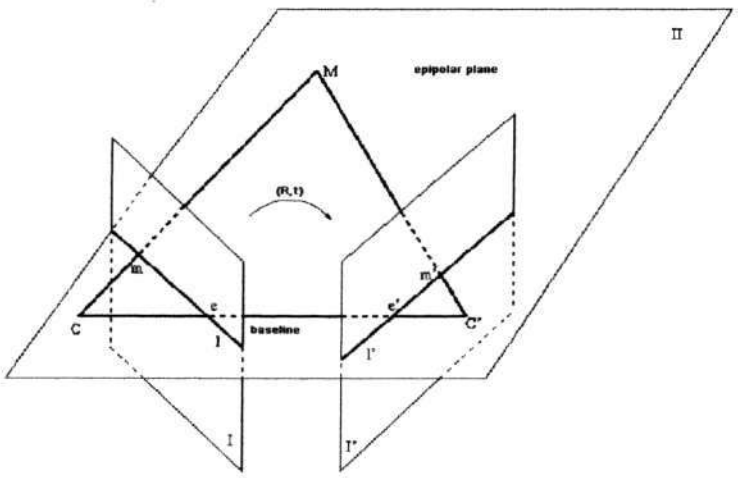



Figure 2.5: The epipolar geometry [12]

$C$, $C'$ are the optical centers of cameras, and $\pi_i$, $\pi_i'$ are the respective image plane respectively. For the first image $\mathbf{m}$ of $\mathbf{M}$, its corresponding point is $\mathbf{m}'$ in the second image. For simplicity, we assume the world coordinates is aligned with the first camera coordinates so that $\mathbf{M}$ is expressed in the camera coordinate system of the first camera. Then for each camera, we have:

$$s_1\mathbf{m} = \mathbf{K}_1 \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{M} \tag{2.19}$$

$$s_2\mathbf{m}' = \mathbf{K}_2 \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \mathbf{M} \tag{2.20}$$

where $\mathbf{K}_1$ and $\mathbf{K}_2$ are the intrinsic matrices of the two cameras respectively. $s_1$, $s_2$ are arbitrary non-zero scale factors, $\mathbf{R}$ and $\mathbf{t}$ are the rotation and translation from the first camera coordinates to the second one. By eliminating $\mathbf{M}$, $s_1$ and $s_2$, we can obtain the fundamental equation:

$$\mathbf{m}'^T \mathbf{K}_2^{-T} \begin{bmatrix} \mathbf{t} \end{bmatrix}_x \mathbf{R}\mathbf{K}_1^{-1}\mathbf{m} = 0 \tag{2.21}$$

and define

$$\mathbf{F} = \mathbf{K}_2^{-T} \begin{bmatrix} \mathbf{t} \end{bmatrix}_x \mathbf{R}\mathbf{K}_1^{-1} \tag{2.22}$$

which is known as fundamental matrix, and $[\mathbf{t}]_x$ is skew-symmetric matrix:

$$\begin{bmatrix} \mathbf{t} \end{bmatrix}_x = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & t_x \\ t_y & -t_x & 0 \end{bmatrix} \tag{2.23}$$

Let us refer to Fig. 2.5 again. The line passing through the optical centers of two cameras is called the baseline. The epipoles $e$ and $e'$ are the points of intersection of baseline and image planes repsectively. That is, the epipole is the image in one image plane of the optical center of the other camera. An epipolar plane is the plane indicated by baseline and the 3-space point $\mathbf{M}$. Obviously, there is a pencil of epipolar planes. An epipolar line $l$(or $l'$) is the intersection of an epipolar plane with the image plane $\pi_i$(or $\pi_i'$), and all epipolar lines intersect at epipole in each image plane. Mathematically, we have:

$$l' = \mathbf{F}\mathbf{m} \tag{2.24}$$

or

$$l = \mathbf{F}^T \mathbf{m}' \tag{2.25}$$

and for the epipoles

$$\mathbf{F}\mathbf{e} = \mathbf{F}^T \mathbf{e}' = 0 \tag{2.26}$$

Then the epipolar constraint can be summarized as, for a given point in one image, its corresponding point in the other image must lie on the epipolar line of this point. One important property is the rank of $\mathbf{F}$ is two.

If the cameras' intrinsic parameters $\mathbf{A}_1$ and $\mathbf{A}_2$ are known, then equation 2.1.8 is equivalent to

$$(\mathbf{K}_2^{-1}\mathbf{m}')^T \left[\mathbf{t}\right]_x \mathbf{R}(\mathbf{K}_1^{-1}\mathbf{m}) = 0 \tag{2.27}$$

which is the original equation derived by Longuet-Higgins [5] involving the essential matrix:

$$\mathbf{E} = \left[\mathbf{t}\right]_x \mathbf{R} \tag{2.28}$$

Compared with the fundamental matrix, the essential matrix has fewer degrees of freedom and has additional properties. The most important one is that a $3 \times 3$ matrix is an essential matrix if and only if two of its singular values are equal, and the third is zero. This property enables a way of decomposing $\mathbf{E}$ into $\mathbf{t}$ and $\mathbf{R}$ uniquely with the pure data or robustly in the least square sense with noisy data. For a proof, see [12].

### 2.3.2 The Plane Object in Two Images

Continuing from section 2.2.3, the discussion here will focus on the relationship between two images of the same world plane object.

Suppose $\mathbf{x}$ and $\mathbf{x}$' are the two image points of the same space point $\mathbf{X}$, P and P' are

the corresponding projective matrix respectively. There will be:

$$\mathbf{x} = P\mathbf{X} = [p_1, p_2, p_3, p_4] \begin{bmatrix} X \\ 0 \\ Z \\ 1 \end{bmatrix} = [p_1, p_3, p_4] \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} = P_{3\times3} \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} \tag{2.29}$$

$$\mathbf{x}' = P'\mathbf{X} = [p'_1, p'_2, p'_3, p'_4] \begin{bmatrix} X \\ 0 \\ Z \\ 1 \end{bmatrix} = [p'_1, p'_3, p'_4] \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} = P'_{3\times3} \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} \tag{2.30}$$

So that:

$$\mathbf{x}' = P'_{3\times3} P^{-1}_{3\times3} \mathbf{x} \tag{2.31}$$

where $P'_{3\times3} P^{-1}_{3\times3}$ is planar projective transformation from one image to the other. It has 8 degree of freedom.

Similarly, if the two cameras are affine, then:

$$\mathbf{x} = P_A\mathbf{X} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ 0 \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{13} & a_{14} \\ a_{21} & a_{23} & a_{24} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} = P_{A3\times3} \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} \tag{2.32}$$

$$\mathbf{x} = P'_A\mathbf{X} = \begin{bmatrix} a'_{11} & a'_{12} & a'_{13} & a'_{14} \\ a'_{21} & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ 0 \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} a'_{11} & a'_{13} & a'_{14} \\ a'_{21} & a'_{23} & a'_{24} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} = P'_{A3\times3} \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} \tag{2.33}$$

Also the relationship between **x** and **x**' could be derived as:

$$\mathbf{x}' = P'_{A3\times3}P^{-1}_{A3\times3}\mathbf{x} \tag{2.34}$$

And $P'_{A3\times3}P^{-1}_{A3\times3}$ is 2D planar affine transformation with 6 degree of freedom. It is important because the affine camera model is also a good assumption with 2 less unknown degree of freedom.

## 2.4 Discussion

This chapter introduces the fundamental geometric concepts on which our research on 3D reconstruction will be based. Firstly, the projective geometry is described as the basis of computer vision. The projective space can be stratified into four overlaid space: projective, affine, similarity and Euclidean. Each of them is the subset of the former one. Also, the corresponding transformations and the invariants to these transformations are also illustrated. The relationship between strata is also explored, which implicitly give the routine of 3D reconstruction. Secondly, the camera models are introduced as finite projective camera and affine camera. The corresponding projective matrix that maps 3D point to 2D image is studied. Furthermore, the planar object, as a special object, is examined both on projective camera and affine camera. The importance of planar object is that a great amount of real world object surface can be approximated by a patchwork of small planes. This assumption, together with less degrees of freedom of the affine camera from projective camera, can greatly facilitate the 3D reconstruction from images. We will be using this homography for our perspective stereo correspondence. Finally, the two-views geometry that describes the relationship between two images of the same object is introduced. The epipolar geometry and the inherent fundamental matrix are illustrated. Then, the relationship between two views that seeing a planar object

is introduced based on the discussion of single view situation. The transformation between images of planar object has less unknowns based on affine camera than the projective camera. That means less information will be needed to solve the transformation from image to image. And the affine and projective camera models will be the basis of our work in this thesis.

# Chapter 3

# A Literature Survey on 3D Reconstruction

The task of 3D reconstruction in computer vision is to recover the third dimension that are lost in camera image by perspective projection. The 3D recovery methods come under the category known as shape-from-X, where X could be representing stereo, multiple views, structure from motion, shading and so on. However, it is never an easy problem to solve although capturing image is very convenient nowadays. Normally, the relationship between two or more images is established, and the 3D structure of the target object is recovered based on this relationship with the camera information. Numerous ideas and algorithms have been proposed for every steps and aspects of 3D reconstruction. However, there is still long way to go because of the wide spectrum and complexity of the problem. In this chapter, we will give a detailed review on the general framework of 3D reconstruction consisting of: corner detection, feature matching and multi-view reconstruction with calibrated cameras. These three processes are important but do not represent fully the 3D reconstruction.

## 3.1 The General Framework for 3D Reconstruction from 2D Images

Different methodologies have been proposed for 3D reconstruction such as stereo vision [23] [24] [25] [26] [27], multiple views [15] [28] [29] [30], structure-from-motion [31] [32] [33] [11]. Various algorithms consider for the different inputs and specific situations. However, they are all concerned with establishing the relationship of input images/frames and the recovery of 3D shapes by exploiting projective geometry properties.

In stereo vision, it is desirable to find dense matches between images [34] [35] [36] [37], and then calculate the disparity based on the correspondence. Because the disparity is directly related to the depth information, most papers give disparity as its final results. However, direct relationship does not mean it is straightforward. In practice, it may not be realistic to obtain the arbitrary depth information based on disparity because it generally is pixel-based and with large error. On the other hand, the length of the base-line in stereo vision will have a great effect on the final 3D results. If the base-line is large, the matching between images will be hard to establish because of the large pose variation, or the high percentage of false match. If the base line is small, the correspondence will be easier to find because of the small pose variation. However, the depth error will be large.

Multi-baseline stereo [15], and the generalized multi-view methods, can greatly resolve the baseline problem in binocular stereo vision. The multi-view methods can recover the 3D shape by exploring the relationship of input images. However, because the pose variation from one camera to another may be very large, more work may be needed to account of this problem.

The structure-from-motion methods normally use a moving video camera to capture different views of the object. It can be formulated as a problem to find the geometry

of "N points in M views". Features are tracked throughout the frames, and are used to establish the relationship between them. Then, the proposed algorithm is to find out the structure in 3D space by knowing which point corresponds to which in different views.

Based on the above brief introduction on the different methodologies, we observe that they share some common properties. They find similar features in each image and find their homologous matches. Then based on a subset of reliable of the established correspondence, the algorithms find the geometry between or across the images. And finally, they reconstruct the 3D structure from the estimted camera geometries and correspondence. So we can summarize this general 3D reconstruction framework as in Fig. 3.1.

In this general framework, the 3D reconstruction is achieved by a sequence of processing. Each step will have great effect on the final reconstruction results, as it will influence the results of the later steps. Thus, we proposed to solve the 3D reconstruction problem in an integrated fashion for potentially improved results. But this entails greater complexities and may make the entire effort intractable. In our proposal, we try to compromise. The feature (corners) extraction stage will be made more robust and the corners with high reliabilities will be used as the kernels to estimated the camera geometry. And an accurate camera geometry can lead to better matching results and facilitate the final 3D reconstruction.

## 3.2 Corner Detection

Corners used in many applications of computer vision, such as image matching, registration, tracking, detection and recognition. They are considered the most stable, invariant points that can be reliably used as reference or landmark points for the latter dense matching. At present, there is no universal definition of a corner

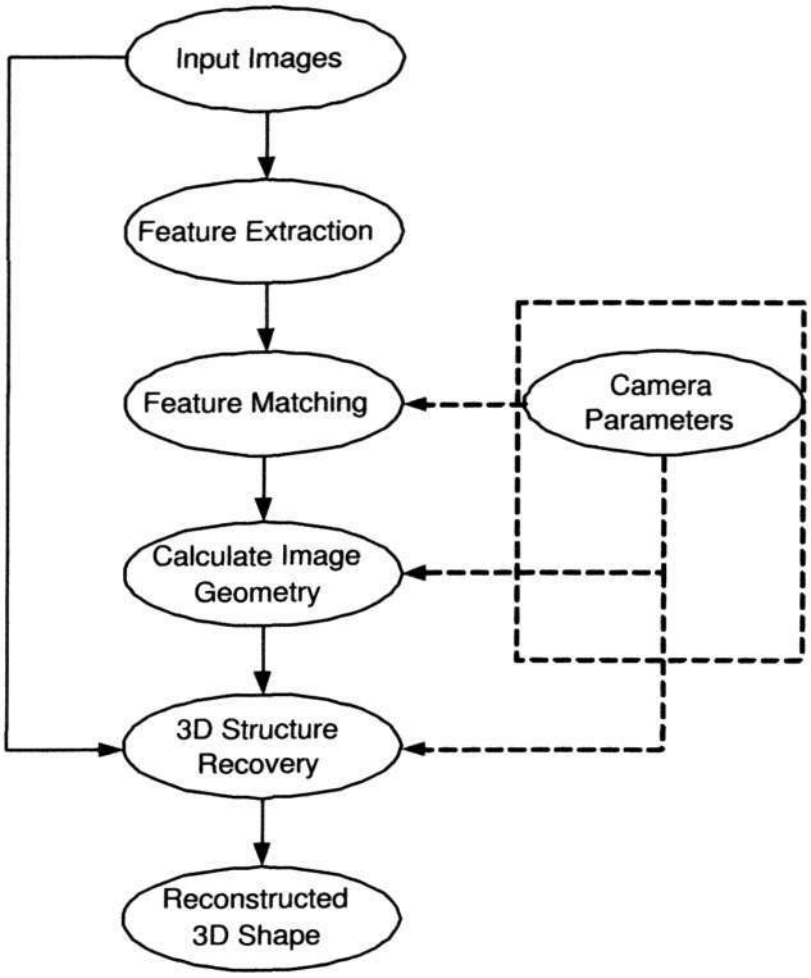



Figure 3.1: The General Framework of 3D Reconstruction from Images

point. And hence no single corner detection algorithm can detect all forms of corners. Consequently, each corner detector is designed to detect what it defines as a corner. There are two main issues in evaluating the performance of a corner detector. Is it a valid corner? and how accurate was a valid corner detected. Some define a corner as a prominent point that is different from the points in its neighborhood.

Corner detector algorithms can be categorized into the following four groups: 1. Auto-correlation based corner detector. 2. Edge-based corner detector. 3. Topology-based corner detector. 4. Structure-based corner detector.

### 3.2.1 Auto-correlation Based Corner Detector

These methods define a corner to be a prominent point, a point with low similarity with all of its neighbors. For a given image point, the intensity correlation or difference measurement between the local window centered on it and the slightly shifted one is calculated. The point is declared as a corner when the difference under all directional shift is large. The shift direction will vary according to different methods.

**Moravec Corner Detector**

It is one of the earliest corner detection algorithm [38] [39] that is defined based on the concept of "point of interest". The intensity variation is calculated by SSD (Sum of Squares of Differences) between corresponding pixels in the overlapping windows before and after shifting to one of eight directions about the current pixel. The local window is shifted in four directions as horizontal, vertical and two diagonals. Then the minimum of the four SSD is taken as the corner response. Then the center point with local maxima of the corner response is reported as a corner.

Mathematically, the Moravec corner detector can be described as:

$$C_{Moravec} = \min \sum_{(u,v)\in W_{local}} [I(u+x, v+y) - I(u,v)]^2 \quad x \in [-1,1], y \in [-1,1], x^2 + y^2 \neq 0 \tag{3.1}$$

where $W_{local}$ is a small image patch about image point (u, v).

The $C_{Moravec}$ of all pixels is compared with a pre-set threshold, the non-maximal suppression is applied to find the local maxima.

The computational complexity of Moravec corner detector is relatively low. However, it is a simple corner detector with some drawbacks. Its response is actually not isotropic because there are only four shift axis of the local window. The Moravec

corner detector will not consistently detect corners if the image is rotated because its anisotropic character. This method is also sensitive to noise as it easily detect false corners along strong edge and isolated points.

**The Plessey Corner Detector**

To address the limitation of the Moravec corner detector, Harris and Stephen [40] designed the Plessey corner detector. Applying Taylor expansion, we have:

$$I(u+x, v+y) = I(u,v) + xI_u(u,v) + yI_v(u,v) + O^2(x^2+y^2) \tag{3.2}$$

So the SSD in Eq. 3.1 becomes:

$$\begin{aligned} \triangle_{SSD} &= \sum_{(u,v)\in W_{local}} [I(u+x, v+y) - I(u,v)]^2 && (3.3)\\ &= \sum_{(u,v)\in W_{local}} [xI_u(u,v) + yI_v(u,v) + O^2(x^2+y^2)]^2 && (3.4)\\ &\approx \sum_{(u,v)\in W_{local}} [x^2 I_u^2(u,v) + 2xyI_u(u,v)I_v(u,v) + y^2 I_v(u,v)^2] && (3.5) \end{aligned}$$

To account for the noisy response in Movarec corner detector, a Gaussian smoothing is added to the local window in Eq. 3.5 as:

$$\begin{aligned} \triangle_{SSD} &\approx \sum_{(u,v)\in W_{local}} g(\sigma,u,v)[x^2 I_u^2(u,v) + 2xyI_u(u,v)I_v(u,v) + y^2 I_v(u,v)^2] && (3.6)\\ &= \begin{bmatrix} x & y \end{bmatrix} \mathbf{M}_{plessey} \begin{bmatrix} x \\ y \end{bmatrix} && (3.7) \end{aligned}$$

where

$$\begin{aligned}
\mathbf{M}_{plessey} &= \begin{bmatrix} \widehat{I_u^2} & \widehat{I_u I_v} \\ \widehat{I_u I_v} & \widehat{I_v^2} \end{bmatrix} \\
\widehat{I_u^2} &= G(\sigma) \otimes I_u(u,v) \\
\widehat{I_v^2} &= G(\sigma) \otimes I_v(u,v) \\
\widehat{I_u I_v} &= G(\sigma) \otimes [I_u(u,v) I_v(u,v)]
\end{aligned}$$

The eigenvalues $\lambda_{e1}$ and $\lambda_{e2}$ of $\mathbf{M}_{plessey}$ are proportional to the two main local curvatures. So the corner can be defined based on the values of $\lambda_{e1}$ and $\lambda_{e2}$:

- $\lambda_{e1} \approx 0, \lambda_{e2} \approx 0$. It means both curvatures are small so that the intensity within the local image region is flat. That indicates no corner.

- $\lambda_{e1} >> 0, \lambda_{e2} \approx 0$, or $\lambda_{e1} \approx 0, \lambda_{e2} >> 0$. It means one curvature is large but the other is small, so that the local window is actually over a ridge. That indicates an edge.

- $\lambda_{e1} >> 0, \lambda_{e2} >> 0$. It means both curvatures are large so that the local window will be greatly different from its slightly shifted one. That indicates a corner.

The above classification is straightforward from the theoretical viewpoint. However, the discrete labelling is difficult to classify in practice. So Harris and Stephen proposed the corner response as:

$$C_{Plessey} = Det(\mathbf{M}_{plessey}) - K_{Plessey}(Trace(\mathbf{M}_{plessey}))^2 \tag{3.8}$$

where

$$\begin{aligned} Det(\mathbf{M}_{plessey}) &= \widehat{I_u^2}\widehat{I_v^2} - \widehat{I_u I_v}^2 \\ Trace(\mathbf{M}_{plessey}) &= \widehat{I_u^2} + \widehat{I_v^2} \end{aligned}$$

and $K_{Plessey}$ is normally set as 0.0401. The cornerness measure is then thresholded by pre-set threshold, and a non-maximal suppression is applied.

The Plessey is well known for its stable performance. However, due to the need to generate the derivatives, the Plessey corner detector gives displaced corner locations and is also sensitive to noise. It needs more intensive computation and the localization of detected corner is not accurate.

**Zheng-Wang Corner detector**

To improve the localization and computational complexity performance, Zheng-Wang-and-Teoh [41] propose an improved cornerness measure based on Plessey corner detector. Starting from the Plessey cornerness function as in Eq. 3.8, They deduced a new cornerness function by a serial of approximations as:

$$C_{Zheng-Wang} = I_u^2(u,v)I_{vv}^2(u,v) + I_v^2(u,v)I_{uu}^2(u,v) - K(u,v)(I_u^2(u,v) + I_v^2(u,v)) \tag{3.9}$$

where

$$K(u,v) = G(\sigma) \otimes [\frac{I_u^2(u,v) + I_{vv}^2(u,v) + I_v^2(u,v)I_{uu}^2(u,v)}{(I_u^2(u,v) + I_v^2(u,v))^2}] \tag{3.10}$$

Then the cornerness is compared with threshold, and non-maximal suppression is followed.

The cornerness function can be further approximated as the gradient module of the gradient direction, so that this corner detector is also called gradient-direction corner detector. This method does not need to conduct the Gaussian convolution

three times compared with the original Plessey, which reduces the computational complexity. Also, because its cornerness function avoids Gaussian smoothing, the localization performance is better than that for Plessey. However, the price to pay are the detection performance is inferior to Plessey and the second derivative term also introduces instability due to noise.

### 3.2.2 Edge-based Corner Detector

These methods normally consider corners as the junction of edges. The traditional approach is to segment the image and find the edges in chain code, then extract the points on boundaries with significant turning. Another approach that goes further from the original idea of Moravec is to use differential geometry to describe the local boundary function, then establish a cornerness measure and thresholding it.

#### Kitchen and Rosenfeld Corner Detector

In [42], the authors proposed several method to detect corners. The most successful one is to define a cornerness based on the rate of change of gradient direction along an edge multiplied by the gradient magnitude. Suppose the gradient direction is $\theta(u,v)$, its partial derivatives are:

$$\begin{aligned}\theta_u(u,v) &= \frac{I_{uv}(u,v)I_u(u,v) - I_{uu}(u,v)I_v(u,v)}{I_u^2(u,v) + I_v^2(u,v)} \\ \theta_v(u,v) &= \frac{I_{vv}(u,v)I_u(u,v) - I_{uv}(u,v)I_v(u,v)}{I_u^2(u,v) + I_v^2(u,v)}\end{aligned}$$

They project the gradient direction vector $(\theta_u(u,v), \theta_v(u,v))$ along the edge $(-I_v(u,v), I_u(u,v))$ and multiplied by the local gradient magnitude, yielding the cornerness as:

$$C_{K\&R} = \frac{I_{uu}(u,v)I_v^2(u,v) - 2I_{uv}(u,v)I_u(u,v)I_v(u,v) + I_{vv}(u,v)I_u^2(u,v)}{I_u^2(u,v) + I_v^2(u,v)} \tag{3.11}$$

In practice, a non-maximal suppression is applied to the edge magnitudes along

the gradient direction before using them for multiplication. By using second order derivatives, their method is also sensitive to noise.

**Wang and Brady Corner Detector**

Wang and Brady proposed this corner detector [43] by the measurement of the total surface curvature. This method is based on the observation that the image total curvature is proportional to the second order directional derivative in the direction of edge normal tangent, and inversely proportional to the norm of the edge normal. Supposing $t$ is the edge tangential direction that is perpendicular to the edge norm $n$, they estimated the total curvature $C_{total}$ as:

$$C_{total} \approx \frac{\partial^2 I(u,v)/\partial t^2}{\|\nabla I(u,v)\|} \qquad when \|\nabla I\| \gg 1 \tag{3.12}$$

By applying the false corner response suppression, the cornerness is defined as:

$$C_{Wang-Brady} = (\partial^2 I(u,v)/\partial t^2)^2 - S_{Wang}\|\nabla I(u,v)\|^2 \tag{3.13}$$

When $S_{Wang}$ is a constant measure of image surface curvature varying with different differentiation masks. The authors also proved that the grey level corner will be detected with a displacement that is linear to the standard deviation of Gaussian convolution. Because this method does not involve smoothing, the localization accuracy is one of its advantages. Another advantage is the computational complexity is low so it can be applied under real time environment.

### 3.2.3 Topology-based Corner Detector

These methods define corner as the interior geometric feature on image. The topological feature of differential geometry of corner is measured and on which the corner response is based.

### Beaudet Corner Detector

Beaudet [44] proposed a cornerness measure based on the Hessian matrix H, and this measurement is rotationally invariant. The cornerness is defined based on second order Taylor expansion as:

$$C_{Beaudet} = det(H) = \begin{vmatrix} I_{uu}(u,v) & I_{uv}(u,v) \\ I_{uv}(u,v) & I_{vv}(u,v) \end{vmatrix} \tag{3.14}$$

The cornerness is then thresholded for local maximum.

The corner detector is rotationally invariant so that it is stable to rotation transformation of the image. It is actually detecting the points with both high curvature and large gradient magnitude. However, it is not very stable to noise because of second order derivatives and the localization accuracy is not very good.

### Deriche-Giraudon Corner Detector

Deriche and Giraudon proposed their corner detector in [45] to mainly account for the localization error in Beaudet's. Their method is based on two observations:

- The exact position of a corner can be detected as a stable zero-crossing in the scale-space
- The local maximum in Beaudet's measure moves in the scale -space along the bisector line that passes through the exact position of the corner point

Then they designed and implemented their method in the following steps:

1. A Laplacian image is calculated
2. The Beaudet's measures $C1_{Beaudet}$, and $C2_{Beaudet}$ are calculated at two scales $\sigma_1$ and $\sigma_2$, and then a local maximum detection in all directions is performed. Suppose $\sigma_1 > \sigma_2$

3. For each local maximum $(u_1, v_1)$ in $C1_{Beaudet}$, search the closest local maximum of $C2_{Beaudet}$ in its local window, suppose that point is $(u_2, v_2)$

4. By the second observation above, the corner point must lie on the line through $(u_1, v_1)$ and $(u_2, v_2)$. It is actually located at the first zero-crossing of the Laplacian in the direction from $(u_1, v_1)$ to $(u_2, v_2)$.

By the multi-scale analysis, this corner detector can find the exact location of corner points. Another advantage is the corner detector is quite stable to noise. On the other hand, the computational complexity is relatively high because of the multi-scale analysis.

### 3.2.4 Structure-based Corner Detector

The structure-based corner detector detects corner by analyzing the surrounding local area structure directly. They define corner as an image point with strong two dimensional intensity change, and is well distinguished from the neighboring points. A popular definition to describe the local structure is USAN which stands for "Univalue Segment Assimilating Nucleus". According to the response in USAN, corner or edge can be detected.

#### SUSAN Corner Detector

The SUSAN corner detector is proposed by Smith and Brady in [46]. Suppose an arbitrary pixel in an image is surrounded by a circular local window, of which the central pixel is called *nucleus*. Each pixel's brightness in the circular local window is compared with the brightness of nucleus, and then the area mask that is defined by pixels with the same or similar brightness is called USAN, or Univalue Segment Assimilating Nucleus. If the nucleus is on a flat area on image surface, the USAN will be at its maximum; If the nucleus is lying on a straight edge then the USAN

will be half of the maximum; also if the nucleus is lying on a corner, the USAN will fall further, e.g. quarter of the maximum. Based on the above observation, they formulate the SUSAN principle as "An image processed to give as output inverted USAN area has edge and two dimensional features strongly enhanced, with the two dimensional features more strongly enhanced than edges". The SUSAN is simply the acronym of "Smallest Univalue Segment Assimilating Nucleus".

Mathematically, for each image pixel $u_0, v_0$, the corner response is defined as:

$$C_{SUSAN} = \begin{cases} g - n(u_0, v_0) & if\ n(u_0, v_0) < g \\ 0 & otherwise \end{cases} \tag{3.15}$$

where $g$ is geometric threshold of half of the mask size, and $n(u_0, v_0)$ is defined as:

$$n(u_0, v_0) = \sum_{u_0, v_0 \in W_{local}} e^{-(\frac{I(u,v) - I(u_0,v_0)}{t_{SUSAN}})^6} \tag{3.16}$$

and $t_{SUSAN}$ is the threshold for similarity on intensity.

Theoretically, SUSAN corner detector is accurate on localization, fast, robust to noise and versatile enough to deal with many types of junctions. Its underlying assumption is that the corner is at the junction of regions with equivalent or almost equivalent intensity. However, because the real images will not perfectly satisfy this assumption and the complexity of junctions that regions make, the algorithm will be sensitive to the setting of threshold and not be very stable especially on noisy images.

### Trajkovic-Hedley Corner Detector

Based on USAN model, Trajkovic and Hedley proposed their corner detector in [47]. They define corner as being the point that the intensity variation of nucleus for all line directions is large. Following this understanding, their cornerness measure is

proposed as:

$$C_{Trajkovic-Hedley} = \min_{p,p' \in W_{local}} [(I(u_p, v_p) - I(u_0, v_0))^2 + (I(u_{p'}, v_{p'}) - I(u_0, v_0))^2] \quad (3.17)$$

when $(u_p, v_p), u_{p'}, v_{p'})$ are two points on the boundary of circular local window intersecting with an arbitrary line passing through nucleus. Three cases are analyzed:

1. A flat point: there is at least one line so that $(u_p, v_p), u_{p'}, v_{p'})$ belong to USAN, which results in low reponse.
2. Point on an edge: There is exactly one line that is tangential to the edge so that $(u_p, v_p), u_{p'}, v_{p'})$ belong to USAN. The cornerness function is also low.
3. A corner point: in all directions, both points $(u_p, v_p), (u_{p'}, v_{p'})$ do not belong to USAN simultaneously , hence the cornerness measure is high.

They also proposed a multi-grid algorithm to decrease the computational complexity and suppress false responses. Also, they take advantage of inter-pixel approximation to deal with strong edges. However, the corner response will be large on diagonal edges as well as at corners, which will result in stability problem.

### COP Corner Detector

Another work based on USAN is proposed in [48] as COP corner detector. The COP stands for crosses as oriented pair. Two masks of oriented crosses are used: one responds strongly to $\pm 45^o$ edges, another responds to the horizontal and vertical edges. They pre-defined 15 raw directions, $d_{-i} (i = 1, ..., 15)$. The raw direction is obtained in $3 \times 3$ local window so it is noisy. To provide smoothing effect in real images, they define each probability, $p_i$ of probabilistic dominant directions, $pd_{-i}$ from the raw direction distribution as:

$$p_i = w_{ik} f_k \quad (3.18)$$

where $w_{ik}$ is the weight factor and $f_k$ is the number of occurrences of $d\_k$ in the $3 \times 3$ window.

Furthermore, the probabilistic dominant directions, $pd\_i$ is defined as:

$$pd\_i = max_{i=1,2,4,5} p_i \tag{3.19}$$

Also, because the corner is a point with great orientation changes, the cornerness measurement need to include at least two dominant orientations. They define:

$$T_{ij} = \int\int_{\Omega_i^j} p_i d\Omega_i^j \qquad (i = 1,2,3,4, j = 1,2) \tag{3.20}$$

where $T_{ij}$ represents the probability of $pd\_i$ in a sector, $\Omega_i^j$, with area size is about 6 pixels. Then the changing rate of $T_{ij}$ in i-direction, $D_{rate}$ is defined as:

$$Dr_i = absolute(T_{i1} - T_{i2}) \qquad (i = 1,2,4,5) \tag{3.21}$$

from which the corner dominant direction is defined by:

$$Dom\_i = max(Dr_k) \qquad k = 1,2,4,5 \tag{3.22}$$

Then the corner response function is defined as:

$$C_{COP} = Dr_i \times Dr_j \qquad (i \neq j) \tag{3.23}$$

and local maximum is accepted as corner. The implementation of their method contains four steps:

1. The inverted USAN area is calculated using COP

2. Four probabilities of 15 pre-defined raw directions from COP response are

computed

3. Corner candidates are classified to have multiple dominant corner direction according to six pre-defined corner types

4. The false alarms are discarded and corners are identified by taking local maxim

The concept of this corner detector is the same as SUSAN but they use two oriented cross kernel, COP (crosses as oriented pair) on detection. Their comparison results also show this method have satisfactory performance in detection and stability, but few applications on computer vision are based on it.

### 3.2.5 Discussion

Corner detection has been under intensive research over the past three decades, and plenty of methods have been proposed. The above sections only give a partial but relevant review for the existing algorithms. Due to space limitation, various ideas, like corner detection over scale space [49] [50] [51], affine invariant [52] [53] [54] and comparing survey [55] are not included here. However, the basic challenges are still the same: is it robust for pose variation? is it accurate and stable? can its results facilitate further processing? In [56], the experimental results showed that the corner detection is still an open research area, as there is only 3% of all corners between images can be matched for viewpoint changes beyond $30^o$ based on the combination of different detectors and descriptors. Based on the above review, the classical methods still have their shortcomings. Although the plessey detector give good performance as reported in [57], its accuracy is compromised by the required Gaussian filter. And this accuracy will result in large errors in calculating the relationship between images via matching. On the other hand, the SUSAN corner detector can give accurate results and perform very fast because the masking operation. Furthermore, each of the edge-based detectors discussed above work on its own and is

designed for generic applications. No domain knowledge has been used. So far, no edge-based detector has been designed for specific applications. This motivates us to try to enhance the performance by introducing domain knowledge. Based on this idea, a further avenue is open for investigation.

## 3.3 Point Feature Matching Review and Uncalibrated Two-view Matching

Given two images from different views of the same scene, the correspondence between them must be established in order to reconstruct the scene. This problem is generally called un-calibrated matching if the relationship of the two images is fully or partially unknown. Many methods and algorithms have been proposed. However, it is a very difficult task due to the variations between images such as pose variation, perspective distortion, occlusion, noise and sensor difference. Generally, the correspondence is found based on certain defined similarity measurement, A match is found if the similarity measure is maximum compared to other matches but subjected to certain constraints, such as a point or an image patch in one image can only be matched to a point (or an image patch) in the other view within an expected neighborhood. Another important constraint is the epi-polar constraint that was discussed previously in [58]. Matching paradigmns can be categorized into area-based and feature-based ones. The area-based method establishes the correspondence of image patches in one image to the image patch in the other image. And feature-based methods only find the correspondence based on selected features, e.g. points or lines. Due to the variations between images, the same part of a scene may look different from different views. This leads to the situation that the point pair with strongest similarity indication may not be the true match. As the result, a straightforward application of similarity measurement will generally result in appre-

ciable percentage of mismatches. The general solution is to take into account more constraints other than the similarity, such as epipolar constraint, continuity constraint, uniqueness constraint and order constraint [58]. Also, different optimization algorithm are applied to find out the true matches from noisy data, if the mismatch is defined as noise.

### 3.3.1 Similarity Measurement

There are various different methods to measure the similarity between two sets of image data. Some well known ones are standard cross correlation, SAD (Sum of Absolute Difference), SSD (Sum of Squared Difference), Mutual Information [59], Entropy of Difference, Kolmogorov-Smirnov Distance and so on. One widely used method is the ZNCC: zero-mean normalized cross correlation:

$$ZNCC(\mathbf{m}_1, \mathbf{m}_2) = \frac{\sum_{i=-m}^{m}\sum_{j=-n}^{n}[I_1(x_1+i, y_1+j) - \bar{I}_1(x_1, y_1)] \times [I_2(x_2+i, y_2+j) - \bar{I}_2(x_2, y_2)]}{(2m+1)(2n+1)\sigma_1 \times \sigma_2} \tag{3.24}$$

where

$$\bar{I}_k(x_k, y_k) = \frac{\sum_{i=-m}^{m}\sum_{j=-n}^{n} I_k(x_k+i, y_k+j)}{(2m+1)(2n+1)} \qquad k = 1, 2. \tag{3.25}$$

is the average intensity of the window centered at point $(x_k, y_k)$, and

$$\sigma_k = \sqrt{\frac{1}{(2m+1)(2n+1)}\sum_{i=-m}^{m}\sum_{j=-n}^{n}[I_k(x_k+i, y_k+j) - \bar{I}_k(x_k, y_k)]^2} \qquad k = 1, 2. \tag{3.26}$$

is the standard deviation of this window. The ZNCC value ranges from -1 to 1, which is represented that the two correlation window are not similar at all or identical, respectively.

### 3.3.2 Dense Two-Views Matching

The area-based matching methods do not detect features and match them, they work directly on the image patches and compute their similarity based on a predefined measure. Each point in one image will be examined on similarity with the points in another image, and the match with the highest similarity score is deemed the best match. The same process is done all pixels in the image, establishing dense correspondences between two images.

The cross correlation is taken as the similarity measure in [60] [61]. Different similarity measures are compared in [62], in which the author pointed out that ZNCC performs best according to their experiments. Detailed discussions can also be found in books [17], [63] [64]. To resolve the mismatching problem, the multiple-baseline [65] approach takes the idea that the more views are used for matching, the greater the probability of correct match.

The correlation window size is also an important factor for the matching results. Intuitively, the size of correlation window should be large enough to incorporate sufficient information for unique matching, but a large window will result in large perspective and scale distortion between the two images, and make the similarity measurement invalid. To account for this problem, in [66] the local intensity pattern is used to adapt the correlation window. Further in [67], the local variation and depth information are taken to decide the window size. Multiple window sizes are used in [68], and an optimal result is obtained based on the combination of these results by a genetic algorithm.

Dynamic programming has also been attempted to the matching problem as a global optimization search by progressively using the solution of sub-problem. It is introduced in [69] [70] on edge-based matching, and more recently have been applied to dense matching [71] [72] [73]. It can explicitly handle the partial occlusion problem

in images but the selection of occlusion cost parameter is not easy. It also needs the "images following ordering" constraint, and the consistency between scan lines is not easy to maintain.

For matching on a more global scale, relaxation labelling iteratively update the candidate pair of matches by the matching values of its neighbors. It is based on local calculation but obtain results in an overall global optimization manner. In this algorithm, the initial group of matches are produced, then the final results are recognized by constraint propagating. Different methods are proposed using different disambiguating constraints [74] [75] [76] [24]. The implementation of relaxation methods is complicated and its computational complexity is high.

There are many other techniques for two-view matching, such as simulated annealing, graph cuts. More details can be found in review [77].

### 3.3.3 Point Matching

The area-based method directly use the image intensity values to find correspondence which results in that it will be sensitive the the variations that will lead to intensity changes. On the contrary, with the significant local structure information carried by image intensities, the features, are comparatively more stable against variations between images from different viewpoint. The feature-based matching methods are thus intensively studied. A typical feature-based matching method normally includes feature detection and then followed by feature matching. Most feature-based algorithms are based on the assumption that the features are already detected accurately and stably across images. In practice, this assumption is somewhat ideal. In this thesis, our proposed method, which includes self-calibration, will start at the beginning by developing corner detectors that detect and retain only robust and accurate corners.

After corners are detected we will examine robust methods for finding reliable and accurate corner points for matching. Similarity measures such as the cross-correlation or sum of squared difference are examined. We will choose the zero-mean normalized cross correlation (ZNCC) for our purpose for reasons to be explained. These set of matched corner points form the kernel to estimate the camera geometry. Generally a local window surrounding a corner point is taken to calculate how similar this corner patch is to those in another image. The value of similarity between two corner patches is an indication whether these two corners in different images are representing the same real world point. The higher the similarity value, the stronger the indication of a true match. The pair with highest value is deemed the best pair of match. This process, though, is not as straightforward as described above. The different image views of the same object will be somewhat different because of pose variation, illumination variation, imaging system difference and other physical factors. On the other hand, the information that the local window around corner contains is also partial, so that the similarity value may be high even for corners that represent different real world object point. As a result, the similarity measure need to be carefully selected and a robust algorithm will be needed to find the real solution from large percentage of mismatches, or called outliers. We believe establishing correspondence is also an information exploration process, so a successful method needs to explore as much information as it can.

We categorize the feature-based matching method into the epipolar oriented method and the direct method. The epipolar oriented method finds some reliable match pairs and recover the epipolar geometry between images, then find the remaining matches using the found epipolar constraint. We call the other category the direct method because it does not have the two step process like the epipolar oriented method. The direct method yields output matches by using some other constraints directly.

**The epipolar oriented method**

To reduce the ambiguity introduced by direct application of similarity measurement, different constraints have been investigated to eliminate the false matches. The epipolar geometry is an outstanding one because it lower the searching dimension from 2D to 1D.

A landmark method is proposed by Zhengyou et al. in [58]. They use the traditional correlation and relaxation method to establish an initial set of matches, and then take the advantage of robust algorithm, LMedS (least median of squares) [78], to discard the false matches in the initial set, and calculate the true epipolar geometry based on the eight-point algorithm [79] [80]. With the recovered epipolar geometry, more matches can be found and the correlation-relaxation scheme is still applicable.

In the correlation-relaxation scheme, they use zero mean normalized cross correlation to find the candidate matches by a given threshold. Then, the relaxation process is designed to identify the correct matches among its neighbors withcontinuity, uniqueness and local homography constraints. The core of the relaxation technique is founded on a special definition of the strength of match function. It incorporates the calculation for the 'correctness' of a given pare of candidate match, and a some-winners-take-all update strategy, that is used to iteratively eliminate the detected false matches. Consider a matching pair $(\mathbf{m}_{1i}, \mathbf{m}_{2j})$, where $\mathbf{m}_{1i}$ is the $i$th corner in the first image and $\mathbf{m}_{2j}$ is the $j$th corner in the second. Suppose $N(\mathbf{m}_{1i})$ and $N(\mathbf{m}_{2j})$ are the neighborhoods of $\mathbf{m}_{1i}$ and $\mathbf{m}_{2j}$ within a disc of radius R, respectively. And, for $\mathbf{n}_{1k} \in N(\mathbf{m}_{1i})$ and $\mathbf{n}_{2l} \in N(\mathbf{m}_{2j})$, the strength of match function is defined as $S_M$:

$$S_M(\mathbf{m}_{1i}, \mathbf{m}_{2j}) = c_{ij} \sum_{\mathbf{n}_{1k} \in N(\mathbf{m}_{1i})} \left[ \max_{\mathbf{n}_{2l} \in N(\mathbf{m}_{2j})} \frac{c_{kl}\delta(\mathbf{m}_{1i}, \mathbf{m}_{2j}; \mathbf{n}_{1k}, \mathbf{n}_{2l})}{1 + dist(\mathbf{m}_{1i}, \mathbf{m}_{2j}; \mathbf{n}_{1k}, \mathbf{n}_{2l})} \right] \tag{3.27}$$

where $c_{ij}$ and $c_{kl}$ are the ZNCC values of candidate matches $(\mathbf{m}_{1i}, \mathbf{m}_{2j})$ and $(\mathbf{n}_{1k}, \mathbf{n}_{2l})$,

respectively. The $dist(\mathbf{m}_{1i}, \mathbf{m}_{2j}; \mathbf{n}_{1k}, \mathbf{n}_{2l})$ is the average distance of the two pairing:

$$dist(\mathbf{m}_{1i}, \mathbf{m}_{2j}; \mathbf{n}_{1k}, \mathbf{n}_{2l}) = [d(\mathbf{m}_{1i}, \mathbf{n}_{1k}) + d(\mathbf{m}_{2j}, \mathbf{n}_{2l})]/2 \tag{3.28}$$

where $d(\mathbf{m}, \mathbf{n}) = \|\mathbf{m} - \mathbf{n}\|$ is the Euclidean distance between $\mathbf{m}$ and $\mathsf{n}$. The $\delta(\mathbf{m}_{1i}, \mathbf{m}_{2j}; \mathbf{n}_{1k}, \mathbf{n}_{2l})$ is defined as:

$$\delta(\mathbf{m}_{1i}, \mathbf{m}_{2j}; \mathbf{n}_{1k}, \mathbf{n}_{2l}) = \begin{cases} e^{-r/\epsilon_r} & \text{if } r < \varepsilon_r, \\ 0 & \text{otherwise.} \end{cases} \tag{3.29}$$

where $r$ is the relative distance difference given by

$$r = \frac{|d(\mathbf{m}_{1i}, \mathbf{n}_{1k}) - d(\mathbf{m}_{2j}, \mathbf{n}_{2l})|}{dist(\mathbf{m}_{1i}, \mathbf{m}_{2j}; \mathbf{n}_{1k}, \mathbf{n}_{2l})} \tag{3.30}$$

and $\epsilon_r$ is a threshold value on the relative distance difference.

For the case when one point may have several potential candidate matches in another image, another disambiguation term is defined as:

$$U_A = 1 - S_M^{(2)}/S_M^{(1)} \tag{3.31}$$

where $S_M^{(1)}$ is the strength of match of $P_i$, and $S_M^{(2)}$ is the strength of match of second best candidate match.

The sum of all matches $S_M$ is taken as an energy function and minimized by a 'some-winners-take-all' updating strategy. The updating strategy only eliminate the candidate matches that have both relatively low strength of match $S_M$ and the unambiguity $U_A$. The algorithm is proved to converge.

In [81], a feature-based method is proposed based only on geometric constraints: the epipolar geometry and homography. The epipolar geometry is assumed known, so it can be seen as the second step for epipolar oriented methods. By a four steps

procedure, the homography is taken to iteratively remove the false matches and to obtain the true matches. This method is under the implicit assumption that the object surface must be smooth so that the local homography computation can work.

Different methods to estimate the fundamental matrix, are also proposed. In [82], they proposed a direct method that is complemented by feature-based method to obtain the fundamental matrix by pseudo-warping, using geometric and brightness constraints, to obtain the transformation of image pixels. A radial fundamental matrix and its linear estimation method is proposed in [83]. The situation that allow two images to have different distortions is studied, and the result is a four by four radial fundamental matrix that encodes the standard fundamental matrix and the distortion parameters of both images.

A review of methods that estimate the fundamental matrix is given in [84]. The comparison results show that LMedS (Least Median of Squares) produced the best results when low computing time is not required, and it is the most appropriate for outlier detection. Another review on recovering epipolar geometry is given in [85].

**Other methods for point feature matching**

The epipolar geometry is a strong constraint for images matching. However, there are other considerations. These are described below.

In [86], they used affine transformation to approximate the geometric relationship between two small homologous patches of the same continuous surface patch of the scene object. They explicitly compute the 2D affine transformation by an exhaustive search in the neighboring space of a pair of corners, then estimate the affine transformation and compute the repeatability. Consequently, the computational complexity is high. Moreover, the affine assumption may not be satisfied for the nearest five corners in its neighborhood. Finally, the test images used are either aerial images or the objects are almost planar ones, which are suitable for the affine

approximation.

In [87], a median filter is applied to detect the local outlier based on their assumption that the two images only undergo a translation. They claim that the cross correlation may not be the best similarity measurement in the ideal case. However, they also pointed out that under the circumstances of larger degrees of distortion between images, cross correlation performs better than SSD and others.

In [88], points of interest are characterized by differential invariants and color consistancy [89]. Then, an incremental algorithm is designed to find the correct correspondence by relaxation technique, which is also integrated the constraints of epipolar geometry and Delaunay triangulation. Finally, this method works on color images only.

A graph based approach using mutual information [59] is proposed in [90]. The plessey corner detector is used to find corners, and the corner matching problem is formulated as the solution of a maximum weight maximum cardinality flow problem on graph which incorporate mutual information. In this way, the solution is obtained in an efficient manner.

In [91], sparse matching of interest points using robust techniques based on the geometric constraint encoded by the fundamental matrix has been very successful. This method can be used for camera geometry estimation, self calibration for both calibrated and un-calibrated images. But in their work, they are propagating sparse matching to a quasi-dense matching.

More recently, a method that is robust to affine transformation is proposed in [92]. The method is based on two observations: the absolute curvature values of corners detected by contour-based corner detector changes slightly under affine transformations; also, the affine length of one curve is relatively invariant to affine transformations. An iterative procedure is designed to calculate the affine transformation

parameters, and match the corners with the help of this calculated transformation. The main drawback is it can only deal with global affine transformation that is only valid on planar objects.

Other works can also be found on uncertainty handling of corner matching [93], and external optimization [94].

### 3.3.4 Discussion

The area-based and feature-based matching methods are reviewed in this section. In the context of un-calibrated or weakly calibrated camera setting, it is obvious that feature-based matching method has some advantage over area-based ones. [95] [96] [58] [97] [98]. But, feature-based methods are inherently sparse. A general procedure is to estimate the transformation by taking advantage of feature-based matching method, and then use the calculated geometry to guide the dense correspondence. This procedure has been investigated by [99] [91].

At the same time, constraints other than epipolar geometry, such as smoothness, uniqueness, ordering and so on, should also be included to ensure good matching results. The landmark work following this idea is by Zhengyou in [58]. However, there is still room for improvement if we formulate the matching problem as an information exploration problem. We found in literature that almost all matching methods assume the features have already been reliably detected. This in practice is not easily achieved. Also, although robust methods like LMedS can give very good results, their time complexity is rather high. So we are motivated to design a computaional method that takes the advantage of LMedS but that can reduce the calculation time. This will be very meaningful for practical problems.

## 3.4 Reconstruction from Multiple Images

The reconstruction from multiple images could be split into two problems: dense correspondence problem and reconstruction problem. The dense correspondence problem is to establish the correspondence for each pixel on images. If the cameras are not calibrated, the self-calibration technique will be applied. For our works, a robust self-calibration algorithm is developed. The reconstruction problem can be formulated as the finding of the 3D shape/surface with the found dense correspondences. To deal with these two problems, various methods have been proposed in literature. We roughly categorize them as image space methods, which resolve the two problems sequentially, and direct 3D space methods, which resolve them simultaneously. At the same time, the methods could also be differentiated by their assumptions, and the ways they deal with variations because of viewpoint difference, perspective projection characteristics and sensor attributes.

### 3.4.1 Image Space Methods

#### Multi-view reconstruction

Most multi-view reconstruction methods resolve the correspondence problem based on the fronto-parallel hypothesis. Under this assumption, the cameras' retinal plane are identical and the object surface is made up of many small patches that are parallel to this retinal plane. Although it is somewhat an ideal assumption, it is applicable for many kinds of scene, and could also be valid for an initial scene model.

In [100] a four-step automatic reconstruction system of stationary 3D objects from multiple views is given. Although this method implements the volumetric reconstruction method of [101] as its fourth step, their main contribution is to show the feasibility to operate in image space. After calibrating the intrinsic parameters of

the moving camera, they establish an initial 3D shape from the first two image views. Then, a model-based shape and camera motion estimation is conducted based on this initial 3D shape, which contains a novel "sliding textures" method. The final 3D object is built by the volumetric reconstruction with a calibrated multi-view system.

Another idea is to use the subsets of input images to partially reconstruct the target object and then combine them to the final result. A simple method is proposed in [102] recently based on normalized cross correlation. The method is to reconstruct a depth map for each view with a confidence level calculated based on its several neighboring views. Then the depth map is merged with method in [103]. The idea is that the object shape can only be partially reconstructed with high confidence from each view, so that a combination of the confident portions from all views can give a good reconstruction result. The author also pointed out the limitations of their method. As there will be holes on reconstructed results due to low texture and each surface point to be reconstructed must be seen in at least three views.

Following in the same idea, the method in [104] used 3 to 6 neighboring cameras to establish the depth map, called the visible surface model, for every image using the method in [65]. And a complete surface model is made by fusing all the visible surface model with the method in [103] . A further improvement is made in [105]. Here the complete surface model is taken to restrict the search space for visible surface models, which are integrated into a new complete surface model. In this way, much of the false correspondences are then eliminated, and this processing is repeated until the model converges.

A mesh-based object centered method is proposed in [106] with partially relaxing of the fronto-parallel hypothesis. A depth map is calculated by using the method in [60] for each image, still based on the hypothesis. Then all depth maps are combined in 3D space to give an initial set of unstructured 3D point. After that, a mesh

is generated from the point cloud by the algorithm proposed in [107], which is optimized by a cost function. There are three terms in this cost function: a deformation term, a shading term and a multi-image correlation term. Being different from the above methods that calculate correlation in image space, here the multi-image correlation term computes it by sampling on the 3D mesh facet. In this way, it takes the place of fronto-parallel hypothesis and results in better accuracy. A similar idea can be found in [67] although it only uses adaptive window on image space.

**Structure-from-Motion**

Unlike the configuration with multiple cameras, the structure-from-motion method normally uses one camera. In SFM, something must move with time. Either the camera is moved with respect to the static scene or the camera is fixed and it captures frames of moving objects. A more difficult case will be when both camera and scene objects are moving simultaneously. Therefore, the structure from motion methods need to recover both the structure, or the shape of the object as in multi-view ones, and the motion, or camera parameters. The literature on structure-from-motion is huge so we only focus on the most prominent ones with arbitrary camera motion and multiple input images.

A landmark work is presented in [11] based on one uncalibrated video camera. Their work assumes that the camera has no skew and the principle point is close to the center of image. Based on this assumption, a set of corner features are tracked across images and from which the fundamental matrix is recovered. Then more correspondences can be found with the help of the newly estimated fundamental matrix. The reconstruction result is therefore a sparse set of projective points. A self-calibration method is then proposed to compute the camera parameters by the point correspondences, and then to upgrade the sparse reconstruction from projective to metric [108]. The author then rectified the images and applied a dense stereo correlation-based matching between images. The depth image is refined by

all input images and smoothed with a second-order spline. The final 3D surface is made by polygonizing the depth map.

A similar work is proposed in [109]. The self calibration method is adopted from [108] but includes both point and line feature correspondences. The sparse metric reconstructed points are best fitted with neighborhood planes, and a texture mapping is applied from the images that are most parallel to the current plane. The final result is a planar texture mapped 3D surface model. A volumetric approach is also discussed for non-polyhedral scenes by volumetric intersection from the calibrated cameras.

### 3.4.2 Direct 3D Space Methods

#### Volumetric reconstruction

The stereo vision and SFM approaches we discussed previously firstly find correspondences between images and then convert the correspondences to 3D points or other features in world space. The major challenge of these approaches is the determination of accurate and reliable correspondences. For face reconstruction, a dense matching must be established, whereas a feature-based method, though less sensitive to illumination changes, can only give a sparse set of corresponding points. At the same time, the correspondence accuracy is dependent on the amount of texture in the scene. A region with little texture will not encourage good correspondences. In addition, as the viewpoints get far from each other, the input images will contain increasing amounts of occlusions which will make the matching problem more difficult. In this section, the volumetric reconstruction approaches are reviewed. The key motivation is to avoid image correspondence search by working in a discretized scene space [111]. The multi-camera volumetric methods presented here enclose the scene to be reconstructed in a reconstruction volume [110]. This working volume is

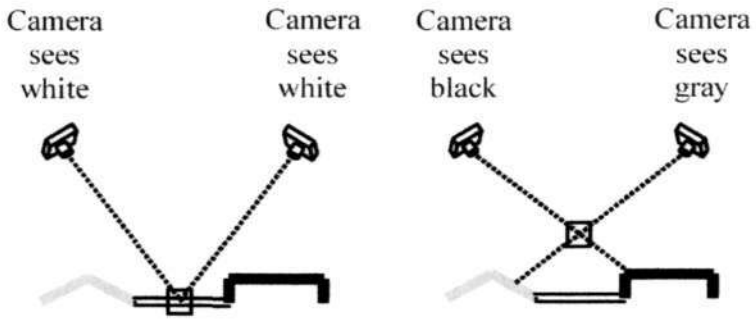



Figure 3.2: Color consistency check [110]

divided into voxels, which are typically small cubical volumes. The premise is that by projecting this voxel onto each of the image planes, the imaged regions should all exhibit very close color to one another.

The algorithms to be discussed in this section all use color consistency [111] to distinguish whether a voxel is on the surface of the scene or not. As shown in Figure 3.2, the color consistency means if a given voxel is on the surface of the scene, the color of all the pixels in each image that "see" this voxel should be consistent. Then, it can be used to distinguish points on a surface from points not on a surface. On the left of Figure 3.2, two cameras see consistent colors at a point on a surface, and on the right, the cameras see inconsistent colors at a point not on the surface. This consistency of colors can be defined by their standard deviation or, the maximum of the $L_1, L_2$ or $L_\infty$ norm between all pairs of colors. Then, a voxel will be considered on surface of the scene if the measure is less than a threshold. Usually, real world surfaces include abrupt color boundaries occurring at surface boundaries. The voxels that span such boundaries are likely to be visible from a set of views that are inconsistent in color. Thus, color consistency test will fail for these boundary voxels. This problem may be minimized with an adaptive threshold that increases when voxels appear inconsistent from single images.

### 3.4.3 Voxel Coloring

The voxel coloring algorithm [111] can reconstruct the "color" (radiance) at surface points in an unknown scene. It assumes that the cameras are fully calibrated, both the scene and lighting are stationary and, the scene surface follows a known, locally computable radiance function, e.g lambertian lighting model. The term "locally computable" means the radiance at any point is independent of the radiance of all other points in the scene, that is, the scenes global illumination effects such as shadows, inter-reflections and transparencies can be ignored. Under the second assumption, the radiance of each point is isotropic and can be quantified by a color vector.

The voxel coloring algorithm begins with a reconstruction volume of initially opaque voxels that encompasses the scene to be reconstructed. As the algorithm runs, opaque voxels are tested for color consistency and those that are found to be inconsistent are carved away. The algorithm stops when all the remaining opaque voxels are color-consistent. After being assigned the colors which the final voxels project to the input images, they form a model of the original scene. That is, the picture of the colored voxels from each input viewpoint should be as close as possible to the original image.

Another crucial point to note is the visibility problem. That is, when checking the color-consistency of one voxel, a given image pixel may not correspond to this voxel if there is a closer voxel occluding it, so the visibility of a voxel must be determined first. To resolve this problem, Seitz [111] introduces a novel geometric constraint on the input camera positions that enables a single visibility ordering of the voxels to hold for every input viewpoint. This **ordinal visibility constraint** is satisfied whenever no scene point is contained within the convex hull [111] of the input camera centers. This constraint requires the cameras be placed in the positions that all the voxels are visited in a single scan in near-to-far order relative to every

camera. A typical example is when all the cameras are placed on one side of the scene and scanning voxels in planes that go successively further from the cameras. In this way, every voxel is visited only once and the transparency of all voxels that might occlude a given voxel is determined before the given voxel is checked for color-consistency. As shown in Figure 3.3, an occlusion bit map is established for each input image with one bit per pixel. Initially, the mask is set to zero. When a voxel is found to be consistent, a bit in the occlusion bitmap is set to 1 for each pixel in the projection of a consistent voxel into each image, as shown in Figure 3.3 ($a$). When another consistent voxel project to pixels with value 1 in bitmaps of some viewpoints, as shown in black in Figure 3.3 ($b$), this voxel will be occluded in this viewpoint. Thus, the visibility set of the later voxel is simply the pixels in the voxel's projection whose occlusion bits are "0". In other words, occlusion bit being "1" means in this viewpoint, the voxel is occluded by the voxel computed before.

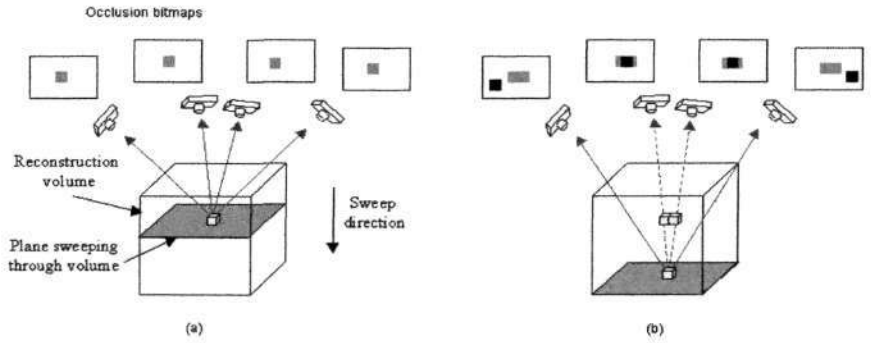



Figure 3.3: Occlusion bitmap [110]: a. no occlusion. b. With occlusion.

The runtime of voxel coloring is another key consideration. It is related to the number of voxels. Choosing a small voxel size will result in good spatial resolution, but the runtime will be very long and the color consistency constraint will be less reliable. On the other hand, if a large voxel size is chosen, it will be fast to compute and with more texture region will give more reliable consistency check but the spatial resolution will be poor. Prock [112] proposed a coarse-to-fine approach which can

achieves as much as 40 times speedup. A rapid reconstruction is performed using large voxel size. Then some voxels that are less occupied by scene are likely to be carved away. Before the voxels are divided into smaller size, the carved voxels that are adjacent to uncarved ones are added back into the model. Then the algorithm goes into the next step of voxel coloring with divided voxels. The same procedure will repeat as long as warranted by the input image resolutions.

Voxel coloring is elegant and efficient. It is the complement of the stereo correspondence method and thereby avoids this inherent problem of passive stereovision. However, the ordinal visibility constraint is a significant limitation. Because all the voxels are scanned from near-to-far order relative all the cameras, the cameras can not surround the scene. This restricts the camera placement. In the following sections, we will talk about methods that allow arbitrary camera placements.

### 3.4.4 Space Carving

In the voxel coloring algrithm, we take note that it may fail if voxels that are color consistent in the final model are also carved away. It is shown in [113] that this can be prevented if a suitable consistency measure is used. Specifically, the measure must be monotonic: if a set of pixels are found to be inconsistent, then the superset of those pixels will also be inconsistent. Since the algorithm only changes opaque voxels to transparent or carves them away and never vice versa, the remaining opaque voxels will become more visible as the algorithm runs. In other words, the algorithm never carves a voxel it should not. Furthermore, it is proved in [114] that the algorithm finds the unique color consistent model that is a superset of any other consistent model and calls it the photo hull. Here the term "photo hull" means the spatially largest set of points in 3D space that project to photo-consistent colors in the reference images. In addition, a point in space can be photo-consistent under two conditions: ($i$) the point does not project to the background in the images and

(*ii*) when it is visible, the light emiting from the point (e.g radiance) in the direction of each reference view is equal to the observed color in the photograph. Then, photo hull is the largest volume that projects to photo-consistent colors in the reference images and contains the scene surfaces being reconstructed.

Space carving [114] is a reconstruction method based on the theory of photo-consistency under general conditions similar to that for voxel coloring. It can be seen as a generalization of silhouette-based techniques like volume intersection to the case of gray-scale and full-color images, and generalizes voxel coloring and plenoptic decompostion to the case of arbitrary camera geometry. Actually, the aim of space carving is to reconstruct a virtual object that is photo consistent with viewpoints of the input images, and that is the tightest hull containing the real world object. Space carving adapts background constraints and radiance constraints to obtain this photo hull.

As compared with voxel coloring, space carving achieves the goal of allowing arbitrary camera placement by using multiple scans along the three axes. In each scan, the voxels on the sweep plane are evaluated for color consistency as done with voxel coloring. In practice, it forces the scans to be near-to-far, relative to the cameras, by using only images whose cameras have already been passed by the moving plane. Thus, when a voxel is evaluated, the transparency is already known of other voxels that might occlude it from the cameras currently being used.

As in voxel coloring, space carving also requires very precise calibration of the cameras. In [115], a variation of space carving called "approximate space carving" is presented that addresses this problem of inaccurate camera caliberation. In addition, in [116], a new consistency check criterion was proposed instead of the criterion in [114]. That is, to sample the image by a new statistical technique instead of the centroid sampling method. The consistency check is based on F-statistic: the data is modelled as a fixed value plus noise. A voxel is only removed if there is sufficient

evidence to suggest that the image samples could not have had the same mean. This is evaluated by computing the ratio of the between-class variance, to the within-class variance.

The space carving algorithm is effective, but it is also conservative, in the sense that the determination of a voxel's consistency is made only using a subset of the cameras that can see the voxel in one plane-sweeping. This may result in uncarved voxels. In the next part, an algorithm that determines a voxel's consistency using all the cameras in which it is visible is introduced.

### 3.4.5 Generalized Voxel Coloring

Another extension of voxel coloring is generalized voxel coloring proposed in [117]. Unlike voxel coloring, GVC allows input cameras to be placed at arbitrary locations in and around the scene. Furthermore, GVC uses the entire set of images from which the voxel is visible in comparison with that of space carving that usually uses only a subset of those images. The computation complexity is also reduced due to effective data structure implementation. There are two variants of the algorithm, called GVC and GVC-LDI. They use different data structures to compute the visibility of voxels as is shown in Figure 3.4.

In GVC, every voxel is assigned a unique ID and an item buffer is constructed for each image. The item buffer contains a voxel ID for every pixel in the corresponding image and the distance from this voxel to the corresponding viewpoint. When rendering voxel to item buffer, if the distance from the camera to the present voxel is less than the distance stored in the pixel, this pixel's stored distance and voxel ID are over-written with the present voxel. Thus, after rendering, each pixel will contain the ID of the closest voxel that projects onto it and this is the visibility information needed. When checking the color consistency of a given voxel, all the

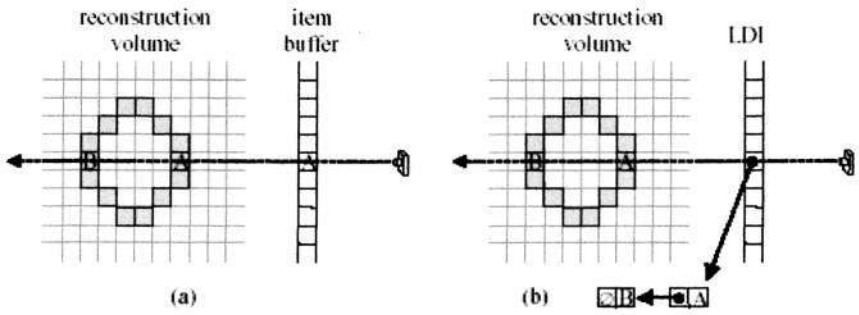



Figure 3.4: The data structures that are used to compute visibility. An item buffer (a) is used by GVC and records the ID of the surface voxel visible from each pixel in an image. A layered depth image (LDI) (b) is used by GVC-LDI and records all surface voxels that project onto each pixel [117].

pixel with the same voxel ID in its item buffer will be computed. Since carving a voxel changes the visibility of the remaining uncarved voxels, the item buffers need to be updated. But, because the updating is so time-consuming, we can only update it periodically and the item buffer will be often out-of-date. However, it still can produce a good output because the carving is conservative.

In GVC-LDI, a layered depth images is used instead of item buffer, which can efficiently and immediately update the visibility information when a voxel is carved and also can precisely determine the voxels whose visibility has changed. By extending the item buffer, the LDI stores for each pixel a list of all the surface voxels that project onto this pixel according to the distance of the voxel to the image's camera. The head of LDI stores the closest voxel, which is the same voxel in an item buffer.

When a voxel is carved, the LDI can be updated immediately. Voxels can be added to or deleted from an LDI by first finding the pixels in the voxel's projection and then adding or deleting the voxel from the LDI lists for those pixels. When LDIs are updated, any voxel that moves into or out of the "nearest" position in a pixel's LDI list has different visibility after the update. The main advantage of using LDIs that it is only necessary to recheck a voxel's consistency if its visibility changes.

### 3.4.6 Graph cut based 3D reconstruction

In graph cut based 3D reconstruction methods, generally an energy function is built based on the input information for the target object. Then a specialized graph are constructed for the energy function such that the minimum cut on the graph also minimizes the energy function, hence to recover the target object. Furthermore, the minimum cut problem can be efficiently calculated by max flow algorithm [118].

An energy function is established in [119] with treating the input images symmetrically and handling the visibility properly. This energy function is then minimized by a graph cut to obtain the local minimum. To preserve the discontinuities, the local spatial smoothness is added into the energy function according to a 'master' camera. A generalized version of [119] is proposed in [120], in which the smoothness factor can be added according to all the cameras. [119] then become a special case of the formulation in [120]. Both methods compute a set of depth maps using multi-baseline stereo with graph cuts, then merge the results into a voxel space by computing the intersections of the occluded volumes from each viewpoint.

A volumetric formulation method based on graph cut is proposed in [121]. The 3D space is labelled into 'object' and 'empty' by this method. The visual hull is used for visibility check and an energy function based on photo consistency [114] is minimized by cutting weighted graphto obtain a minimum cut solution. In addition, the recovered object representation is viewpoint independent.

As there are always smoothness factor built in the energy function implicitly or explicitly, the graph cut methods are inclined to give over smooth surface for the actual high curvature ones. In this thesis, we take another approach and try to avoid this minimal surface problem by explicitly work on the volumetric space.

### 3.4.7 Level set

Level set is a numerical technique that is designed for tracking the evolution of interfaces or shapes [122]. It defines the problem in one higher dimension space than the problem space itself. In the typical level set method, the interface or shape is normally embedded as a zero level set of signed distance function. The target is to match the interface with the zero level set of the level set function, and resulting in an initial value partial differential equation, the Hamilton-Jacobi equation, for the evolution of the level set function. In this way, the numerical computation can be easily performed without parameterizing the interfaces or shapes. The level set method can also easily deal with changing topology, especially when this topology is the target as in 3D reconstruction of computer vision.

A quasi-dense approach is proposed in [95]. They resolve the correspondence problem by adopting the quasi-dense matching method in [91]. To effectively eliminate false correspondences, a best- first strategy is taken to propagate from seeds with the epipolar geometry being enforced. The authors claim their matching method can deal with much larger separated image views. However, this claim is weakened by their similarity measurement, zero mean normalized cross correlation, which is not suitable for wide baseline situation, even though the method calculates the local homographies for seeds propagation. Based on the quasi-dense correspondence, a set of 3D quasi-dense points are reconstructed. And a level set based algorithm is designed by formulating the dense reconstruction as minimal surface searching problem. This formulation incorporates both the quasi-dense 3D points and all 2D images information, such as correlation and silhouette. To evolve the surface efficiently, a bounded regularization method is proposed and its stability is proved. A large set of results were demonstrated for the success of the proposed method.

Other 3D reconstruction works based on level setting can be found in [123] [124] [125].

### 3.4.8 Discussion

The advantage of 3D space oriented methods over image space oriented methods is it tries to test each point in 3D space by measuring its projected points' similarity. The image space oriented methods may find the correspondences that are hard to be converted to one 3D point. On the contrary, the 3D space direct methods take the constraint that it must be a real 3D point first, then test its similarity. It is a compulsory constraint that in principle the ray from each camera optical center should converge into one 3D point.

The volumetric reconstruction avoids having to find matching across the input images, which is a very hard task in computer vision. However, this set of methods needs accurate camera calibration, large number of images and the condition that the background can be easily eliminated.

All the above methods are based on the Lambertian lighting model. The same carving criterion, namely color consistency, is also used in these methods. The difference among them is the approach in dealing with the visibility problem. Because both voxel coloring and space carving depend on sweeping planes, the camera placements are restricted. It is also assumed that voxels in the same voxel layer can not occlude each other in these two methods. On the contrary, in GVC, this possibility is considered.

## 3.5 Conclusion

3D reconstruction from 2D images is a very tough problem in computer vision. The critical sub-problems include calibrating camera, establishing image correspondence, finding the 3D shape and so on. From the literature review in this chapter, we observe that the 3D reconstruction problem, and its critical sub-problems, is still

an open question for researchers.

Due to the ill-posed characteristic of 3D reconstruction problem, we formulate it as an information exploring problem based on camera system with known intrinsic parameters. Hence, we start to recover partial information at the beginning, being the extrinsic camera parameters, then with this partial information, we start to find more information until the target 3D shape is recovered.

The point matching methods in the literature usually are based on the assumption that point features, or corners are available. Unfortunately the detection performance has direct effects on the matching results. Consequently, we try to consider the corner detection and matching tasks jointly to improve the matching accuracy. An improved SUSAN corner detector and a new energy function are proposed to work cooperatively for uncalibrated image matching problem.

For the correspondence problem, we take the same idea as space carving to test each 3D point. However, the color consistency criteria is too weak and is only valid with larger number of input image views. We relax this criteria by a common but much stronger one, correlation. We also try to avoid the hard decision by an iterative searching procedure. Our experiment is only based on 4 or 5 images for reconstructing the frontal face, and can be applied to more images configuration.

# Chapter 4

# Improved SUSAN Corner Detector for Matching

SUSAN corner detector [46] is well known because of its fast detection and accurate localization. Due to its mask-based operation, it is also reasonably robust to noise. It defines corners as junctions of regions with equivalent or similar intensities. Because the synthesized images are normally free of arbitrary noise, this assumption is quite reasonable on synthesized images so that SUSAN corner detector has good performance on them. On practical images, though, this assumption is somewhat ideal given the complicated situations in real applications. Although known to have among the best localization accuracy, SUSAN is also reported to be unstable to viewpoint change [126], so it may detect a lot of false alarms. In a practical situation such as in passive stereovision, both the localization and stability are critical. Although robust algorithms are available for finding the correspondence between images, the existence of too many false alarms will deteriorate their performance. At the same time, for applications such as for computing the epipolar geometry, only a small number (8) of point matches will be sufficient if they are accurate and stable [127] [79]. Most corner matching methods assumes that corners have been

ideally detected. However it is not true of most the time. It becomes clear that accurate corner detection and robust matching are linked to the overall stability and accuracy of the 3D reconstruction. This inspires us to think that the design of corner detections and the matching algorithm must necessarily be done together to achieve the needed accurate and robust 3D results.

In this chapter, we study the limitations of the SUSAN corner detector and suggest a way to improve on its stability. The improved SUSAN corner detector is also a part of our corner matching scheme. We will compare our corner detector with the original SUSAN and the Plessey detector. The result of this improved SUSAN corner detector is then taken as the input of our proposed corner matching algorithm to give corner matching on binocular images.

## 4.1 Analysis of SUSAN corner detector

The SUSAN principle [46] can detect the corners with ideally maximum curvature. However, this condition is ususally not met given the complicated local structure and noise image, resulting in presence of false corners.

In [46], two enhancements have been proposed to suppress the false alarms. The first premises that the center of gravity of the USAN should be far away from the nucleus, and the second premises that all the pixels on the line from nucleus towards the center of gravity within the local mask must be part of USAN. Although these two additions can partially solve the problem, they are not well defined to account for all of the possible complicated real situations. Some possible examples are illustrated in the following.

The first enhancement will not hold under certain conditions such as specific corner types, e.g. X-type corner or noisy image, where the center of gravity will be close

to nucleus for a true corner. Fig. 4.1 shows a simplified example where a smoothed X type corner is illustrated. Here and onwards, we will use '+' to represent the nucleus and '*' to represent the gravity center of USAN in figures, and USAN is shown in grey. When the threshold is set to an empirical value used in [46] such as 25, the center of gravity is at the nucleus, but it will be rejected by the above first condition.

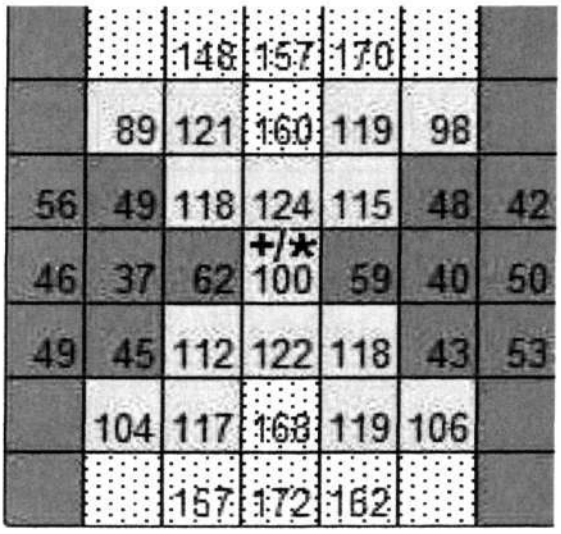



Figure 4.1: X type corner will be rejected

Moreover, as illustrated in Fig 4.2 , an edge pixel may be accepted by the first enhancement because its small USAN and far away center of gravity.

The second enhancement cannot effectively reject false alarms either because the enforcement of continuity will not be satisfied in the real image very well. As illustrated in Fig 4.3. although the enhancement can also reject a false alarm, it can reject the true corner at the same time.

In this section, based on the analysis of SUSAN principle, we find that although a portion of SUSAN corners will be the pixels with maximum curvature, there will be many false alarms too. Furthermore, both of the two false alarm suppression enhancements can not properly distinguish true corners from false corners.

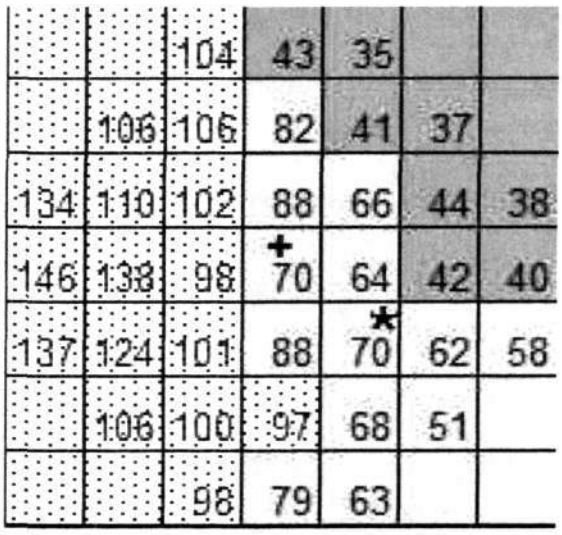



Figure 4.2: An accepted corner, but it lies on an edge

In this chapter, we propose an improved SUSAN corner detector for the above problems. By analyzing the SUSAN principle, we observe that the USAN will be small in situations where the current nucleus is not a corner. At the same time, the false alarm detection procedure is not effective enough. It is designed to detect false alarm only under specific situations defined by the above two enhancement whereas corner patterns are more varied. In some situations, it can even wrongly discard the true corner. Based on the neighborhood analysis of a given nucleus that satisfy SUSAN principle, we derive our proposed corner detector. We calculate its two main curvatures, and then introduce an algorithm to identify and discard false alarms. Although this calculation is expensive, the total computation complexity will not increase remarkably since only a smaller data set will be involved. In other words, we believe that, for our purpose at hand, less but good corners will be better than more corners but containing high percentage of false and inaccurate ones. Compared with SUSAN and Plessey corner detectors , our corner detector can detect the false alarm effectively without compromising the localization accuracy, and thereby contributing to real applications. However, the price to pay is to detect less corners.

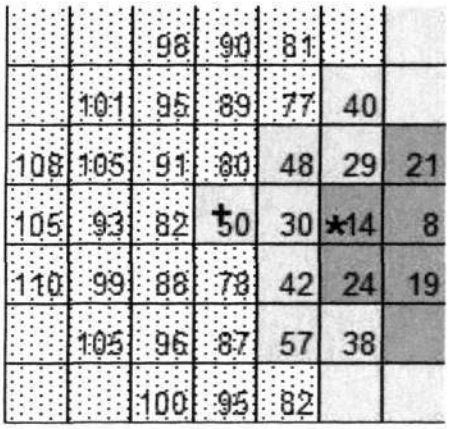



Figure 4.3: Real corner but rejected

## 4.2 Improvement to the SUSAN Corner Detector

Theoretically, the SUSAN corner detector is accurate on localization, robust to noise and sound enough to deal with all type of junctions. Its underlying assumption is that the corners are the junction of regions with equivalent or similar intensity(Fig. 4.4). In addition, the other two assumptions of SUSAN that the center of gravity of USAN is away from the neuclus and the contiguity, requires that the USAN must be clustered and be pie-piece shape as illustrated in Fig. 4.4. So it is safe to generalize that the basic structure of local topology around a corner is pie shaped composing of two straight edges. For example, a uniform area can be interpreted as zero edge, a local area with one edge can be considered as the two edges with 180 degree intersection, and so on(please refer to Fig. 4.5).

However, because the real images will not perfectly satisfy this assumption and the complexity of junctions that regions make, the algorithm will be sensitive to the setting of threshold resulting in a lot of false alarms. Although SUSAN has developed two criteria to detect the false alarms, the mechanisms are not so effective as illustrated in last section. The task of how to deal with the false alarms is still a demanding problem.

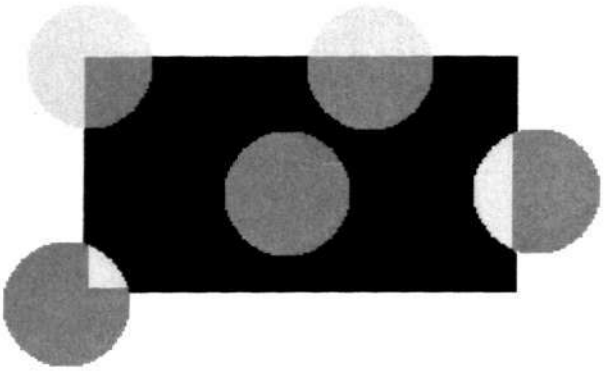


Figure 4.4: USAN demonstration, USAN is represented by the dark grey area

We address this problem by formulating corner detection as a local pattern classification task based on local information. That is, we need to measure the local pattern and detect the ones that are not corners from USAN corners[i]. We start from the intensity changes caused by a displacement $\Delta\vec{r}$ from a given pixel with position vector $\vec{r}$:

$$\Delta I = I(\vec{r} + \Delta\vec{r}) - I(\vec{r}) \approx \Delta\vec{r} I'_{\vec{r}}(\vec{r}) \tag{4.1}$$

Having used the Taylor expansion up to the first order and supposing $\Delta\vec{r} = [\Delta x, \Delta y]$ and $I_{\vec{r}}(\vec{r}) = [I_x, I_y]^T$. We obtain:

$$\Delta I^2 = (\Delta x I'_x + \Delta y I'_y)^2 \tag{4.2}$$

and finally

$$\Delta I^2 = \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \begin{bmatrix} I'^2_x & I'_x I'_y \\ I'_x I'_y & I'^2_y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{4.3}$$

The matrix above captures the local structure of a given point. Its two eigenvalues

[i]We use USAN corner to represent the corners detected by SUSAN principle without false alarm suppression.

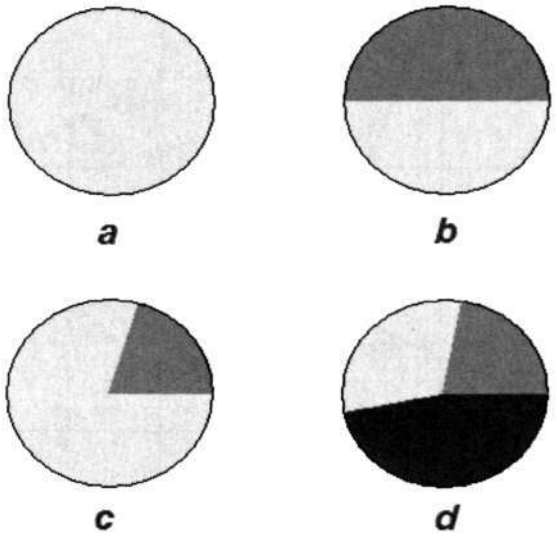



Figure 4.5: Our definition for corner. a. Uniform Area; b. One Edge; c. Two Edges; d. Three Edges

are proportional to the two local principal curvatures, which are always perpendicular to each other. To account for noise effects, a Gaussian smoothing is applied on each image patch [40] in practice.

The local curvatures are a good representation of local spatial information, and they are used in the Plessey corner detector to detect 'point of interest' when both curvatures are high enough. However, we are not going to identify a "point of interest" by the eigenvalues as Plessey did. Instead, we are trying to take advantage of them to identify the false alarms of USAN corners.

The main obstacle is whether there are common attributes between USAN corners and 'points of interest'? To answer this question, we need to analyze the fundamental assumption of SUSAN corners, so we started from the ideal assumptions at the beginning of this section.

In the following figures, we take a 37 pixels area as local mask and draw the pixels of USAN in dark grey. Fig 4.6, 4.7 and 4.8 show the situations when the angle between

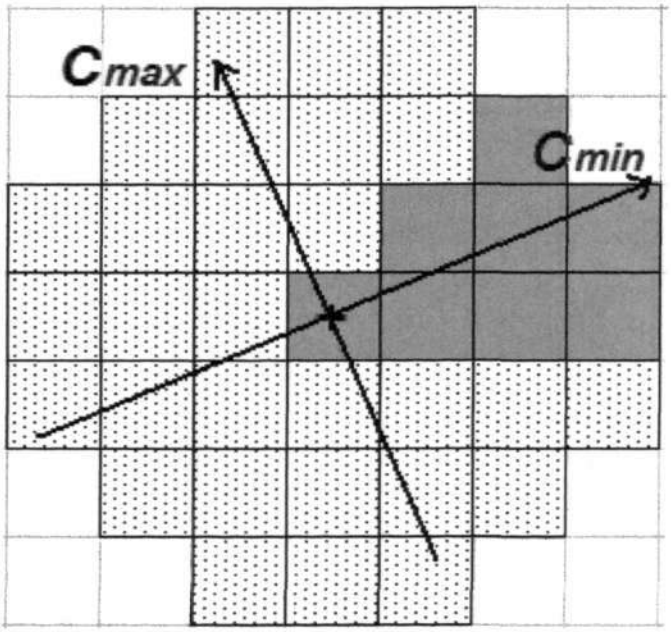



Figure 4.6: The situation where the angle between two edges is less than 90 degree

two edges is less than 180 degrees. Under all the three situations, the SUSAN corner detector should report positive corner response with different parameter settings. We also illustrate the directions that corresponding to the principal curvatures. We use $C_{max}$ and $C_{min}$ to represent them. Furthermore, we qualitatively demonstrate the maximum and minimum curves in Fig. 4.9. Bearing in mind that corners should be reported in the three cases, we can observe that the minimum curve is following a step function.

Simultaneously, we also illustrate the situation when the angle of two local edges passing nucleus is equal and greater than 180 degrees in Fig. 4.10 and Fig. 4.11. We also qualitatively illustrate the corresponding principal curves in Fig 4.12. We can see that although the intensity profile along the maximum curve is following step functoin, the one along the *Cmin* is just a flat line.

From the above analysis, we observe that the ideal SUSAN definition for a corner actually coincides with the 'point of interest' definition. The observation is that their

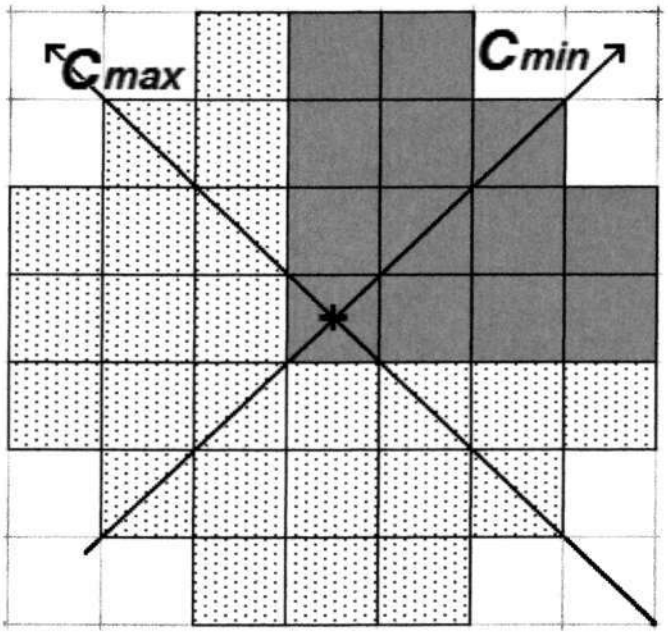



Figure 4.7: The situation where the angle between two edges is equal to 90 degree

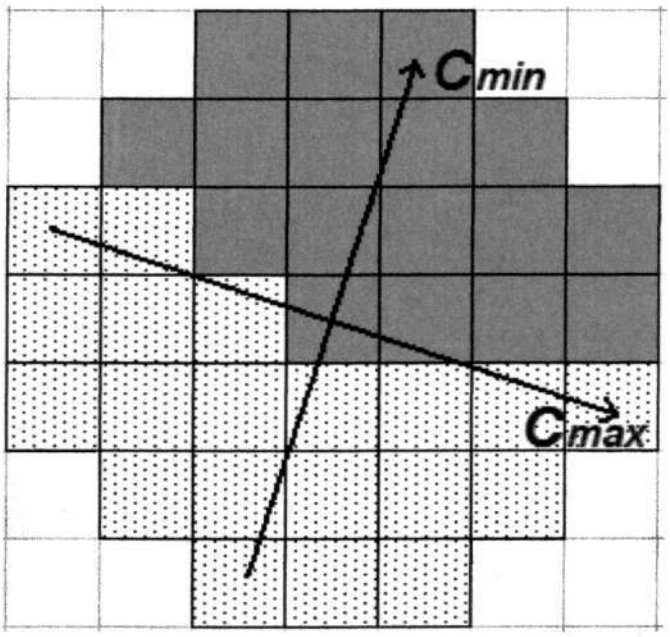



Figure 4.8: The situation where the angle between two edges is greater than 90 degree

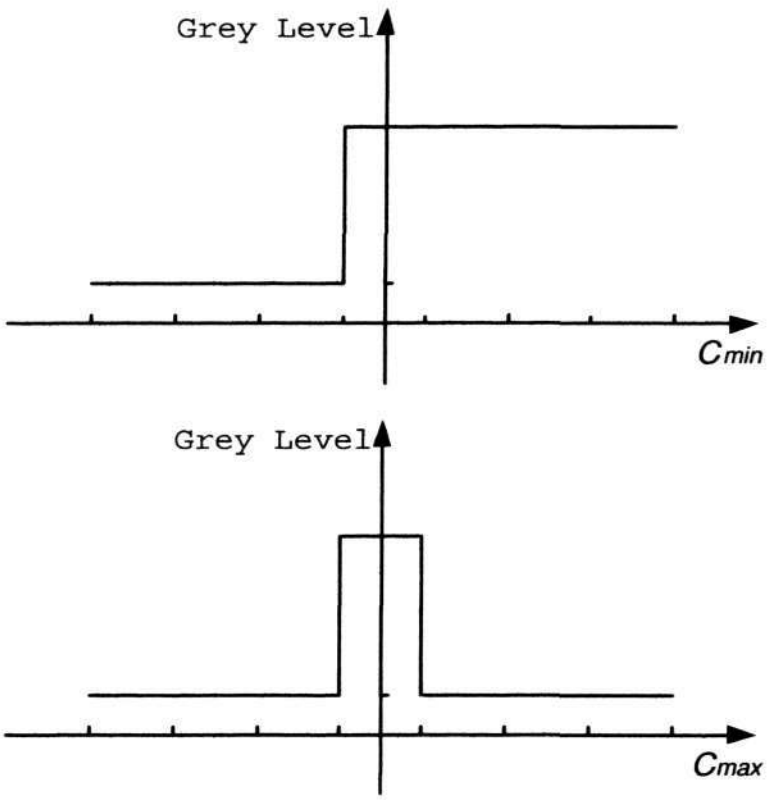



Figure 4.9: The profiles of principal curvatures along $C_{min}$ and $C_{max}$ for cases in Fig. 4.6, 4.7 and 4.8

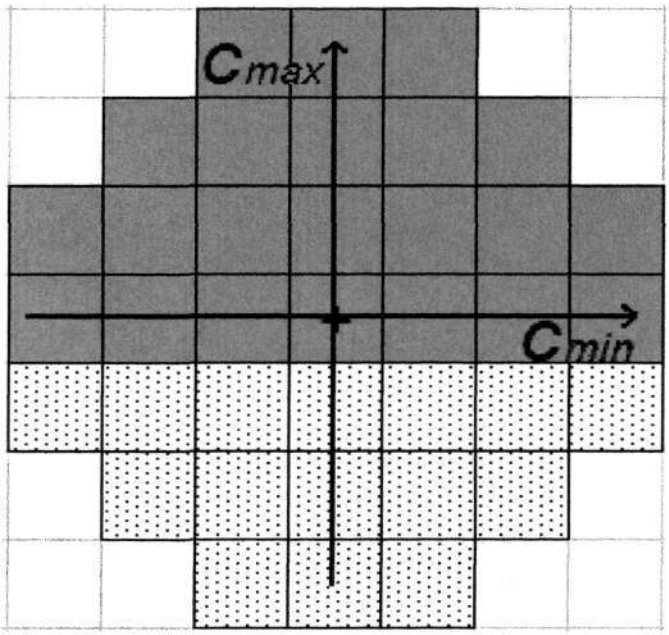



Figure 4.10: The illustration of straight edge

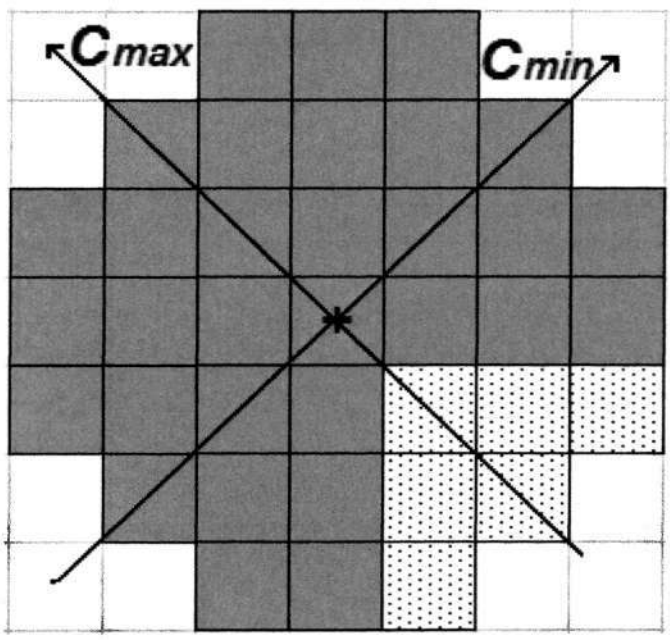



Figure 4.11: The situation where the angle between two edges is equal to 270 degree

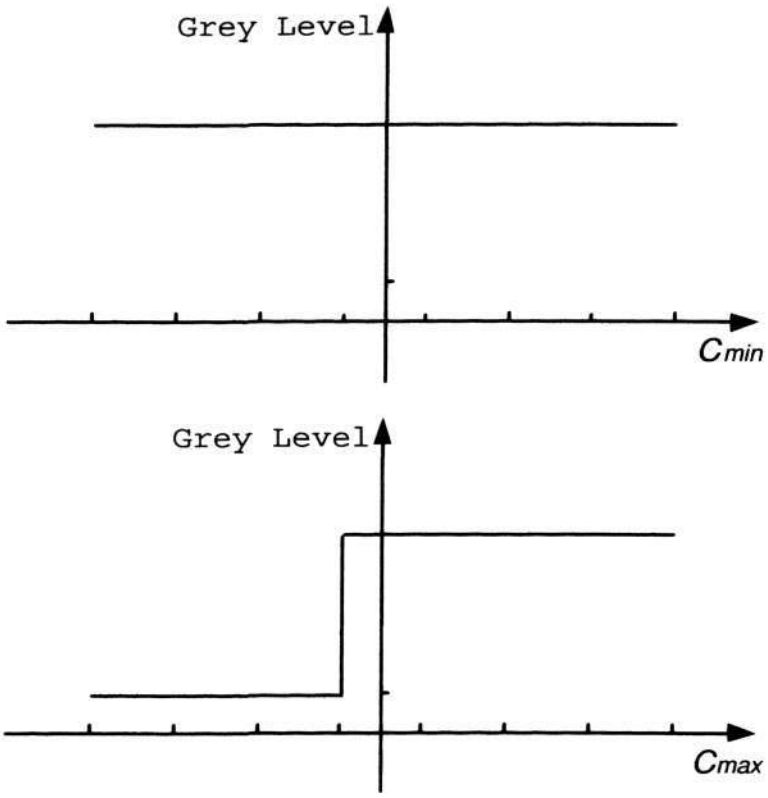



Figure 4.12: The principal curvatures along $C_{min}$ and $C_{max}$ for cases in Fig. 4.10 and 4.11

detected corner locations are not identical because of two points: a) the displacement of Plessey corner[ii]; and b) the false alarms from SUSAN mechanism. However, our objective is to measure the local information contained in the autocorrelation matrix and take advantage of it to identify the false alarms. So we define false alarms as the USAN corners with low local curvatures. Then, USAN corners with small eigenvalues will be considered as false alarms by our hypothesis.

The calculation of eigenvalues is computationally intensive so that we try to avoid it. From Fig. 4.9 and Fig. 4.12, we can see that our parameters for curvatures must be set to exclude the situations where the minimum curvature is a straight line, or almost straight ones. So with respect to curvature, we set a threshold to a step function. This is done by excluding the points where both curvatures are less than a given step height, although the threshold is set more or less empirically.

Compared with the original SUSAN corner detection, our improved algorithm will perform better for situations listed in section 4.1. In Fig 4.1, our algorithm will not reject the X-type corner, given that the differential scale is not less than 1, which is common in practice. Although our threshold for the smaller eigenvalue of autocorrelation matrix is determined experimentally, we can set it very low only to prevent detection of an edge, which is the situation in Fig 4.2. Definitely our algorithm will not accept the nucleus as corner in Fig 4.2, whereas the original SUSAN corner detector will. In Fig 4.3, because of large local curvatures, it is apparent that our algorithm will correctly accept the nucleus as a corner whereas SUSAN will not.

In this section, we try to improve the SUSAN corner detector by proposing an effective false alarm suppression algorithm, which is in turn based on the analysis of local structure described by auto-corelation matrix. In this way, the stability of SUSAN was improved without loss of the localization accuracy of detected corners.

[ii] we use 'Plessey corner' to represent the corner detected by Plessey corner detector.

However, the price we have to pay is to recover less number of detected corners, but this, according to our hypothesis, is a better sacrifice than to have poorer localization or accuracy in real applications.

## 4.3 Experimental Results and Analysis

We have shown in the above how our method worked on synthetic images in order to explain the principal. In this section, we test our improved SUSAN on standard testing images with another two corner detector: SUSAN and Plessey. To illustrate the practical performance of our algorithm, the corner detectors are also applied on feature based matching. The terms ImpSusan will represent our improved SUSAN corner detector; OrgSUSAN to represent the original SUSAN corner detector in [46], and Plessey to represent the Plessey corner detector in [40].

In Fig. 4.14, the results on a synthetic image are given. All the three corner detectors perform well on this image. From the results we can see that, our ImpSUSAN performs as well as the OrgSUSAN both on detection rate and localization. For the results of Plessey, we can see that although the detection rate is good enough. giving only one false response, the localization is poor. For the T-type corners in the left, the more contrast from left to right, the more displacement will occur. This phenomenon has been identified by some researchers as the Plessey corner detector is actually finding the point with equal derivatives in x, y directions [46].

Fig. 4.15 gives the results on an affine transformed synthetic image. All the results found by our ImpSUSAN are true corners, however, we lost one corner in top left because of the small eigenvalues due to the weak orthogonal contrast. The OrgSUSAN can detect all the corners, but one false response is reported due to the indention and noise of the transformed image. The Plessey corner detector also gave good detection rate, giving only two false alarms. At the same time, its localization

problem is also poor in this image, which is its theoretical shortcoming.

The testing on real image is given in Fig 4.16. The OrgSUSAN performs the worst on this image: false alarms can be seen on strong edge such as the eaves of the house in the bottom right area. Its false alarm rejection mechanism did not work effectively. The Plessey gave much better results on detection rate but had poor location accuracy theoretically. Our ImpSUSAN performs slightly better than the Plessey does. But, the total computational complexity of our algorithm is only about 1/5 to 1/4 of that of the Plessey corner detector and with better localization accuracy. In addition, compared with OrgSUSAN, our method give a more effective rejection of false alarms.

Lastly, sixteen pairs of images are selected from the ALOI database [128]. A well-known featured-based matching algorithm [58] is then applied to find the point correspondence and then the epipolar geometry is computed between images. In order to quantitatively measure the results, we define the residual error as in [12] for the N pairs of corresponding points.

$$Residual = \frac{\sum_{i=1}^{N}(d_{1i} + d_{2i})}{N} \tag{4.4}$$

where $d_{1i}$ and $d_{2i}$ are the distance of the $i$th matching candidate pair to their corresponding epipolar lines in each image. The error is then computed by averaging the sum of distances. As is shown in Fig. 4.13, our ImpSUSAN performs better than or equal to the other two detectors on 13 pairs of images out of 16. This result is not surprising in that although our algorithm finds less number of detected corners, it finds the corners with both high accuracy of localization and high stability, which will remarkably reduce the computation time of our robust algorithm in feature based matching [58].

The computation on the above calculation is intensive because both derivatives

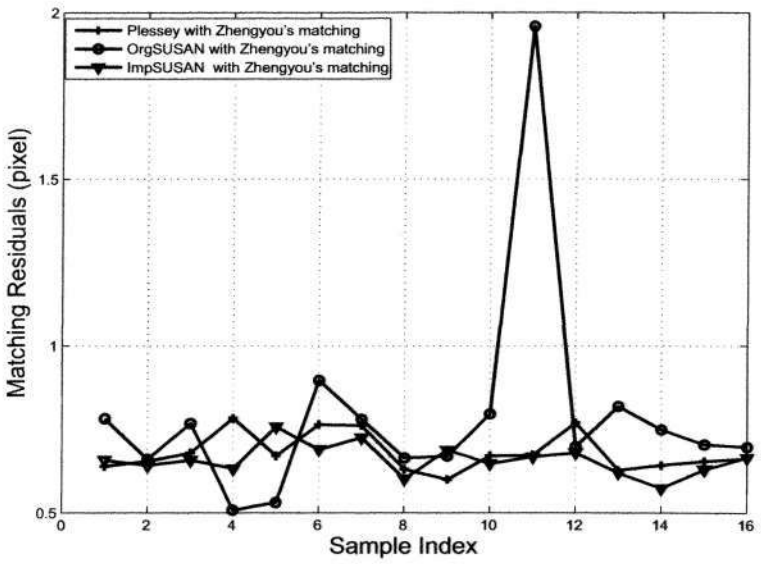



Figure 4.13: Application of corner detectors on stereo matching.

and eigenvalues are needed. Actually the computational complexity is one of the shortcomings of Plessey corner detector. However, our calculation is on a much smaller data space compared with the full image dimension as in Plessey. Although the data space will depend on image content, generally it will be less than one tenth of the full image. As the result, the practical computation time is only 1/5 to 1/4 compared with Plessey corner detector based on our non-optimized programming.

The actual results of the first pair of images are given. Fig. 4.17 shows the corners detected by SUSAN, Fig. 4.18 gives the results of ImpSUSAN. Compared with Fig. 4.17, a portion of corners detected by SUSAN have been removed. Fig. 4.19 shows the results by Plessey detector.

The corners are then input into the matching scheme to obtain the matches and to calculate the fundamental matrix. Then, based on the manually selected matches, the corresponding epipolar line are plotted in the input images together with the matches from Fig. 4.21 to Fig. 4.23. The short white lines are illustrating the

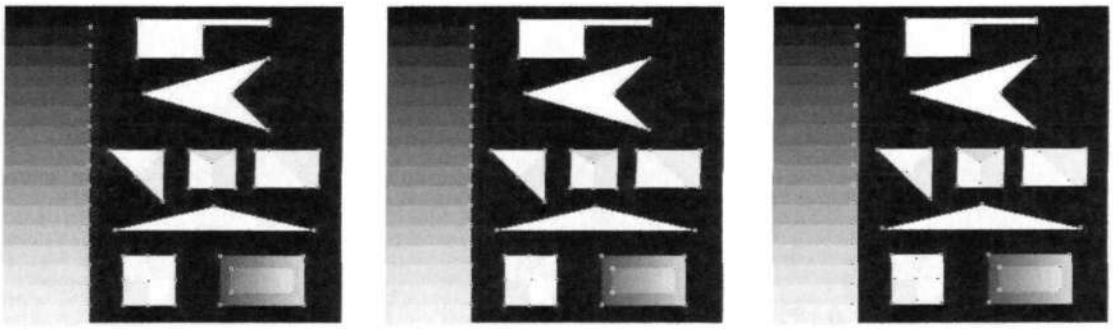


Figure 4.14: Results on a Synthetic Image. Left: ImpSUSAN; Middle: OrgSUSAN; Right: Plessey

matches, showing that one end of the short line in one image is matched to the the other end of the short line in the other image. Fig. 4.20 is showing that the epipolar lines based on manually selected matches. Theoretically, the epipolar lines should be closed to horizontal direction as the left and right image are captured by a horizontal movement.

## 4.4 Conclusion

In this chapter we proposed an improved SUSAN corner detector, that in principle is to add an extra false corner detection and rejection step, for low level image processing. A more effective false alarm rejection method is designed by analyzing the local neighborhood of the SUSAN corners, and the eigenvalues of second moment matrix is thresholded to discard false response. We have made the theoretical bridge between SUSAN corner and the point of interest based on the analysis that both of them are detecting points with maximum curvatures. Experiments has been done on both standard testing images and application of binocular feature matching, the results have confirmed efficiency of our ImpSUSAN.

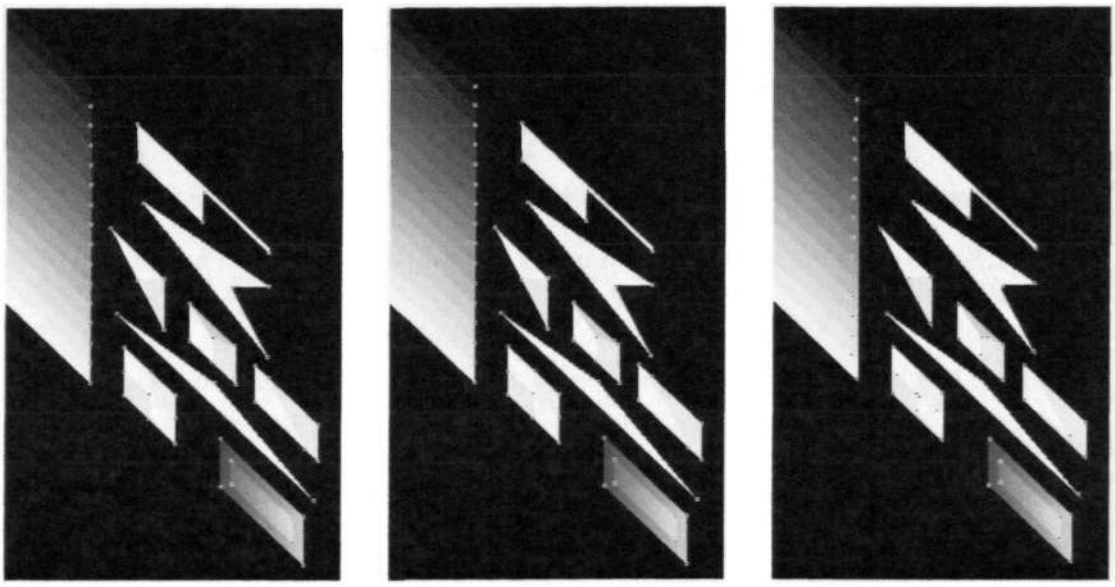


Figure 4.15: Results on Transformed Synthesis Image. Left: ImpSUSAN; Middle: OrgSUSAN; Right: Plessey

Figure 4.16: Results on Standard Building Image. Left: ImpSUSAN; Middle: OrgSUSAN; Right: Plessey

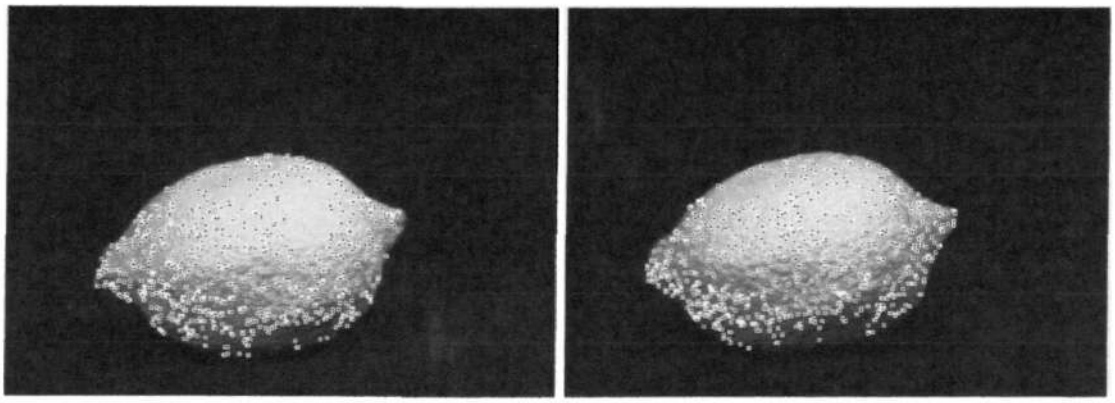


Figure 4.17: Corners by SUSAN

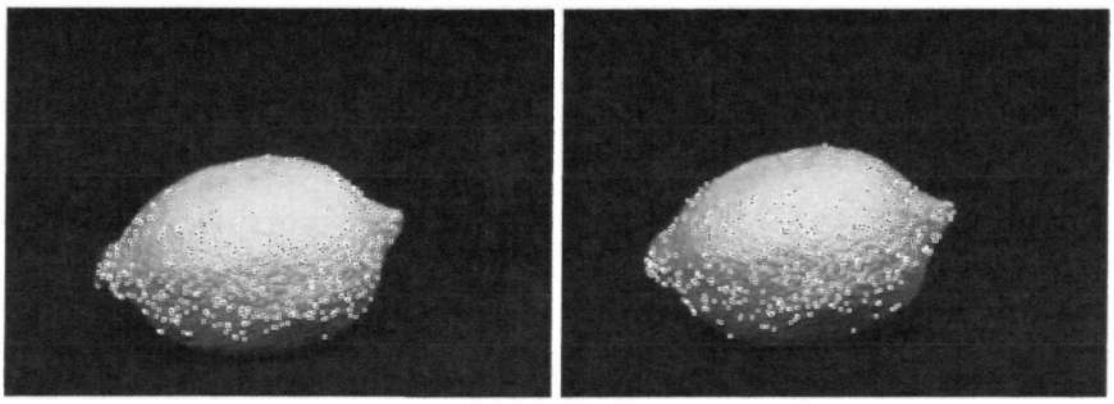


Figure 4.18: Corners by ImpSUSAN

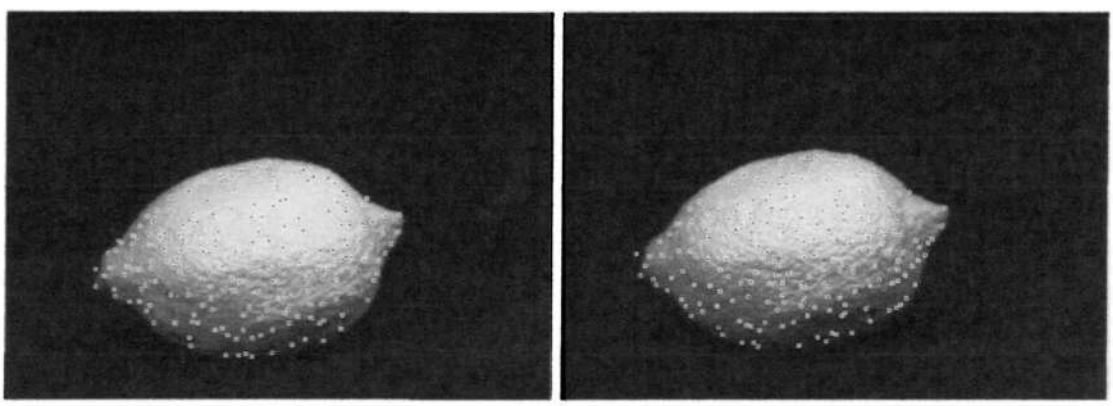


Figure 4.19: Corners by Plessey

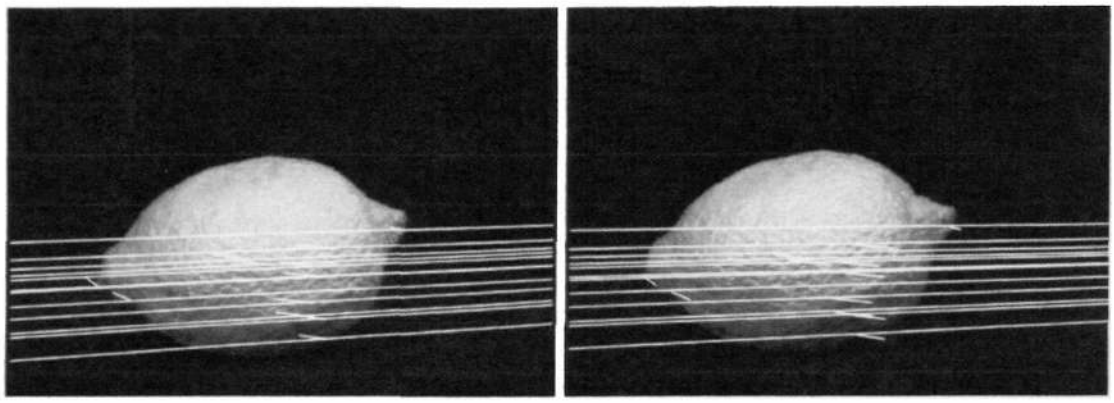


Figure 4.20: The Results by manually selected matches

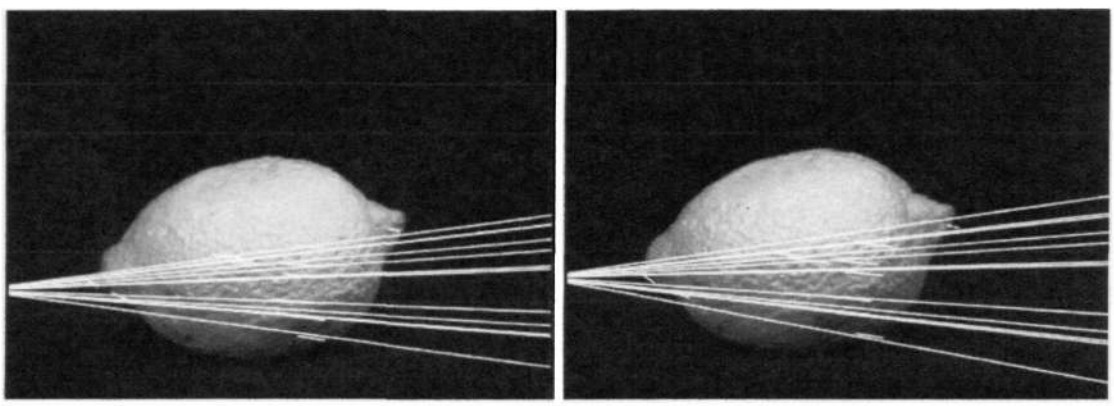


Figure 4.21: Epipolar Line by Zhengyou's method with SUSAN

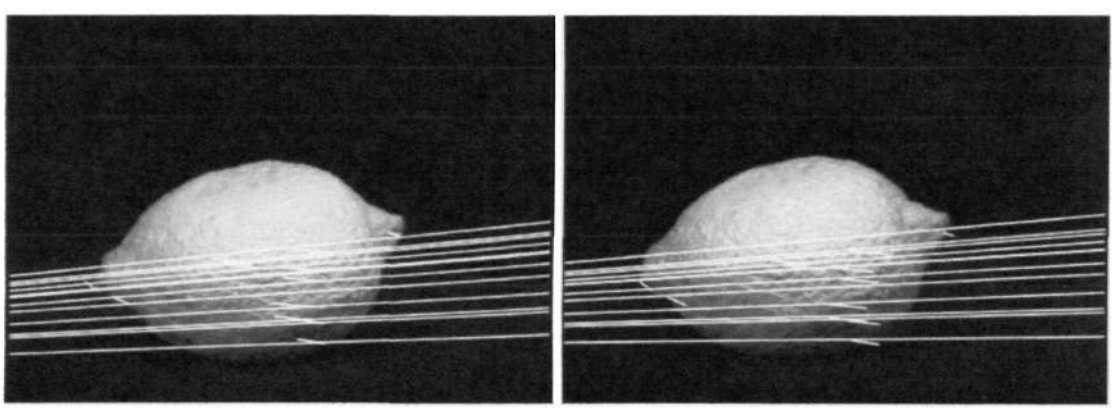


Figure 4.22: Epipolar Line by Zhengyou's method with ImpSUSAN

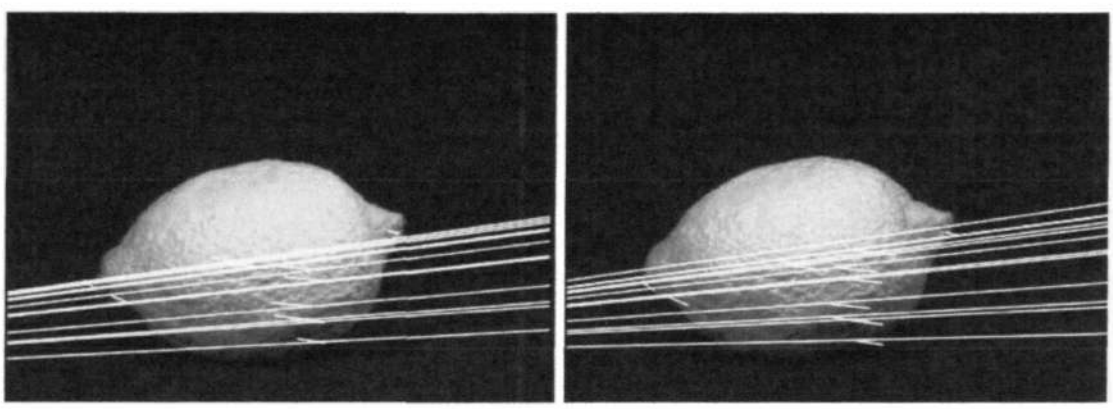


Figure 4.23: Epipolar Line by Zhengyou's method with Plessey

# Chapter 5

# The Two-View Corner Matching Strategy

Corners are good primitives to represent invariant local information in image. In the context of 3D reconstruction, the images are normally captured from different points of view, and are thus subjected to projective transformations. Corners are a good choice of primitives for the finding of the fundamental matrix between images since corners are invariant to projective transformations.

The present set of corner detectors developed, though able to find correct matches, unfortunately finds incorrect/false matches as well. These false matches play havoc in the next stage of stereo matching. Robust algorithms such as the LMedS in [58] were developed to handle these outliers. Such algorithms are computationally expensive because of the necessary iterative search. If instead, these false matches are minimized first, then perhaps this computationally intensive search can be much reduced and the matching made more robust. This is the motivation of our approach, to unify the detection of corner points with the stereo matching algorithm.

In this chapter, we will investigate how to use our detected corners based on our

method in last chapter and proposed new way to match them. Instead of assuming ideal corners, our matching method is actually jointly considered with the corner detection, that is, we try to only detect the robust and accurate corners to reduce the searching space during matching.

For our proposed method, the 3D object can be non-planar but should be and locally continuous. As such, we will decompose the object into sufficiently small planar patches. From projective geometry, a pair of corresponding patches can be related by a projective transformation, for a pair of matching patches that is assumed locally planar, this transformation will be affine. So our task is to recover the global projective geometry by local affine transformation from matching a group of corners.

## 5.1 Analysis of a Classic Corner-based Matching Method

In [58], a landmark work is proposed for matching corners. Their idea is to establish an initial set of corresponding corners by traditional correlation and relaxation methods. The correlation results, called matching candidates, are examined by a relaxation technique to find the good matches.

The basic idea of relaxation is to identify the 'correct' matches by its neighbors based on applicable constraints, e.g. local homography. The key idea of this relaxation in [58] is a 'strength of match' function which calculates the support to the neibhbouring matches according to certain constraints. Further the sum of all 'strength of match' is taken as an energy function to be minimized. During the iterative minimization, bad matches are discard by a "some-winner-take-all" strategy. So the iterative procedure of disambiguating candidate matches is as follows:

**Iterate{**

- Compute the strength of match for each candidate match and construct the energy function.
- update the matches by minimizing the total energy.

**}until the convergence of the energy**

The initial set of corresponding corners are then used to find the fundamental matrix by LMedS robust algorithm.

The crux of the matching method in [58] is the 'strength of match' function:

$$S_M(\mathbf{m}_i, \mathbf{n}_j) = c_{ij} \sum_{\mathbf{m}_k \in N(\mathbf{m}_i)} \left[ \max_{\mathbf{n}_l \in N(\mathbf{n}_j)} \frac{c_{kl}\delta(\mathbf{m}_i, \mathbf{n}_j; \mathbf{m}_k, \mathbf{n}_l)}{1 + d_{ave}(\mathbf{m}_i, \mathbf{n}_j; \mathbf{m}_k, \mathbf{n}_l)} \right] \tag{5.1}$$

The $c_{ij}$ and $c_{kl}$ are the ZNCC values of candidate matches $(\mathbf{m}_i, \mathbf{n}_j)$ and $(\mathbf{m}_k, \mathbf{n}_l)$, respectively. The $d_{ave}(\mathbf{m}_i, \mathbf{n}_j; \mathbf{m}_k, \mathbf{n}_l)$ is the average distance of the two pairing:

$$d_{ave}(\mathbf{m}_{1i}, \mathbf{n}_j; \mathbf{m}_k, \mathbf{n}_l) = [dist(\mathbf{m}_i, \mathbf{m}_k) + dist(\mathbf{n}_j, \mathbf{n}_l)]/2 \tag{5.2}$$

where $dist(\mathbf{m}_i, \mathbf{m}_k) = \|\mathbf{m}_i - \mathbf{m}_k\|$ is the Euclidean distance between $\mathbf{m}$ and $\mathsf{n}$. The $\delta(\mathbf{m}_i, \mathbf{n}_j; \mathbf{m}_k, \mathbf{n}_l)$ is defined as:

$$\delta(\mathbf{m}_i, \mathbf{n}_j; \mathbf{m}_k, \mathbf{n}_l) = \begin{cases} e^{-d_r/\epsilon_{d_r}} & \text{if } d_r < \varepsilon_r, \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

where $d_r$ is the relative length difference given by

$$d_r = \frac{|dist(\mathbf{m}_i, \mathbf{m}_k) - dist(\mathbf{n}_j, \mathbf{n}_l)|}{d_{ave}(\mathbf{m}_i, \mathbf{n}_j; \mathbf{m}_k, \mathbf{n}_l)} \tag{5.4}$$

and $\epsilon_r$ is a threshold on the relative length difference.

First, the 'strength of match' function inherently contains the translation transformation between the corner of a pair of matching candidate. This translation can be simply obtained by the difference of coordinates of the two corners in their own image coordinate system. Second, from the equations above, we can understand that Eq. 5.2 means the support from neighbor matching candidate is inversely proportional to distance. And from Eq. 5.4, we can know the distances must be similar. So the equations are calculating scale transformation, which is also one component of affine transformation. However, the affine transformation is actually applicable from the basic assumption that the surface is locally planar, in another words, the relationship from two image of a planar surface can be described as an affine transformation [12]. Based on this understanding, we find that there is more information we can take advantage of other than this from only considering the translation and scale transformations. Our contribution is to re-design the 'Strength of match' function to incorporate more information.

## 5.2 The Proposed Matching Procedure

We follow the usual step to first find a small group of robust point matches from matching candidates, then using these points to compute the epipolar geometry between the two images. Ideally, this set of point matches should be evenly distributed on the images respectively. Finally, we use the calculated epipolar geometry as guide to find a large group of matches. In order to find the first set of robust matches, we take the same updating strategy and robust computing method in [58] but with a different energy function that we formulate to estimate the epipolar geometry. The proposed energy function, as the key of the matching procedure, will take 2D affine transformation into account, and is minimized iteratively.

Our assumption, that the little patch on the object surface can be assumed planar,

is a fair assumption for a large class of objects, including the face. Based on the above statements, a pair of images for the same scene object but at different pose can be related by a projective transformation,. Hence, for a pair of matching planar patches, this transformation will be a homography and can thus be approximated by on affine transformation. Then, our energy function takes advantage of this affine transformation: we use scale, rotation and translation changes without considering the shear effects to approximate this affine transformation. This approximation is valid for binocular stereo camera or for structure-from-motion cases. The image transformation assumption has been used by many proposed matching methods. However, we differ by integrating it into our robust matching strategy, which itself is a part of the point matching system. The following gives the implementation details of our algorithm in this paper.

Suppose there are $M$ corners in the first image and $N$ corners in the second image from our ImpSUSAN. Also suppose there are $G$ matching candidates after ZNCC. Then, for the $g^{th}$ matching candidate $(\mathbf{m}_i, \mathbf{n}_j)$ where $g \in [1, G]$, $i \in [1, M]$ and $j \in [1, N]$, we define $R(\mathbf{m}_i)$ and $R(\mathbf{n}_j)$ as the neighborhoods of $\mathbf{m}_i$ and $\mathbf{n}_j$ respectively. Hence we define the $h^{th}$ ($h \in [1, G]$ but $h \neq g$) matching candidate $(\mathbf{m}_p, \mathbf{n}_q)$ as neighbor candidate of $(\mathbf{m}_i, \mathbf{n}_j)$, where $\mathbf{m}_p \in R(\mathbf{m}_i)$, $p \in [1, M]$ and $\mathbf{m}_q \in R(\mathbf{m}_j)$, $q \in [1, N]$. From the above definitions, we then obtain vectors $\mathbf{u}_{ip}$ and $\mathbf{v}_{jq}$ as:

$$\mathbf{u}_{ip} = \mathbf{m}_p - \mathbf{m}_i \tag{5.5}$$

$$\mathbf{v}_{jq} = \mathbf{n}_q - \mathbf{n}_j \tag{5.6}$$

Then the 2D affine transformation $\mathbf{A}$ between $\mathbf{u}_{ip}$ and $\mathbf{v}_{jq}$ can be written as:

$$\begin{aligned} \mathbf{v}_{jq} &= \mathbf{A}\mathbf{u}_{ip} \\ &= \mathbf{S}_c\mathbf{S}_h\mathbf{R}\mathbf{u}_{ip} + \mathbf{t} \end{aligned} \tag{5.7}$$

where $\mathbf{S}_c$ represents the scale transformation, $\mathbf{S}_h$ represents the shear transformation and the $\mathbf{R}$ describes the rotation. $\mathbf{t}$ is the translation from $\mathbf{m}_i$ to $\mathbf{n}_j$, which can be derived by substraction of their coordinates. Because we only consider uniform scale, rotation and translation transformation, Eq.5.7 becomes:

$$\mathbf{v}_{jq} \approx s\mathbf{R}(\alpha)\mathbf{u}_{ip} + \mathbf{t} \tag{5.8}$$

where $s$ is uniform scale instead of $\mathbf{S}_c$. Then, the transformation between $\mathbf{u}_{ip}$ and $\mathbf{v}_{jq}$ can be approximated by scale $s$ and rotation angle $\alpha$ inherent in $\mathbf{R}$, where

$$s(\mathbf{u}_{ip}, \mathbf{v}_{jq}) = \frac{\|\mathbf{v}_{jq}\|_2}{\|\mathbf{u}_{ip}\|_2} \tag{5.9}$$

$$\alpha(\mathbf{u}_{ip}, \mathbf{v}_{jq}) = \arctan(\|\frac{\mathbf{u}_{ip} \times \mathbf{v}_{jq}}{\mathbf{u}_{ip} \cdot \mathbf{v}_{jq}}\|_2) \tag{5.10}$$

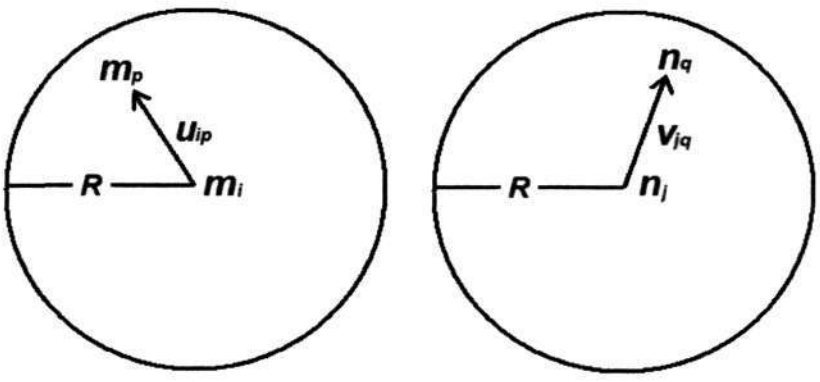



Figure 5.1: The Illustration of Affine Approximation

Then based on the scale and rotation changes derived here, we can establish an energy function as follows.

## 5.3 The New Energy Function

Our energy function will be implemented in the traditional relaxation technique. The relaxation relies on the fact that true matches should satisfy the same projective transformation globally and thus the equivalent affine transformation locally. This implies that the neighboring correct matches must be consistent with the above transformation. On the contrary, the neighboring false matches will not be consistent on any local transformation or global one, they could be assumed as Gaussian outliers. Bearing this in mind, we can assign a measure of strength to each match by its neighbors, that gauges the correctness of this match. A correct match will gain strong support from its neighbors while a bad match, can only obtain weak support from its neighbors because they are not consistent to any transformation.

Consequently, the energy function is simply the sum of the strength of matches and is minimized by gradient descent: in each iteration, the weakest group of matches are eliminated, so the associated match strength is deducted from our energy function. Also, their weak but positive support to strong matches is removed. Because all supports are positive, this removal results in the decreasing of value of the strength of match function. As a result, the energy function is monotonically decreasing over each iteration, and reaches its minimum when there is no match to be eliminated. The minimum could be a local one and the remaining matches may still contain false matches. However, the percentage of false matches has been greatly reduced and can be detected easily by any robust algorithm. The details is provided later.

Our strength of match is defined as the measure of the accuracy of the estimate of the local 2D affine transformation. As mentioned in section 5.2, for a given pair of matching candidate $(\mathbf{m}_i, \mathbf{n}_j)$, if it has $K$ neighbor matching candidates, their corresponding parameters of scale and rotation transformation can be calculated

and described as follows:

$$\mathbf{B} \approx \begin{bmatrix} s_1 & s_2 & \dots & s_K \\ \alpha_1 & \alpha_2 & \dots & \alpha_K \end{bmatrix} \tag{5.11}$$

Since not all of the matching candidates are correct matches, the affine transformations computed based on them cannot be same. In other words, the scale and rotation changes based on wrong matches will be outliers, and should be discarded or assigned with a low weight. The problem here is to find a way to detect outliers and discard them?

Our idea here is that we do not make a crisp distinction for inlier and outlier, but instead, we weigh its membership function according to its distance from the correct transformation. So now the critical problem is: what is the correct local transformation?

Assuming that the total number of outliers is not more than 50 percent, the median of scale and angle changes are taken as the correct changes. Then the difference between each change on scale and angle and their medians is computed as error distances. When the assumption of affine approximation is valid, we have the heuristic condition that the smaller the difference, the stronger the support of the neighbors to the given matching candidate. Furthermore, to compensate the difference between quantities of scale and angle changes, the square differences of both scale and angle are normalized to 1, and different weights are given.

We have assumed that the small patch around a corner is continuous and can be described by affine transformation. However, this assumption is only valid when the patch is small enough, that is to say, only a small enough patch can be assumed as local planar, and hence the transformation between a pair of patches is affine. As a result, the approximated transformation will perform better when the neighbor

matching candidate is close than it does when the neighbor candidate is far. Consequently, we will assign more weight for the closer neighbor candidates than for the further ones. But if the neighboring candidate match is too close, instability arises. So, our algorithm will not consider any candidate matches that are closer than a certain distance $r$.

Then, for the $g_{th}$ matching candidate $(\mathbf{m}_i, \mathbf{n}_j)$ which have $K$ neighbor matching candidates, the strength of match between them $S_M(\mathbf{m}_i, \mathbf{n}_j)$ is defined as:

$$S_M(\mathbf{m}_i, \mathbf{n}_j) = \frac{1}{K}\sum_{k=1}^{K}[10 * 2^{-0.1D_k} * e^{-[\gamma Es_k/Es_N + (1-\gamma)E\alpha_k/E\alpha_N]}] \quad (5.12)$$

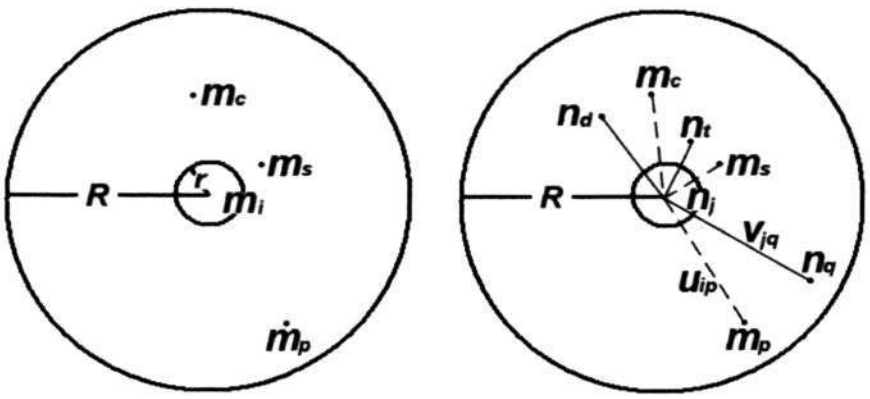



Figure 5.2: Energy Function Illustration

In Fig. 5.2, three neighbor candidates of corner $m_i$ are considered for illustration for the current matching candidate $(\mathbf{m}_i, \mathbf{n}_j)$. Referring to this figure, the details of Eq. 5.12 are as follows. $k \in [1, K]$ is the $k_{th}$ neighboring matching candidate of $(\mathbf{m}_i, \mathbf{n}_j)$, the $D_k$ is the average 2-norm of the two corresponding vectors, refer to Fig. 5.1:

$$D_k = \frac{\|\mathbf{u}_{ip}\| + \|\mathbf{v}_{jq}\|}{2} \quad (5.13)$$

and this average distance is integrated into the function as the $-0.1D_k$ power of 2, which is strictly monotonically decreasing function of it. $D_k$ is used to describe the distance from the current matching candidate $(\mathbf{m}_i, \mathbf{n}_j)$ to its neighbor candidate $(\mathbf{m}_p, \mathbf{n}_q)$ . Larger weights are assigned according to smaller $D_k$ because the local object surface can be approximated more efficiently as planar when $D_k$ is getting small.

The factor of 10 on the right term of Eq. 5.12 is introduced in order to prevent large machine error in case of the value of strength of match is too small. The $E\alpha_K$ and $Es_K$ which for normalizing the distances is the 2-norm of $E\alpha_k$ and $Es_k$, where $k \in [1, K]$ respectively:

$$E\alpha_K = [\sum_{k=1}^{K} E\alpha_k^2]^{\frac{1}{2}} \tag{5.14}$$

$$Es_K = [\sum_{k=1}^{K} Es_k^2]^{\frac{1}{2}} \tag{5.15}$$

where $E\alpha_k$ is the square difference from $\alpha_{median}$:

$$E\alpha_k = (\alpha_k - \alpha_{median})^2 \tag{5.16}$$

and $Es_k$ is the square difference from $s_{median}$:

$$Es_k = (s_k - s_{median})^2 \tag{5.17}$$

The $\alpha_{median}$ and $s_{median}$ are the median of $\alpha_k$ and $s_k$, where $k \in [1, K]$. After the normalization, we have $Es_k/Es_N \in [0, 1]$ and $E\alpha_k/E\alpha_N \in [0, 1]$ so that neither can dominate in their sum but will follow different weight $\gamma$ we assigned. $\gamma$ is the weightage between scale transformation and rotation transformation. In practice, we assign it 0.4 to emphasize rotation a little more.

An implicit factor in Eq. 5.12 is the assignment of the size of neighborhood area. Because we already assigned high weight for near neighbor matching candidates and low weight for far ones, we define our neighborhood as the following if we take the first image the kernel:

$$R(\mathbf{m}) = \mathbf{u} \qquad \|\mathbf{u}\|_2 \in [r, R] \tag{5.18}$$

$R$ is determined by the validity of affine assumption, where the depth variation on the area of radius $R(\mathbf{m})$ is much less than the distance from the corresponding object surface to the camera. Unfortunately the depth variation is unknown in this context. Therefore, $R(\mathbf{m})$ is empirically set to 1/10 of the image width by experience for a camera with normal lens, e.g. not wide-angle or macro lens. The quantity $r$ is necessary to prevent large errors: suppose $\mathbf{u}_{ip}$ is free of noise but $\mathbf{v}_{jq}$ has an absolute error of $\triangle\mathbf{v}$, then the $\triangle s/s$ will be increasing when both $\|\mathbf{u}_{ip}\|_2$ and $\|\mathbf{v}_{jq}\|_2$ are decreasing. In other words, when the neighbor matching candidate is getting closer to the current matching candidate, the relative error of scale and angle changes will be larger. Therefore, we set $r$ to 5 pixels length on the assumption that $\| \triangle \mathbf{v}\|_2$ is not more than 0.5 pixel.

By defining the squared difference based on the median of local transformations, our function can integrate the affine approximation assumption effectively. The smaller the squared difference, the stronger the contribution of the neighboring matching candidate to the current one, and hence we extend the work in [58] by taking translation, scale and rotation transformations when matching pairs of points into account.

## 5.4 Results and Analysis

In this section, we designed a control experiment to compare the performance of our method with other ones. Three types of corner detectors, Plessey, SUSAN, Imp-

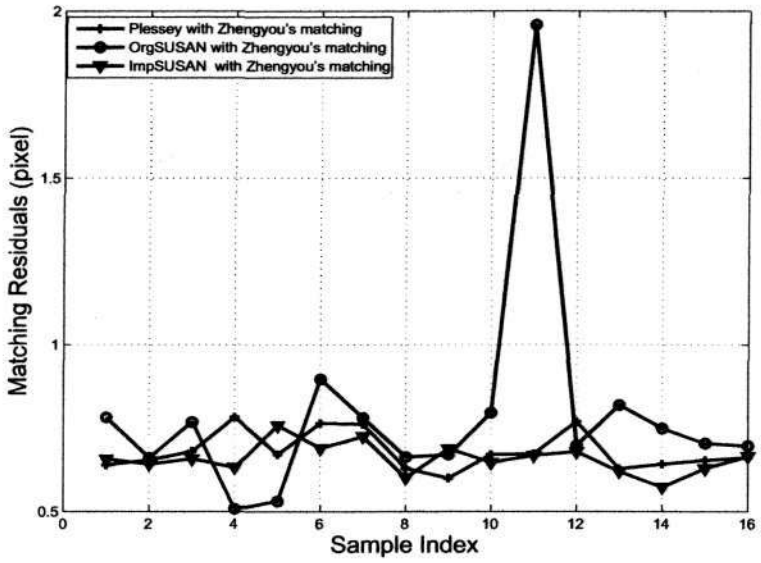



Figure 5.3: Three Corner Detectors each applied to Zhengyou's Matching Strategy.

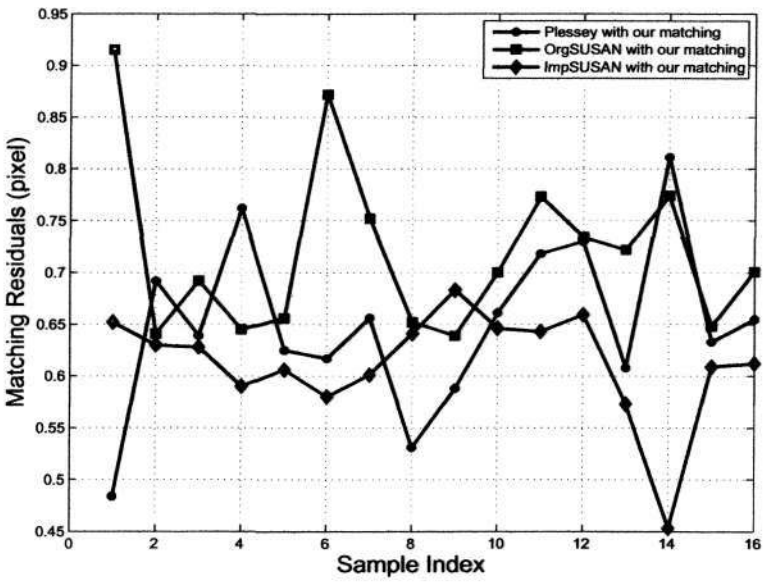



Figure 5.4: Three Corner Detectors each applied to Our Matching Strategy.

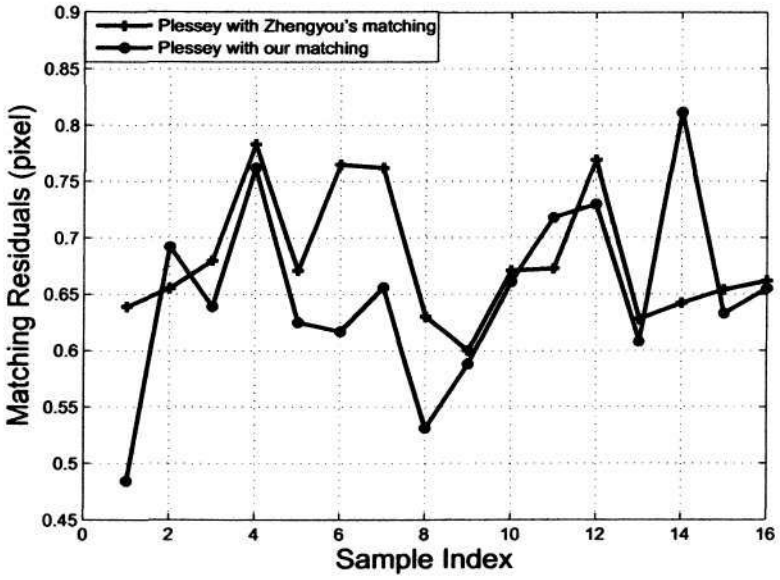



Figure 5.5: Plessey Corner Detector applied to each of the two Matching Strategies.

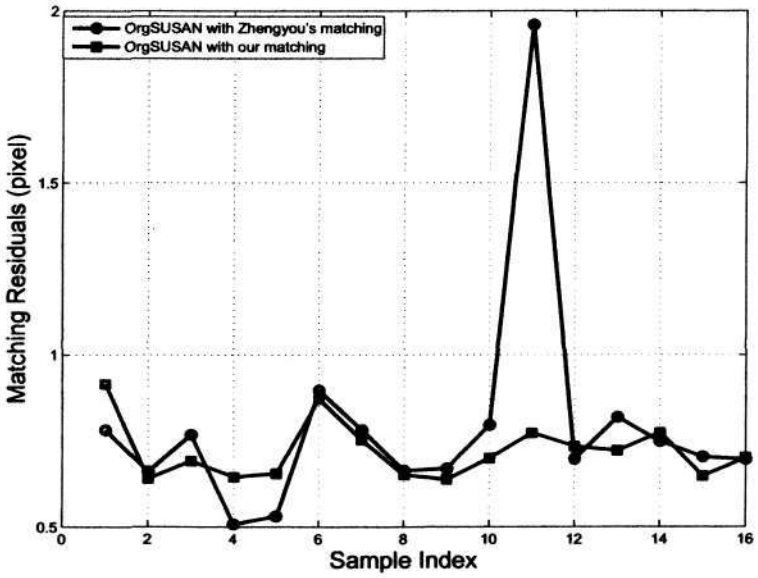



Figure 5.6: Susan Corner Detector applied to each of the two Matching Strategies.

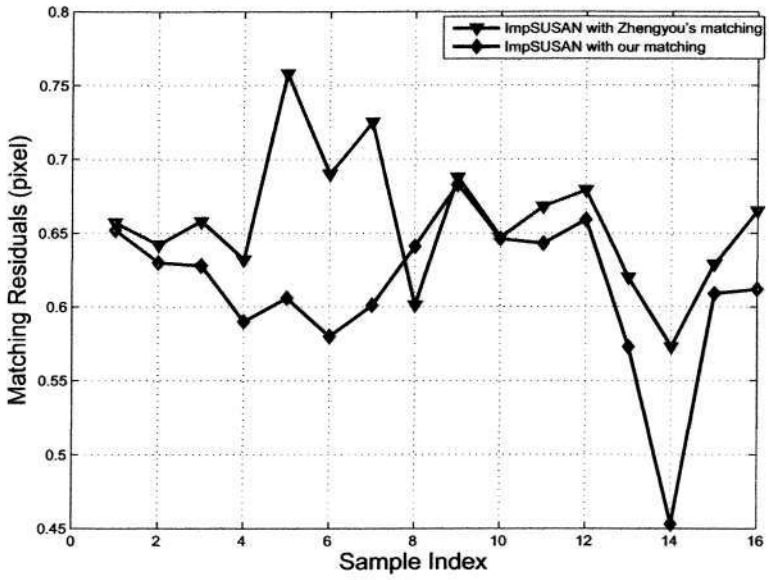



Figure 5.7: ImpSUSAN Corner Detector applied to each of the two Matching Strategies.

SUSAN and two matching strategies, our method and Zhengyou's are implemented for comparison. The control experiment is made by combinations of each corner detector with each of the two matching strategies. That means the only difference between different methods are the target of our comparison, and the other experiment conditions will be the same. We then compute the epipolar geometry between images, and the epipolar lines. In order to quantitatively measure the results, we define the residual error as in Eq. 4.4. The error is then computed by averaging the sum of distances. Finally, a simple analysis of computational complexity compared with former algorithm are given.

### 5.4.1 Comparison 1: different corner detectors with the same matching strategy

To compare the results from different algorithms fairly, we designed a controlled experiment that use the same matching strategy but with different corner detectors on the same set of images. Then only the corner detector is different from each set of experiments. The algorithms are run on sixteen pairs of images from the wide baseline database of ALOI [128], and we use different corner detectors to detect point features and the same matching strategy to find the final matches between images. Fig. 5.3 shows the results of different combinations of three corner detectors with Zhengyou's matching strategy, while Fig.5.4 illustrates the results of different combinations of the three corner detectors with our matching strategy. In Fig. 5.3, we can see that for 81.25% or 13 samples, the ImpSUSAN gave best results or almost indistinguishable from the best one out of the three. On the other hand, in Fig. 5.4, the ImpSUSAN performs best in 87.5% or 14 samples. In addition, the ImpSUSAN gave an overall stable performance.

Another point of view is to study the statistical meaning of the above data. In Table 5.1 and Table 5.1 the mean and standard deviation of different combinations are given. Our methods produce the smallest average errors with a relative small standard deviation.

Table 5.1: The mean and standard deviation of matching results based on the combination of 3 corner detectors and Zhengyou's matching method respectively

| | plessey+Zhengyou's | OrgSusan+Zhengyou's | ImpSusan+Zhengyou's |
|---|---|---|---|
| mean | 0.680 | 0.793 | 0.658 |
| Std | 0.057 | 0.327 | 0.045 |

Table 5.2: The mean and standard deviation of matching results based on combination of 3 corner detectors and our matching method respectively

| | plessey+Ours | OrgSusan+Ours | ImpSusan+Ours |
|---|---|---|---|
| mean | 0.651 | 0.720 | 0.613 |
| Std | 0.082 | 0.082 | 0.052 |

### 5.4.2 Comparison 2: different matching strategies with the same corner detector

To demonstrate the performance of our matching strategy, another comparison is made on the same data sets as in last one, and the results are shown in Fig. 5.5, 5.6 and 5.7. In this experiment, the two matching strategies are run based on the same corner detector: Plessey, OrgSUSAN or ImpSUSAN corner detector, respectively. For the Plessey corner detector in Fig. 5.5, we can see that our strategy performs better on 75%, or 12 out of 16 samples, and our matching strategy is more stable. For OrgSUSAN corner detector in Fig. 5.6, our strategy can produce better results on 72.2%, or 11 out of 16 samples, and the error differences are almost indistinguishable on the other five samples. The same conclusion can be derived from Fig. 5.7, where our strategy win on 87.5% or 14 samples, and produce almost the same results on the other two sets.

The first set of Epipolar line plots are given in the Fig. 5.9 to Fig. 5.11, which generated by our matching method following three corner detectors. In Fig. 5.8 the Epipolar line is plotted by manually selected matches, which we take as a good reference for the ground truth.

### 5.4.3 Comparison 3: computational complexity

The exact overall computational complexity of the whole system is hard to obtain because there are worst and best case in different stages. However, we can decompose

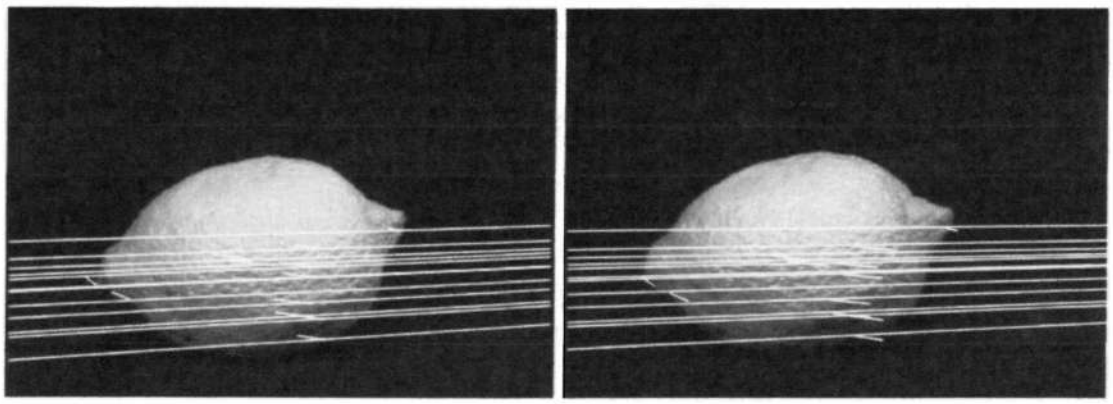


Figure 5.8: The Results by manually selected matches

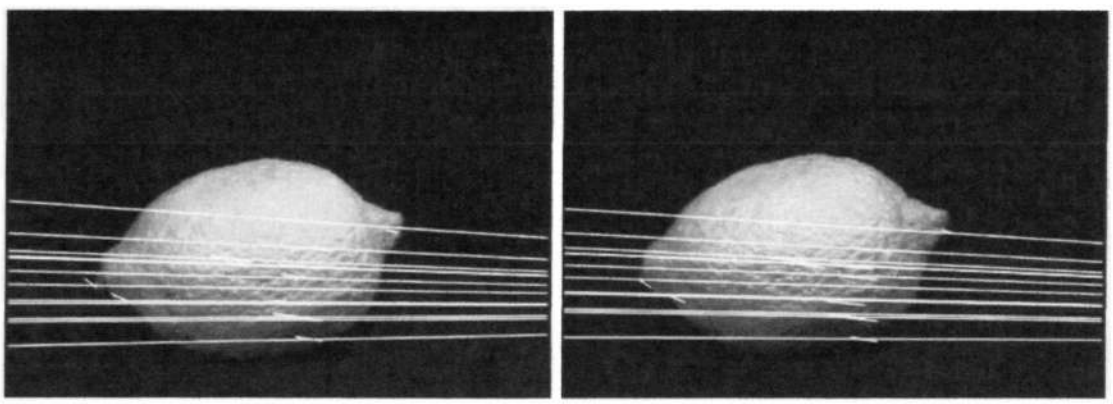


Figure 5.9: The Results by SUSAN and My feature matching

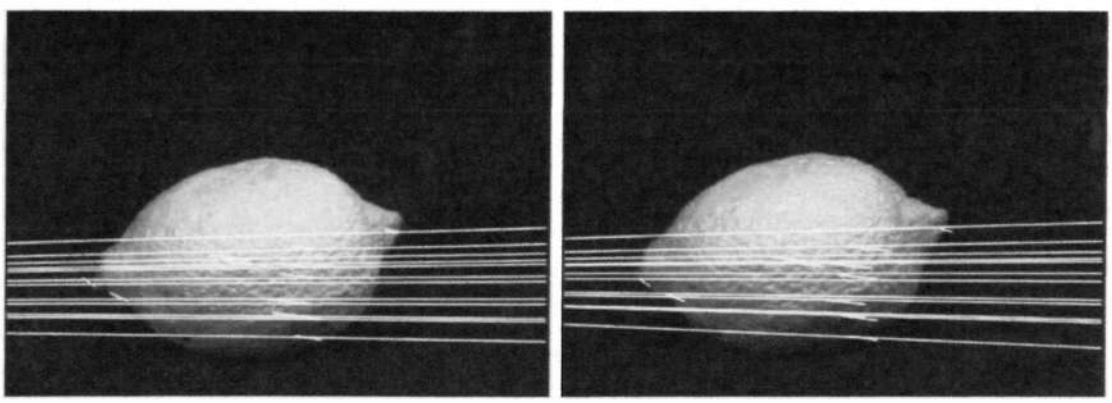


Figure 5.10: The Results by ImpSUSAN and My feature matching

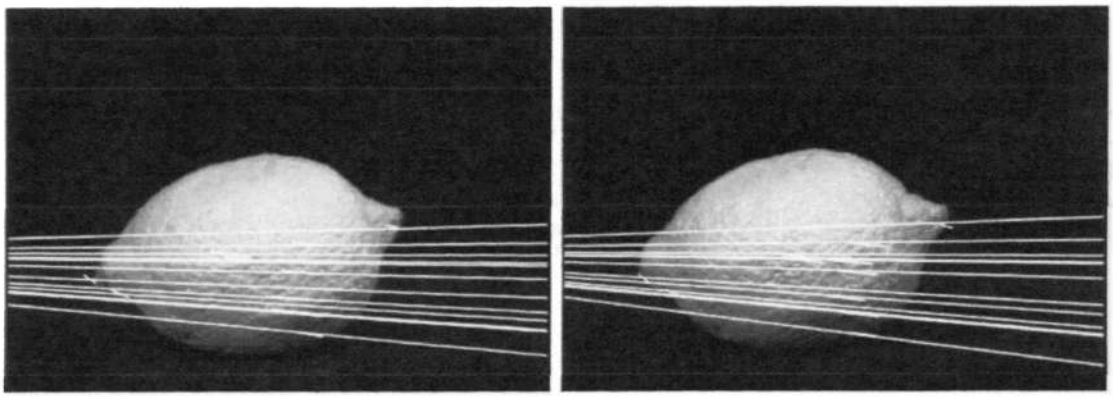


Figure 5.11: The Results by Plessey and My feature matching

it into four parts that have significant computational complexity, and then, a peer to peer comparison with method in [58] is given based on O(.) analysis:

• Corner detection: Plessey is $O(l_1WH)$ while ImpSUSAN is $O(k_1WH)$, where $W$ represents the image width and $H$ represent the image height in pixels. In Plessey, $l_1$ is the trial times which we assume to be 3∼5 times. On the other hand, in ImpSUSAN, $k_1$ is less than 1/10 of $l_1$. This is because the local curvature analysis will only be applied on selected points whereas the the computational load of SUSAN operation is far less than Plessey's.

• Correlation: our system is $O(PQ)$, which is the same as in [58]. Here $P$ is the corner quantity and $Q$ is the average corner quantity in local search window.

• Relaxation: Our energy function is $O(k_2MKk_3)$ whereas Zhengyou's method is $O(l_2MKl_3)$. $M$ is the matching candidate quantity, $K$ is the average neighbor candidate quantity, and $k_2$, $l_2$ are the iteration times respectively. In addition, $k_3$, $l_3$ are the respective prime operations for each strength of matches, On average $k_3 \approx 1.5l_3$, so the overall performance on relaxation is comparable.

• LMedS for fundamental matrix: The computational complexity is $O(N)$ if taking solving linear equations as prime operations. However, suppose $n$ is the actual time

of solving linear equations, $n$ will be:

$$n = \log^{(1-p)}_{[1-(1-\varepsilon)^8]} \tag{5.19}$$

where $p$ is the confidence level, $\varepsilon$ is the percentage of false matches after relaxation process. If we suppose $\varepsilon$ is 15% due to our special process, and 45% in normal case, the actual computation times of special v.s. normal will be 1:38.

In summary, our system makes an automatic searching in the corner detection phase and takes in more information in relaxation phase, which incurs more computational complexity. In the other hand, we also successfully reduce the actual computational complexity in the robust algorithm stage, where the prime operation is solving the linear equations. The reduction of computation is accomplished by carefully finding the corners and initial matches. In practice, the overall computation time are similar, and our method has less user interaction and better results.

## 5.5 Conclusion

In this chapter, we proposed a new energy function which approximates the local transformation more precisely than the one of classic method in [58]. The incorporation of more information is leading to a better result on the final corresponding results. Thus our hypothesis that taking into consideration rotation between two planar patches is significant and verified the improvement over the original energy function proposed by [58]. In addition, our matching method is actually cooperating with our ImpSUSAN corner detector in last chapter. By jointly considering the corner detection and matching, we improve the overall performance in terms of accuracy. On computational complexity side, although we paid some price on detecting robust and accurate corners and exploring more information in matching,

this is more than compensated for during the robust LMedS computation, which is nonlinear to the sample size. The final results confirmed our expectation.

# Chapter 6

# Reconstruction by 3D Adaptive Window Correlation and Layered Depth Image

**Homologous points** are defined as the images points from different cameras or frames that originate from the same 3D scene point. And the **corresponding points** are the estimation of homologous points. An image of a 3D scene is two dimensional where the third dimension is lost during image capturing. This capturing itself can be described by projective transformation. The 3D reconstruction from multiple images is based on the establishment of correspondence between homologous image points. The key issue, known as the correspondence problem, is to find the points from different images that correspond to the same 3D space point. If two images are captured from the exact same viewpoint and camera, then they can be perfectly matched by just overlapping them. However, none of the third dimension information can be extracted because their disparity is zero. To recover the third dimension, images must be taken at different poses, or different relative positions of camera to object.

The change of pose is an obstacle that makes the correspondence problem very difficult to solve due to perspective distortion, occluding boundary, occlusion and illumination changes. Many matching methods have been proposed to resolve the difficulties of corresponding problem and reconstruct the 3D object [91] [67]. However, due to the ill-posed nature recovering the unknown third dimension, the corresponding/reconstruction problem is still open for research.

A matching method must be formulated as to search for the homologous points across image and produce corresponding points as the estimation of homologous points.

As discussed in chapter 3, the corresponding problem is normally resolved in the 2D image space. A 2D similarity measure between points from respective image is calculated over a local window with the same size. However, the methods based on this principle can hardly cope with the variations due to viewpoint change because the corresponding 3D patch of the local window can be arbitrary shape. In this chapter, we will analyse and address this shortcomings of 2D correlation window will be analyzed in detail. We then propose a **3D correlation window**, which is a square plane in 3D space with a given size and orientation.

Then with this 3D correlation window, we define a scheme that integrates this adaptive 3D window and layered depth image to resolve the correspondence and reconstruction problem. We start from a correlation-based 3D window with fixed size and orientation to obtain coarse reconstruction. Our 3D correlation window can then be adapted to the coarse reconstructed shape both in size and orientation, and then our proposed algorithm will further refine it to the final results. The layered depth image is used to track the visibility of target 3D point to eliminate the effects of occlusion. Our cooperative method can deal with perspective distortion, occlusion and pose variation but it assumes that the object surface is Lambertian. It reconstructs the 3D object surface and establishes correspondence simultaneously.

## 6.1 Analysis on the 2D Correlation Window

### 6.1.1 The limitation of 2D Image Correlation

Finding correspondence between images is as mentioned the key problem in 3D reconstruction from multiple images. The problem can be generally described as: for an unknown surface point in 3D space, finding its respective image points on each image from where the surface point is visible. To check whether two points from their respective images correspond to each other, similarity measures such as ZNCC (Zero-mean Normalized Cross Correlation) or SSD (Sum of Squared Differences) have to be calculated over their respective neighborhood, generally in a fixed size square window. This is normally called area-based method. If the signals in neighborhood of corresponding points are the same and unique in different images, the above method will find the correct correspondence. However, the similarity measure depends on not only the signal's inherent characteristics such as variation and sampling order, but also the window that contains the signal. This leads to the following dilemma.

On one hand, finding the correct correspondence requires that the similarity measure has sharp extremum in the searching space, which in turn demands that the local signal is unique and robust to noise. For this, the signal variation and length must be large enough, in other words, the sampling window have to be large. On the other hand, the usage of 2D window is based on assumption that the object surface is pair-wise smooth and can be approximated efficiently by a small plane. Then, the contents in respective windows will correspond to the same part of object surface. If the 2D window for computing similarity is large, the above assumption will be invalid. Furthermore, for a boundary between different depths, a small window size will locate the boundary more accurate than a large one. Then, a square window with fixed size will not satisfy the arbitrary situations globally in practice, the

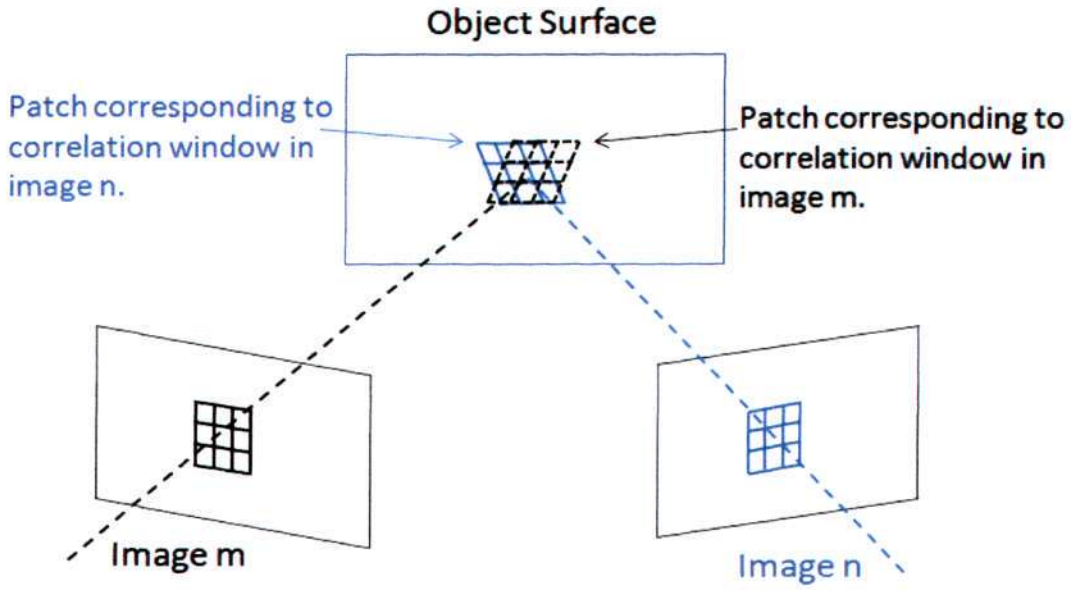



Figure 6.1: Error illustration when image windows of fixed shape and size are matched

adaptive tradeoff of window size is needed.

Some significant work have been proposed to use adaptive 2D correlation window in [67]. These methods gave improvement over the performance of 2D method. But even if corresponding windows have the optimum adapted window size, they do not in general project back onto the same corresponding 3D surface patch because of the perspective distortion and pose variation. The larger the window size is, the greater will be the 3D surface patch mismatch. So here lies the dilemma. For larger window sizes, the similarity measure is less immune to noise but then suffers from the fixed window size problem just discussed. Conversely, the 3D surface mismatch will be reduced at the expense of noisy similarity measure when the correlation window size is small.

Referring to Fig. 6.1 and Fig. 6.2, we show the theoretical difference between 2D correlation window and 3D one. In Fig. 6.1, the similarity is calculated between

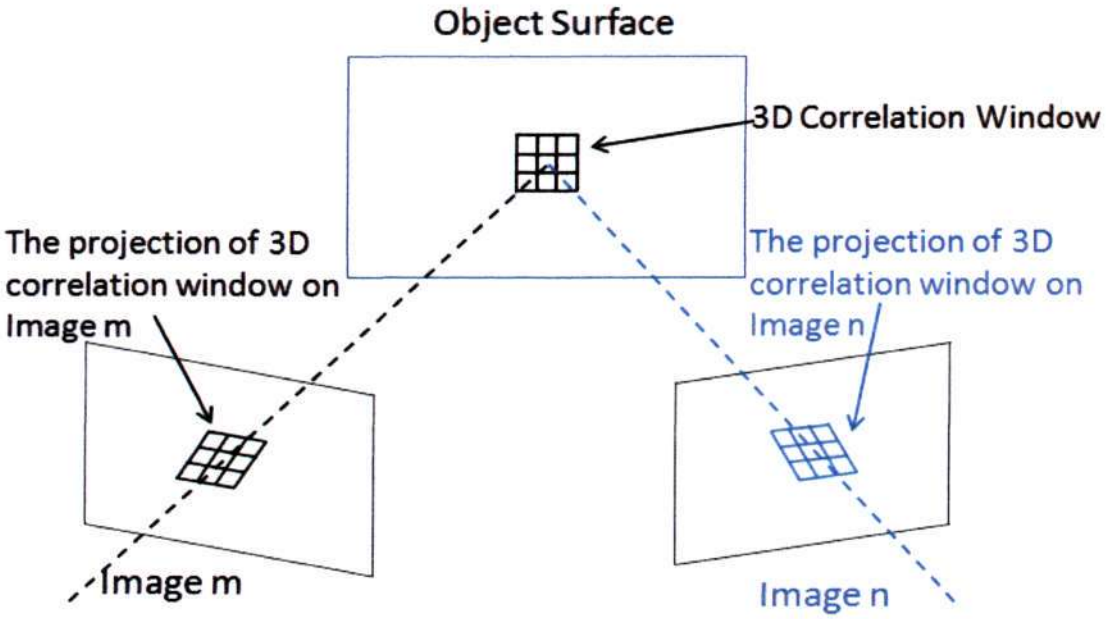



Figure 6.2: The illustration of 3D correlation window

2D windows from each image, and the conventional 2D windows are the same size rectangles. However, because of projective distortion and pose variation, the two windows are mostly representing different area on the object surface, although overlapped to some extent. Some similarity measurement, like correlation, is sensitive to the sampling order in sampling spaces.

On the contrary, the 3D window does not have this ambiguity. In Fig. 6.2, we can see the two projected areas on each image are corresponding to the exact same 3D window. Furthermore, the sampling order when calculating similarity is also exactly the same, say, the same order as the sampling on the 3D window. If the 3D window is a good approximation of the local object surface, then the similarity calculated based on its projections on each image will be higher than that in Fig. 6.1 and thereby more accurate.

We assume that the object surface patch is piece-wise smooth, that means the local

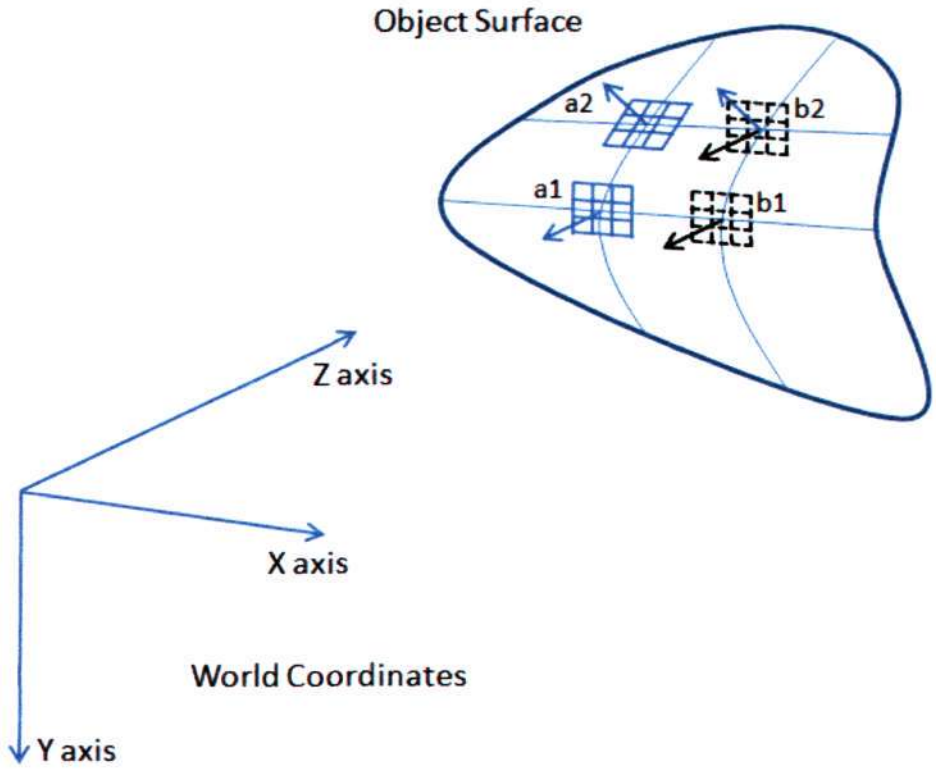



Figure 6.3: The illustration of 3D orientation adaptive correlation window

surface can be approximated quite closely by a small plane. We proceed further to align the orientation of our 3D correlation window with the object surface normal, and then the 3D window can be an effective approximation of the local surface. In addition, the size of the window will be adaptively determined by estimating the local curvature of object surface.

We will explain here why a fully adaptive window is desirable for the correct area-based matching paradigm. There are three significant attributes of the 3D correlation window: location, size and orientation/normal. To search for the true object surface in 3D space, a 3D correlation window must keep changing its location and similarity is measured at this location. We define a **3D fully adaptive window** as the one that both its size and orientation are adaptive. Also we define **3D**

**size-adaptive window** as the one with adaptive size only, its orientation is fixed. Furthermore, we define **3D non-adaptive window** as the 3D window that only location is changing. Because of better approximation of the object local surface, the correlation value applied to 3D full adaptive window will greater than that from 3D size adaptive window, which is in turn higher than the correlation value based on the 3D non-adaptive window. Apparently, the reconstruction algorithm that uses a correlation window that better approximates of the local object surface will produce better result.

The difference between 3D full adaptive window (in blue) and 3D non-adaptive window (in black) is illustrated in Fig. 6.3. Surface points $a1$ and $b1$ have the same surface normals, and the normals at surface points $a2$ and $b2$ are the same. The 3D windows at $a1$ and $a2$ are tangent planes representing the fully adaptive windows in which their normals can be oriented in any direction. The 3D windows at $b1$ and $b2$ are non-adaptive window which are always parallel to the xy-plane of world coordinates and with fixed size. We can observe that the 3D non-adaptive Window could be good approximation of the local surface under some conditions like situation $a1$ and $b1$, where the local surface normal is perpendicular to xy-plane. At the same time, we can observe that a piece-wise smooth object will always has surface normal parallel with Z-axis of world coordinates. However, the 3D window in situation $a2$ will be a better approximation of the local surface than that for situation $b2$. As the result, the correlation value based on window in $a2$ will be much higher than that based on situation $b2$, indicating that the 3D reconstruction based on 3D full adaptive window will produce better results.

Based on the above analysis, we can draw our conclusion that the 3D full adaptive window is preferred over 2D window in image when calculating similarity like correlation. To adapt the 3D window according to local surface characters, we need to know the object surface first. However, as our objective here is to reconstruct

the 3D object surface, we actually do not know the local normal or curvature of the surface. How then can we use the 3D full adaptive window, which is to adapt itself according to object surface information, to recover the 3D object surface? This problem will be addressed in next section.

### 6.1.2 The Layered Depth Image

The Layered Depth Image (LDI) [117] is an efficient data structure that we will adopt in our work. It can be used to track the visibility of voxels. They defined the voxel as the smallest unit of the discretized object surface. In their work, LDI is only used to track visibility when voxels are deleted. This means the voxel is no longer on the object surface. Also, LDI can keeps track of voxels being added to the current list as a voxel is added onto the surface. In Fig.6.4 we illustrate the cases when a voxel is added to or deleted from the current voxel list, and how LDI are updated.

Occlusion or occluding boundaries are very important factors to consider for 3D reconstruction work. It is very hard to know before hand whether a given part of object surface can be seen from a given image. The LDI gives a powerful tool to tracking whether a 3D point or voxel is visible from a given viewpoint, or the visibility.

## 6.2 3D Adaptive Window Correlation Based Reconstruction with LDI

In last section we discussed the advantage of 3D adaptive window in 3D reconstruction problem. Also, we believe LDI is a powerful tool for tracking the visibility of object surfaces. Unfortunately, both of them need to work on a given known surface,

but our objective here is to recover that surface. The solution of how to use them in 3D reconstruction is illustrated in this section.

### 6.2.1 The Definitions

#### Problem Formulation

The objective of 3D reconstruction is to recover the 3rd dimension information from 2D images. In another words, the 'new' information is extracted from the known. Based on this, we formulate our reconstruction problem as an information exploring problem. Before we establish the 3rd dimension information, we will obtain intermediate information based initially on the strictest constraints. This allows us to build a rudimentary surface as an initial information of the scene. Then we progressively relax our constraints based on the former intermediate results and obtain next level intermediate information. The relaxing of assumptions allow for greater flexibility in adaptation and thereby building a more accurate surface. This process is repeated until the convergence condition is met.

#### Basic Constraints

As mentioned, our algorithm will first build an initial rudimentary surface based on the strictest constraint. The we relax the constraints as the iterative process continues. Here, we define the three levels of constraints used. Our work is based on the assumption that the target object surface is continuous and piece-wise smooth. This assumption is valid for a lot of natural objects. The assumption can also be interpreted as the local object surface, if small enough around a given point, can be effectively approximated by a tangent plane located at this point. We call it our **first level constraint**. The size of tangent plane should be proportional to the local surface curvature. If the local curvature is high, then a small size of tangent plane will be a better approximation; if the local curvature is low, it is means the

local surface is flat, and hence a larger tangent plane can approximate the local surface better. In Fig.6.5 we illustrate the relationship of tangent plane size and local surface curvature.

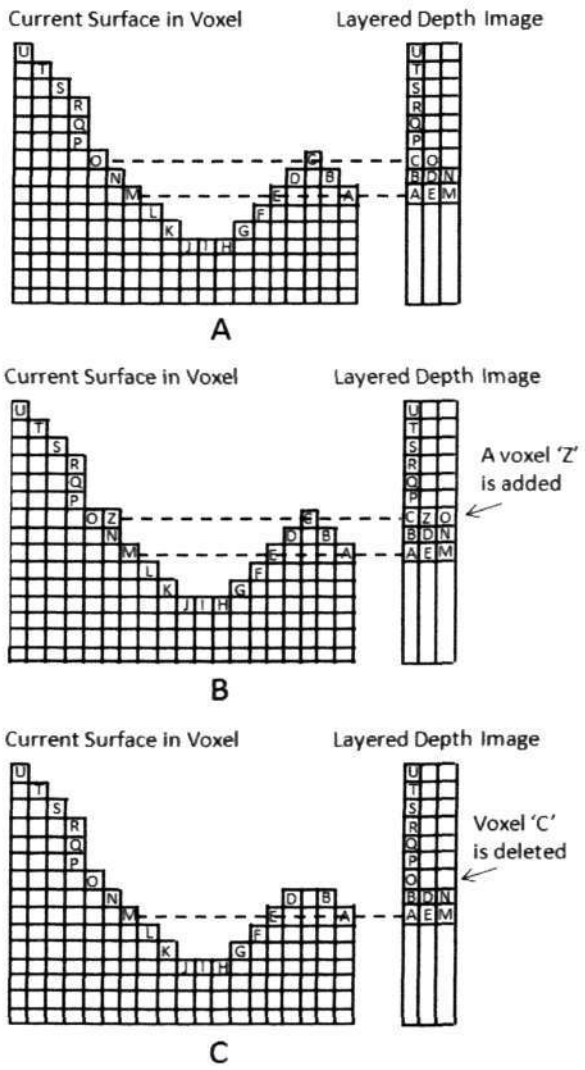



Figure 6.4: LDI: A) An initial surface profile; B) A voxel is added; C) a voxel is deleted.

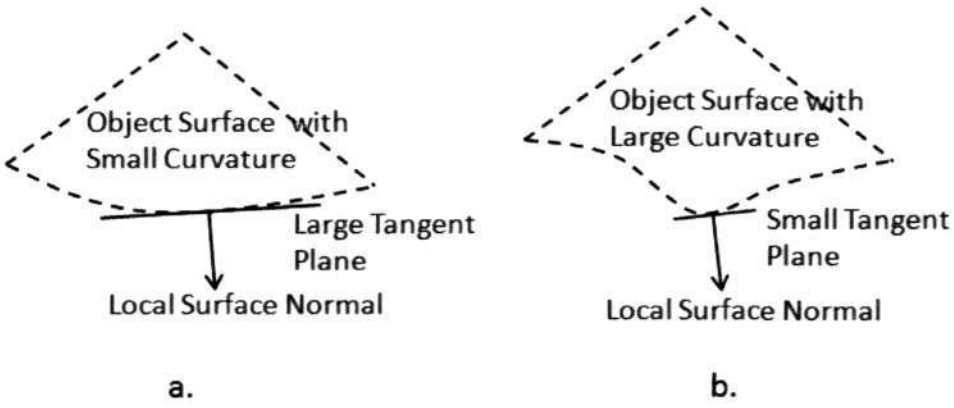



Figure 6.5: The first level constraint: a. small local curvature, large tangent plane; b. large local curvature, small tangent plane

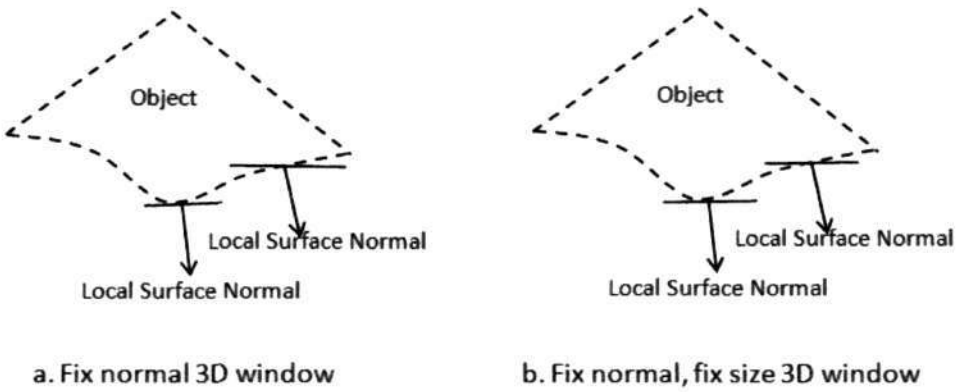
From the first level assumption, two stronger levels of condition are defined as showed in Fig.6.6. The **second level constraint**, for a continuous and piece-wise smooth object, is that its local surface can be approximated by a fix normal but adaptive size 3D window, the window size is adjusted according to the local surface curvature, and the window orientation is aligned with the Z-axis of world coordinates. It is shown in Fig. 6.6*a*.

The **third level constraint** is, for a continuous and piece-wise smooth object, its local surface can be approximated by a fixed normal and fixed size 3D window, the window orientation is aligned with the Z-axis of world coordinates. This assumption is shown in Fig.6.6*b*. The third level constraint is thus the strictest while the first is the least strict.

### Primitive Definition

**Workspace** is defined as a volume that contains the target object. The size and location of the volume can be decided by prior knowledge.

**Node** is defined as the sampling point inside of workspace. The workspace is dis-




a. Fix normal 3D window

b. Fix normal, fix size 3D window

Figure 6.6: Two Stronger Level of Constraints. a. the second level constraint; b. the third level constraint.

cretized into nodes and sampled $Xnum$ times along X-axis, $Ynum$ times along Y-axis and $Znum$ times along Z-axis. The workspace is discretized into nodes.

**Node surface** is defined as a node list that represent the 3D object surface.

### 6.2.2 The Proposed Method

In last section, we defined our three level of constraints. The first level constraint is a weak assumption so that the 3D reconstruction result based on it will be closer to the true object. However, an arbitrary correlation window that is tangent to the local object surface depends on the local surface information, such as local normal, local curvature and so on. This puts us in the dilemma that the reconstruction result is expected based on our first level constraint, but to take the advantage of this constraint we need the object surface information, which in turn, is our target to recover. Consequently, the straightforward application of our first level constraint on 3D reconstruction is not possible due to lack of any information.

**Coarse to fine strategy**

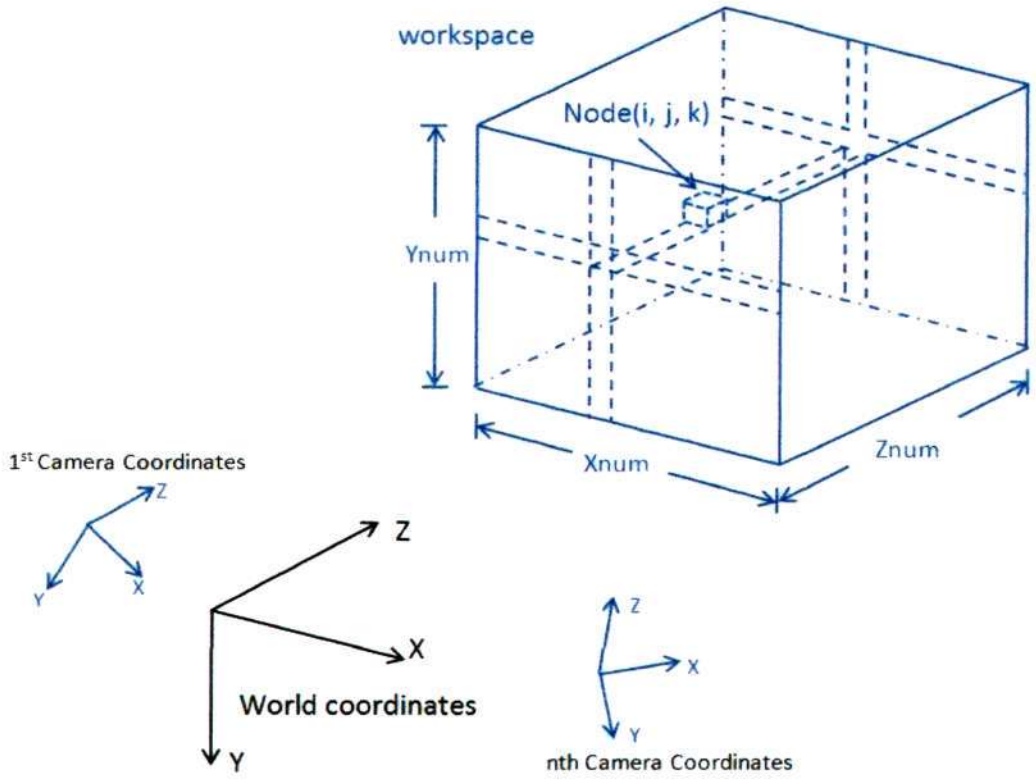



Figure 6.7: The Basic Definitions

We formulate the 3D reconstruction based on multiple images as an information exploration problem. That is, we start with a strong constraint, that is, our third level constraint, to establish a coarse recovery of the 3D object surface first. Then, together with the known coarse 3D shape, we relax the strong constraint by a weaker one, that being the second level constraint, to refine the former coarse surface. Again, with the refined 3D surface, we can further relax the weak constraint to a weaker one, our first level constraint, and further refine the recovered surface.

At the same time, the same dilemma also exists in dealing with pose variation and occlusion problem. With the 3D surface information found, we still have not determined the occluding areas or the changes due to pose variation. Even further, the

projective distortion can not be addressed either. All of the above factors depend on our recovering the target, the 3D surface information. Fortunately, the same coarse to fine strategy is also applicable here, before finding rough surface information, we do not need to incorporate so many factors into our reconstruction algorithm. As such, LDI is applied for occlusion tracking only after a coarse 3D surface has been established.

**Principle of the Algorithm**

Suppose the 3D workspace and camera parameters are given, either by other algorithms or manually. The workspace is then discretized into nodes as $Xnum$, $Ynum$ and $Znum$. Also, suppose our cameras are at ordinary positions [114]. Our reconstruction algorithm will start from the frontal face of workspace, which is close to the cameras, and searching away from the cameras for the object surface. The frontal surface is taken as the initial current node surface.

Without knowing anything about the object surface other than the input images, we start with our third level constraint. A fixed size, fixed normal 3D window is put on each node and sampled. The projection of this 3D window in each images are sampled in the same order as the 3D window's and correlation is calculated between images. The average of the correlations are taken as the correlation value of the current node. After each node is processed, we build up the first current node surface by selecting the node with the largest correlation value along Z-axis. The Z index of the current node surface is stored in matrix, $CurSurf$ with dimensions of $Xnum$ and $Ynum$. Simultaneously, a searching range for each element of $CurSurf$ is also extracted for the first and last node position of which the correlation value is greater than 60% of the correlation value of the corresponding node in $CurSurf$. Then, for each element of $CurSurf$, we have a searching range, which is much less than $Znum$. Fig. 6.8 shows the flow chart of our overall idea algorithm.

After the first node surface is recovered, we start the LDIs to track the visibility.

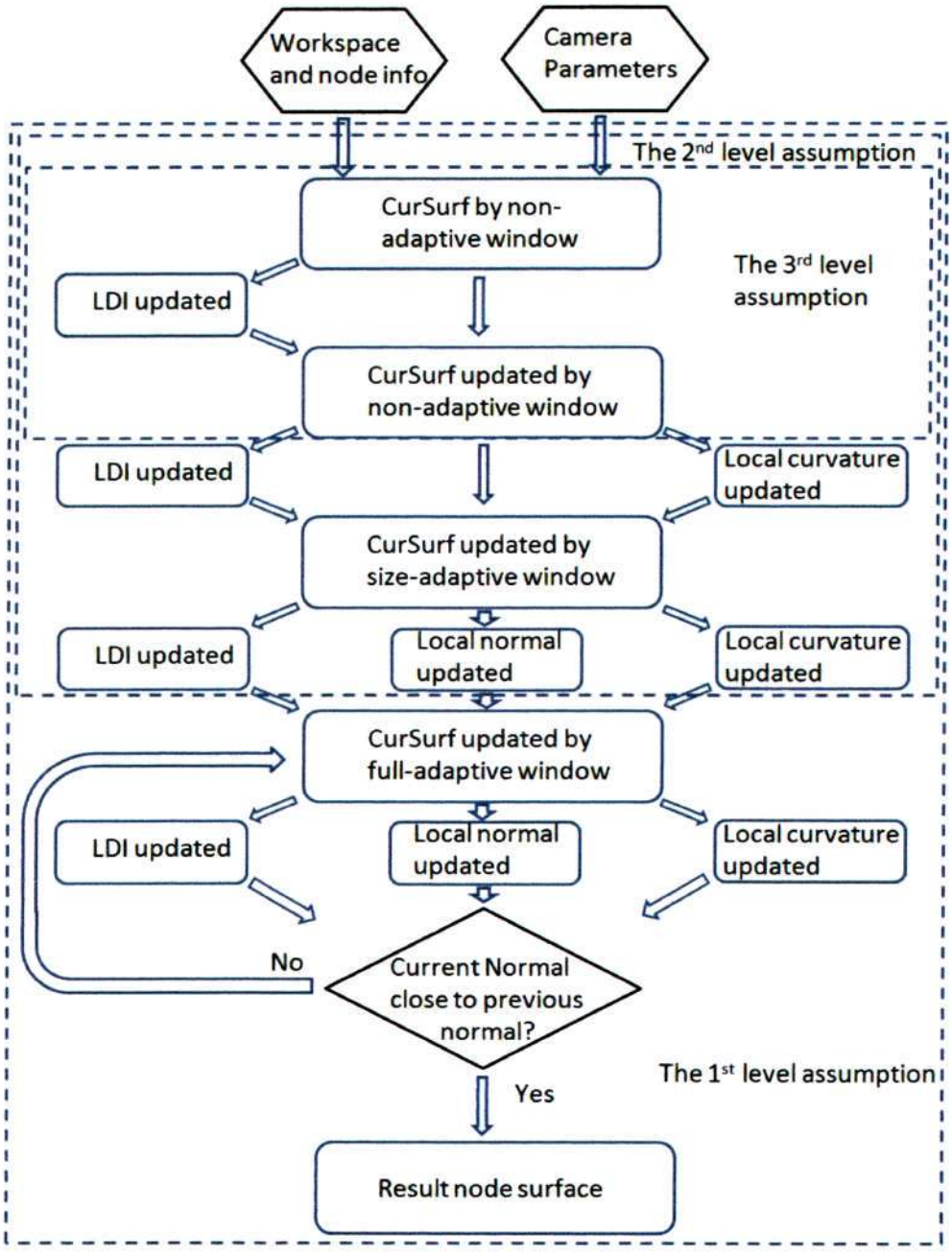



Figure 6.8: The flow chart of proposed concept

Then, the above calculation is repeated in the searching range but now with some knowledge of whether a given node is visible from an image currently. The current

node surface is then updated.

The node surface based on the third level assumption will apparently not be very accurate. It did not take occlusions into account, and the fixed normal, fixed size window is not a good approximation either. However, it still reveals a lot of information to us. For example, if the object local surface normal is aligned with the 3D window normal, and the local surface is not occluded, then this area will be correctly recovered as indicated by high correlation values. So our focus next will be on the areas that the third level assumption are not applicable.

The local curvature is measured on smoothed current node surface, and the 3D window size is then adapted according to the local curvature. A large window size will be applied on the areas with greater curvature, and a smaller one will be suitable for high curvature area. This will be fulfilled under our second level constraint instead of the third level constraint. The nodes in the searching range are then re-calculated for correlations as done above. The $CurSurf$ can then be updated by the new nodes with maximum correlation value over the searching range.

From the current node surface, the visibility tracking is implemented by updating LDI of each image. Knowing the current visibility state, and the updated local curvature measure, the nodes between searching range are re-calculated for correlation again with the new 3D window sizes. The current node surface, $CurSurf$ matrix, is then updated by the nodes with maximum correlation values. And the LDIs are also updated by the $CurSurf$.

Finally, the second level constraint is relaxed again to the first level constraint. Based on the current node surface, the normal at each node can be computed. We then determine adaptively the size of 3D window by local curvature measure and align our 3D window normal to that of the surface local normal, and again, re-calculate the correlation value based on the adapted 3D window. The current node surface and also LDIs, local curvature and local normal are then updated.

The updated local normal is then compared with the former local normal at the same index of $CurSurf$. The nodes that the angle between two normals are greater than 15 degree will be re-calculated with the current normal and current LDIs. The process is repeated until the updated normal is within 15 degree of the former normals for all nodes.

### 6.2.3 Convergence Analysis

The proposed method based on coarse to fine strategy is clearly reasonable. However, it is still not clear whether the algorithm will converge to the true object surface. A formal proof will be very hard to accomplish because of the arbitrary form of 3D object surface. Instead, we provide an explanation for the conditions necessary for convergence of our algorithm.

To establish correspondence across images and reconstruct 3D object surface, we need to deal with the following obstacles:

*a*. Pose variation.

*b*. Projective distortion.

*c*. Occlusion or occluding boundary.

*d*. Illumination variation.

*e*. False match due to image attributes.

*f*. Correlation window is far different from the real local surface.

*g*. other variations, e.g. quantization error.

The above obstacles can be seen as the reasons that we can not accurately reconstruct the 3D scene. Obstacle $d$ is simplified by assuming Lambertion surface. Due to the theory of multi-baseline stereo [65], obstacle $e$ can be minimized by use of

multiple images. Obstacle $g$ is out of our discussion and only have minor impact. Our focus is on obstacles $a$, $b$, $c$ and $f$.

Whether our method will converge to the true object surface is firstly depending on whether the rough surface reconstructed under the third level assumption is reasonable or not. When a 3D window, adaptive or not, is applied, we can always split the object surface into two types, $Area_c$ that the 3D window can be good approximation of local object surface, and $Area_{nc}$ that the 3D window is not good approximation of local object surface, for example, the normal of local surface is far from the normal of 3D window.

Furthermore, we can categorize the object surface as with and without occlusion. In Fig. 6.9, the possible results from application of fixed size and fixed normal window to local surface without occlusion are shown. Three primary local structures, flat, convex and concave, are showed by 2D illustrations. After the application of 3D windows with fixed normal and fixed size, we can have a heuristic expectation as when the 3D window fit the local surface, the surface points in this area will be successfully recovered with high accuracy; when the 3D window normal is far from the local surface normal, the surface points will then be recovered with high ambiguity. The blue error bound curve, plus and minus, showed this understanding. It means the result by correlation based on 3D window with fix size and normal will be at any point in between the positive and negative error bound.

The cases of surfaces with occlusions is shown in Fig.6.10. Our basic assumption which supposes the object surface is continuous and pair-wire smooth, can be deduced in the following way. That is, no matter where the world coordinates are, there will always be local surface whose normal is aligned with the Z-axis of world coordinates, and is the frontal and back extremum along Z-axis. This is the manifestation of the mean value theorem of two dimension functions. In our 3D reconstruction context, these areas of surface can always be recovered accurately under the third

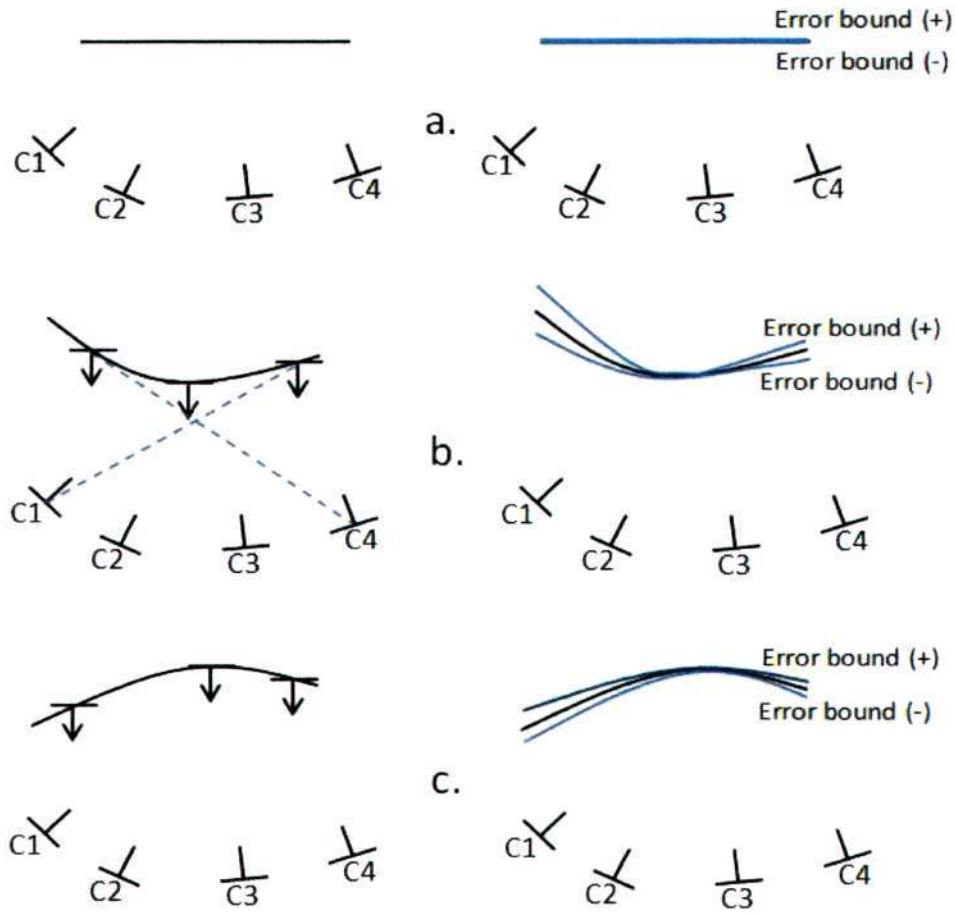



Figure 6.9: The Fix window application without occlusion

level assumption because the frontal extremum will not be occluded.

Fig. 6.10 shows the cases that the angle between 3D window normal and local surface normal is large, e.g. greater than 90 degrees. Although the 3D window is not a good approximation to the local surface, the rough surface normal can still be closer to the real surface normal than the normal of 3D fixed normal window if only we can correctly recover the area without occlusion.

Based on the above analysis, we can conclude that under our third level assumption, the recovered rough surface is reasonable but with large error on the low

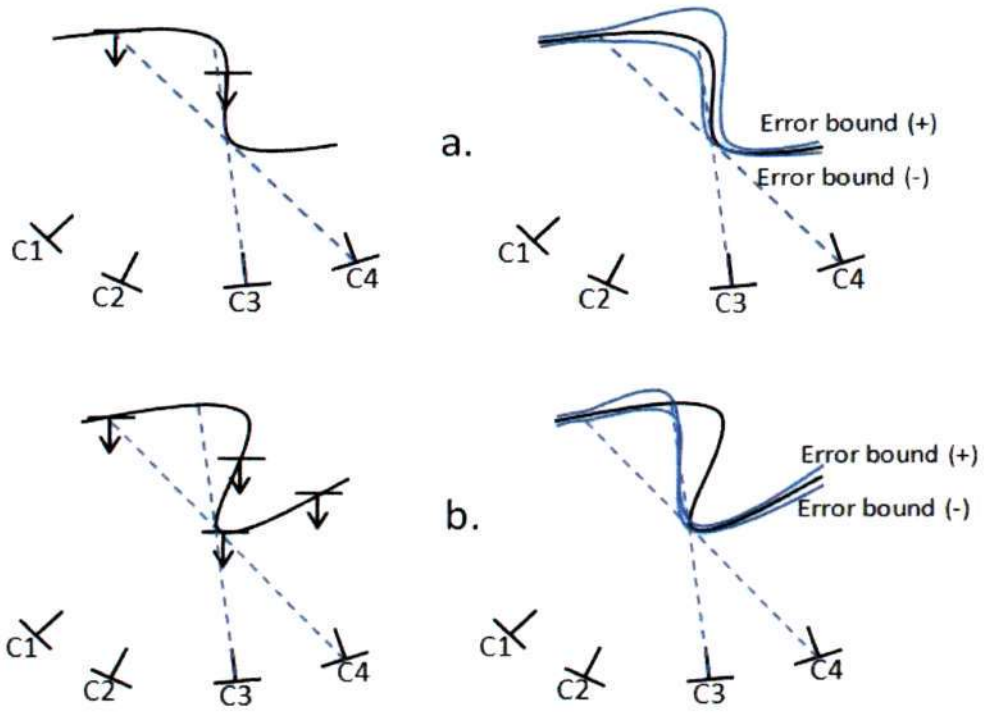



Figure 6.10: The Fixed window application under complicated situation

non-confidence area.

Secondly, the convergence of our method also depends on the visibility tracking. If the visibility tracking wrongly determine a given node is visible from an actually occluded camera viewpoint, it is just the situation in the first step where all nodes are assumed visible to all viewpoints. If the visibility tracking do not assign visibility to a camera viewpoint which actually can see it, the rest of cameras still can correctly recover the 3D position because the visibility tracking is conservative.

Finally, the convergence of our method depends on the full adaptive 3D window reconstruction under the first level of constraint. The first question is how close between current and former normal can be defined as convergent. Our definition is based on the understanding of the wide baseline stereo definition, which considers

view angle differences larger than 30 degrees as wide baseline. So here we define our convergence criteria as half of 30 degrees, which means the surface point can be correctly recovered if the 3D window normal is within 30 degree of local surface normal.

The second question is that whether the process will converge to the correct surface under the above convergent criteria. From our heuristic condition that a 3D window which better approximate the local surface will result in more accurate reconstruction, the iteration algorithm here is actually a greedy algorithm so it will converge fast.

## 6.3 Implementation of Proposed Method

In last section, we proposed our formulation and solution for 3D reconstruction from multiple images. In this section, we concentrate on the implementation and computational cost saving of our proposed algorithm. Considering the complexity of the whole algorithm, the detail implementation for several key points are discussed first, and then the overall scheme is showed in pseudo code.

### 6.3.1 The Key Steps

#### Workspace and Mapping Index

As defined previously, the workspace is normally a volume that contains the target object. It can be specified either by user or by feature-based reconstruction. In Fig.6.7, the workspace is further discretized into nodes. In our practice, the workspace is divided into $Xnum$ by $Ynum$ by $Znum$ nodes along $X$, $Y$ and $Z$ axis. Also, a reference coordinates is selected so that the $Z$ axis is perpendicular to the frontal face.

Our implementation is based on the observation that our proposed method will computationally intensive. We then make trade-offs between time complexity and space complexity as the computer hardware is improving rapidly. We define every node as a structure, and contains not only the node coordinates, also containing its projected image coordinates on each image and the relevant correlation value.

Based on the above definition, our workspace is like a mapping index. After filling in all the information, for every node in the workspace, we can easily know its projection coordinates in each image by indexing the node ID.

**Pre-reconstruct based on Silhouette**

Our focus in this chapter is the reconstruction of 3D object, or specifically human face. Our objective in this section is to find a way to reduce the computational load of the above proposed method. The reconstruction by silhouette methods [129] [130] inspire us to incorporate the silhouette information into account.

A binary image is generated for each input image to separate the foreground and background. Then, with our mapping index to the workspace, every node in workspace is examined on whether it is projected to the foreground or to the background. If in any image a given node is projected to background, then it is eliminated from our workspace. By this way, one fourth to one third memory allocated to workspace could normally be saved.

**Hierarchical strategy**

To further facilitate the computation, we start our algorithm from a sparse reconstruction under our third level assumption. However, the sparsity is only valid on XY-plane, the Z-axis is still transversed at the step of one node. In our implementation, the step along the $X$ and $Y$ axes are $Xnum/8$ and $Ynum/8$ nodes.

After we find the first set of rough node surface, the current node surface is then

interpolated and the new nodes will normally have corresponding searching range which is proportional to the largest searching range of its neighbor. This ratio is set to be 1.5 empirically if it does not go beyond the workspace.

**Layered Depth Image Updating Scheme**

The initialization of LDI is straightforward based on a given node surface. However, the updating of LDI may still involve high computational complexity.

In our context, whenever a node is moved to new location, the former corresponding node will be deleted in the LDI first. The deletion will be found in its neighboring nodes projections on each image by mapping indexing, then searching the corresponding area for this node ID to delete. Secondly, the node at the new location will be added in LDI. The addition process is also facilitated with mapping indexing by limiting the searching area on LDI. Refer to Fig.6.4 for the details.

**Local Curvature Measure and Size Adaptive**

To simplify the size adapting algorithm, we will only vary the 3D window size in several levels instead of continuous changing. In practice, we select 5 different sizes of window for our calculation, which are 7, 11, 21, 31 and 41 node distance for the whole window. The selection is based on the understanding that the 3D window can not be too small to be short of information, nor too large to represent the local object surface. In addition, our reconstruction under the third level constraint is based on the median window size 21.

The local smoothness measure is defined as the standard deviation of derivatives along $X$ and $Y$ axis over the median window size 21. Then, according to this measure, we determine the window size for the given node by thresholding them. An important point to note is that this size adaptation is applied only after the sampling density is high enough in our hierarchy scheme.

**Orientation Adaptive 3D Window**

Our final step is to adjust our 3D window's normal to be aligned with the local current surface normal. Suppose we have the $n$th current node surface and its corresponding node normals, and we align our 3D window to them and update the node surface to the $(n+1)$th, we then compare the two normal at each node and calculate the angle between them. If the angle is less than our threshold, then we will stop updating this node and focus on the ones that the angle is greater than threshold. The frontal nodes on the current node surface are settled first, and this will also pave the way for the rest of nodes. In this way, our area of interest will be smaller and smaller in greedy way and finally be stable.

### 6.3.2 The Overall Implementation Flow

We addressed all the key points in last section and trying to lower or balance the time and space complexities. Our whole algorithm are then implemented in C++ and run on a normal PC with 2.4G CPU and 2G RAM. The overall calculation time will be around half an hour and occupy about 800M RAM. The pseudo code of our algorithm is showed in Fig.6.11.

## 6.4 Experiments, Results and Analysis

In order to verify our proposed algorithm, we need to ensure that the experiment is well controlled, otherwise factors such as noise and inaccurate camera calibrations may confuse the outcome. Thus we choose to test the algorithm with available (ground truth) 3D precision scans of human faces. Our method is applied on faces from OpenGL 3D face database to confirm its performance. We set virtual cameras for the 3D face model and virtually capture images from these set views. Then, the

```
With known WorkSpace and Camera Parameters and Mapping Index
Do
{
        If the node is not projected to background of any image;
        {
                Searching along Z-axis for node with max correlation value;
                Upsample cursurf to next hierarchy level;
                Assign nodes to cursurf;
                Save the searching range;

        }
}While the sampling step is Xnum/2 and Ynum/2;

Update LDI;
Update the cursurf with LDI information in searching range;
Calculate local smoothness measure;
Update LDI;
Update the cursurf with LDI information and adjusted window size;
Update LDI;

Do
{
        Update node normal on cursurf;
        Save cursurf in lastsurf;
        Calculate local smoothness measure;
        Update cursurf with LDI information and full-adaptive window;
        Update node normal on cursurf;
}while the angle between normal on cursurf and normal on lastsurf
        is less than 15 degree;

Output Results;
```

Figure 6.11: The Overall Pseudo-Code

parameters of virtual cameras are used to compute the essential matrices. In this way, we can assume the calibrations are noise free. All the images with subtracted background and parameters, together with a specific workspace established by the user, are input into our algorithm. The primary results are only 3D surface by points surface in 3D space, we then map the texture image onto the surface and show the results by OpenGL.

The input images are showed in Fig. 6.12. We only take four images from different poses as the inputs to our algorithm shown below.

The intermediate results can be seen in Fig. 6.13 without visibility check, and Fig. 6.14 with visibility check. The most obvious difference can be observed at the face around the mouth right corner. The results in Fig.6.13 simply show a big hole over there and in 6.14, it is much more close to the original face model.

The distance from the reconstructed surface to the original model from database is calculated. We define this distance as our reconstruction error. In Table 6.1 the absolute errors are shown for the two reconstruction results. The relative errors for absolute mean error is less than 0.2%, which is showing that our algorithm is really accurate.

Table 6.1: Distance between reconstructed Object and ground truth

| Set | Absolute Mean Error | Standard Deviation | Object Distance From Origin |
|---|---|---|---|
| 1 | 0.0034 | 0.0052 | 5 |
| 2 | 0.0067 | 0.0083 | 5 |

Several Result images in arbitrary pose, together with the corresponding ground truth images, are showed in Fig 6.15 to Fig 6.20. There are several points to note: a) the original texture on low side chin of the object is already blurred because the database models are actually using only one texture image; b), referring to Fig 6.18 and Fig 6.12, the right ear of the model is not correct and was blocked, so it was set to be background; c), the spikes on the boundary of this face model were

due to having less information for correlation for the just-on-boundary nodes, which actually have low effect on the overall reconstructed result.

Another set of experimental results are shown from Fig. 6.21 to Fig 6.27. Fig. 6.21 shows the input images from four different views. Fig.

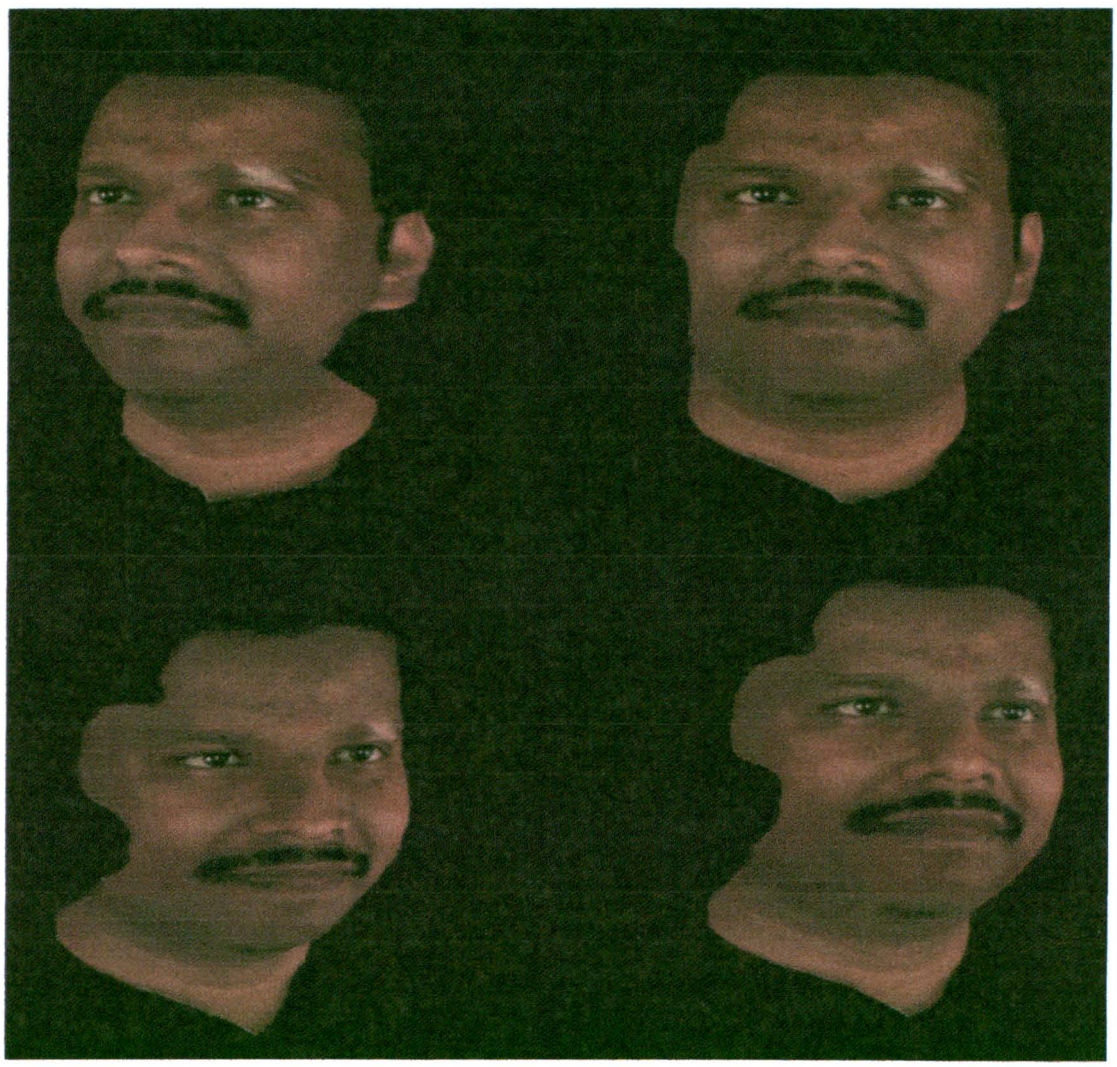


Figure 6.12: The Input Images from Four Different Views

Figure 6.13: Result of 3D correlation window with fixed normal, fixed size and without visibility check

## 6.5 Conclusion

In this chapter, we proposed our 3D face reconstruction algorithm based on multiple images. Our method is under the assumption that the object surface is continuous and pair-wire smooth. Taking this assumption as the starting point, we further give another two levels of constraints. We then start our reconstruction under the strongest constraint in order to get a rough 3D surface of the object for initialization of our iterative, hierarchical algorithm, and step by step the constraint is weakened and applied to the information recovered. From this viewpoint, we have formulated the 3D reconstruction as an information exploration problem.

Figure 6.14: The Result of 3D correlation window with fixed normal, fixed size and visibility check

Our method takes the advantage of 3D correlation window, which will be a better approximation than by 2D window in image, and results in 100% synchronized correlation between image. Our method can also track the occlusion during our process and avoid the ambiguities because of occlusion. In addition, the 3D window takes the projection distortion into account so that the matching areas are aligned across all input images.

Our experiments are based on the 3D face model which is generated from actual human faces. The results confirmed that our method can give reconstructed 3D face model with high accuracy. However, our implementation still has limitation in that a reference view was needed to be set for this 2.5D human face, although theoretically we can select any arbitrary view. We will overcome this limitation in

Figure 6.15: The frontal view comparison. Left: the result 3D face. Right: the original 3D face.

future work for full 3D object reconstruction.

Figure 6.16: The right view comparison. Left: the result 3D face. Right: the original 3D face.

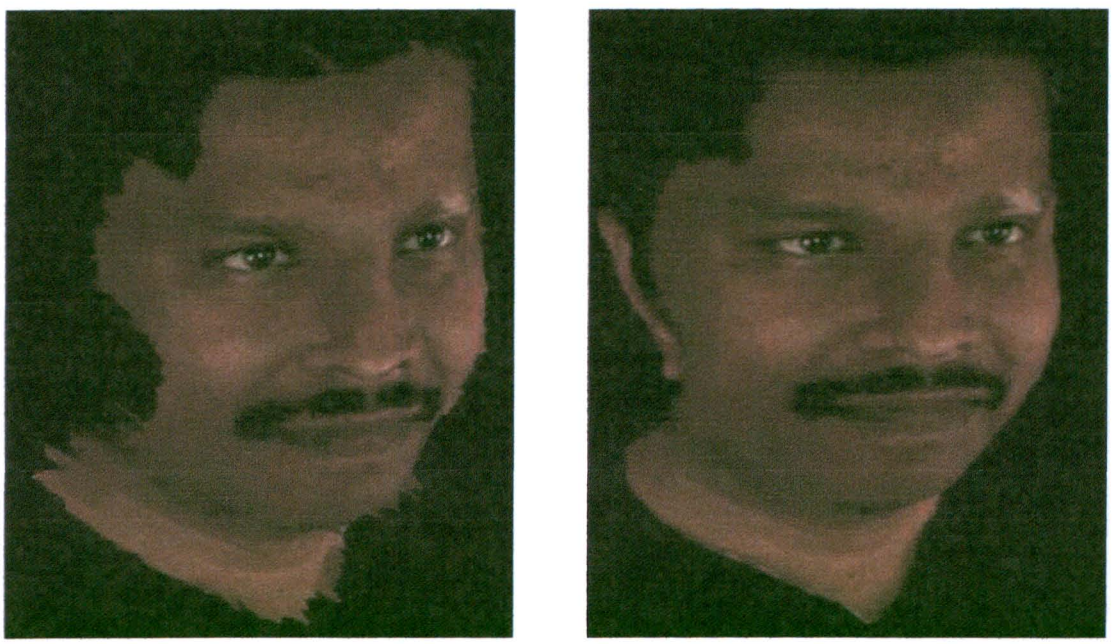


Figure 6.17: The left frontal view comparison. Left: the result 3D face. Right: the original 3D face.

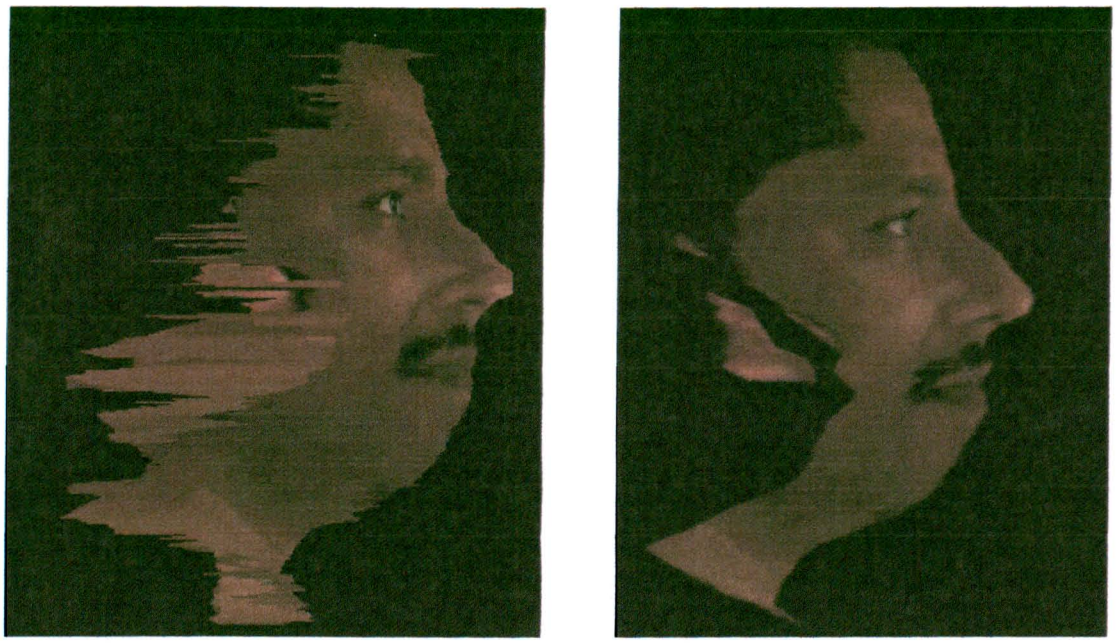


Figure 6.18: The Left view comparison. Left: the result 3D face. Right: the original 3D face.

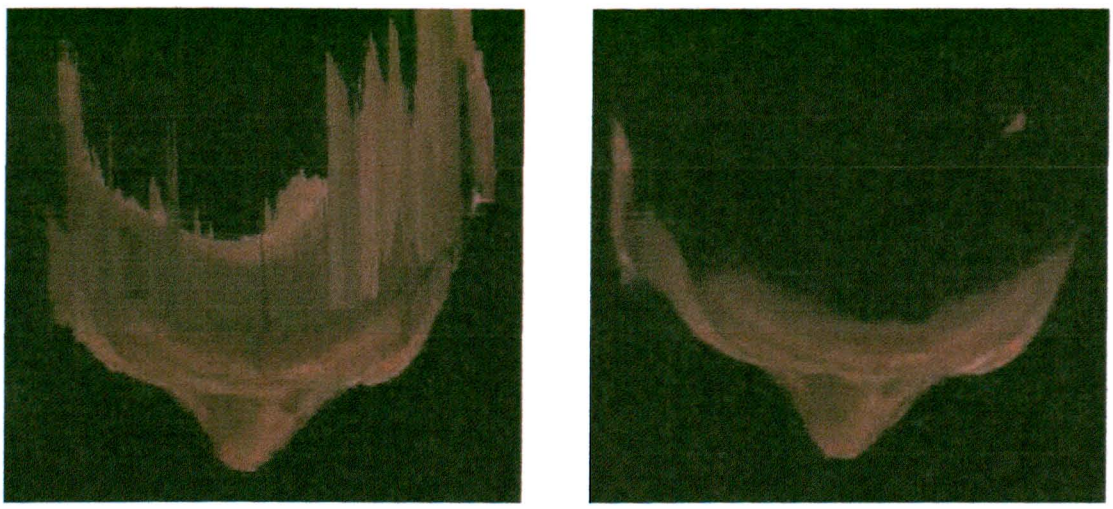


Figure 6.19: The top view comparison. Left: the result 3D face. Right: the original 3D face.

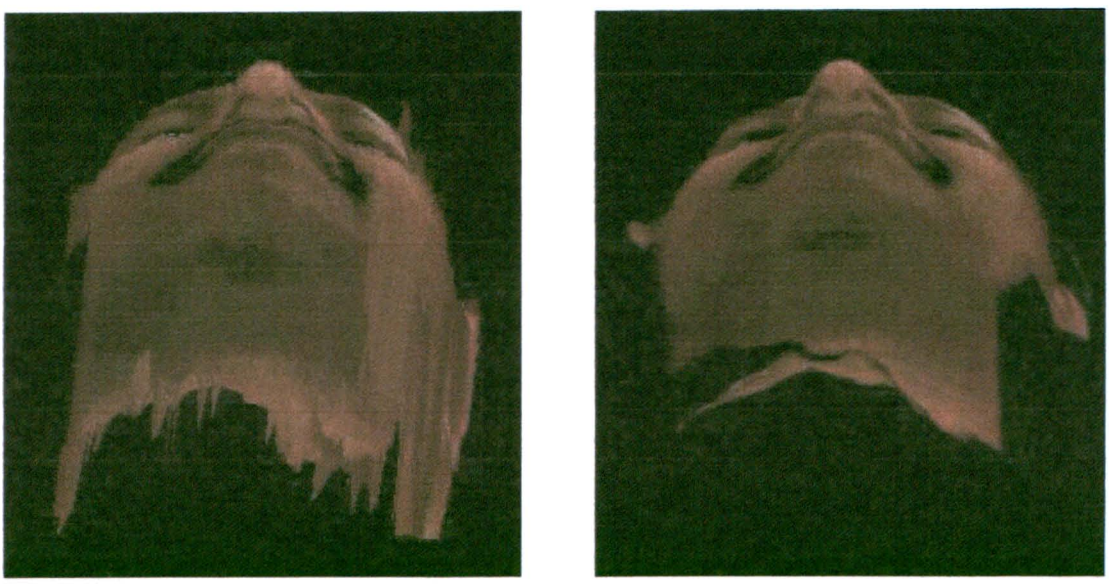


Figure 6.20: The bottom view comparison. Left: the result 3D face. Right: the original 3D face.

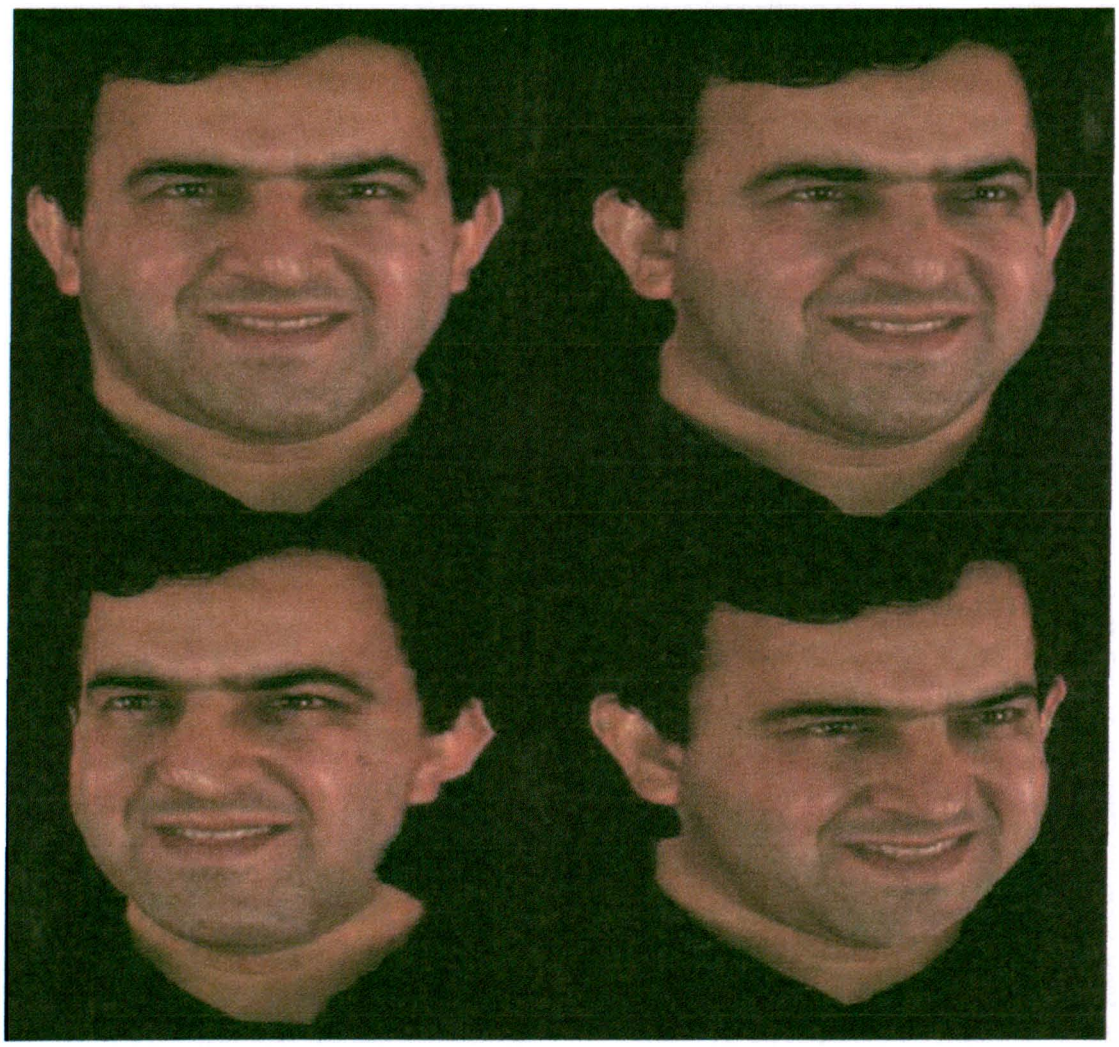


Figure 6.21: The Input Images from Four Views for Result set 2

Figure 6.22: The frontal view comparison for Results set 2. Left: the result 3D face. Right: the original 3D face.

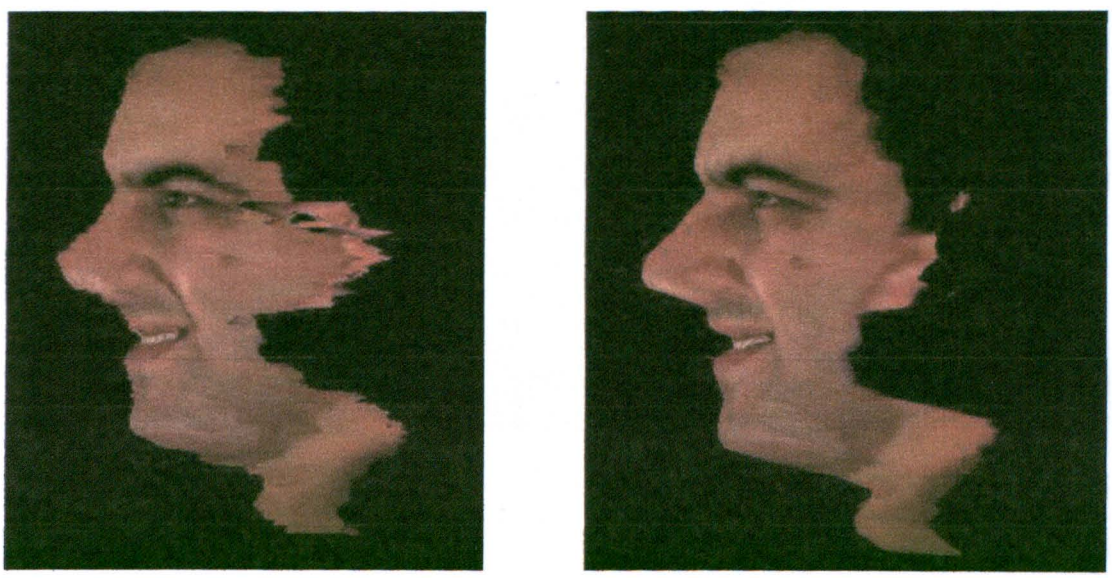


Figure 6.23: The Left view comparison for Results set 2. Left: the result 3D face. Right: the original 3D face.

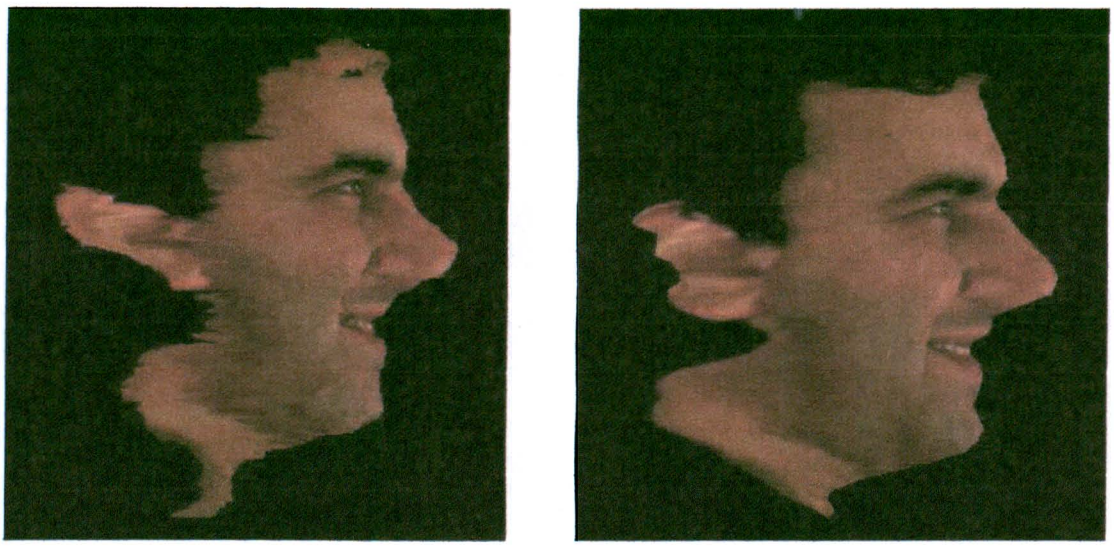


Figure 6.24: The right view comparison for Results set 2. Left: the result 3D face. Right: the original 3D face.

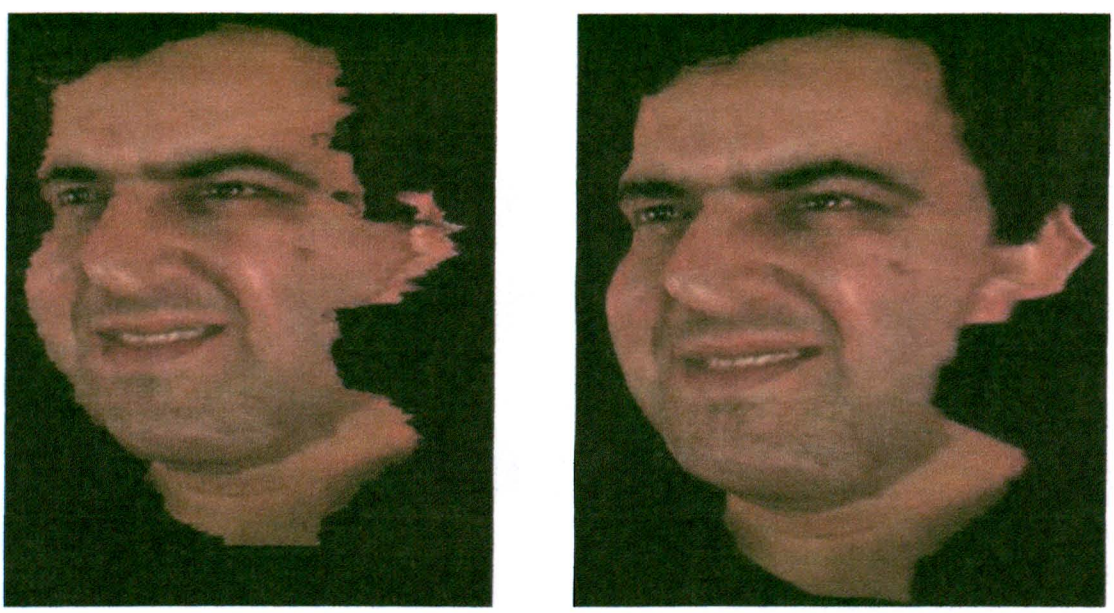


Figure 6.25: The Front right view comparison for Result set 2. Left: the result 3D face. Right: the original 3D face.

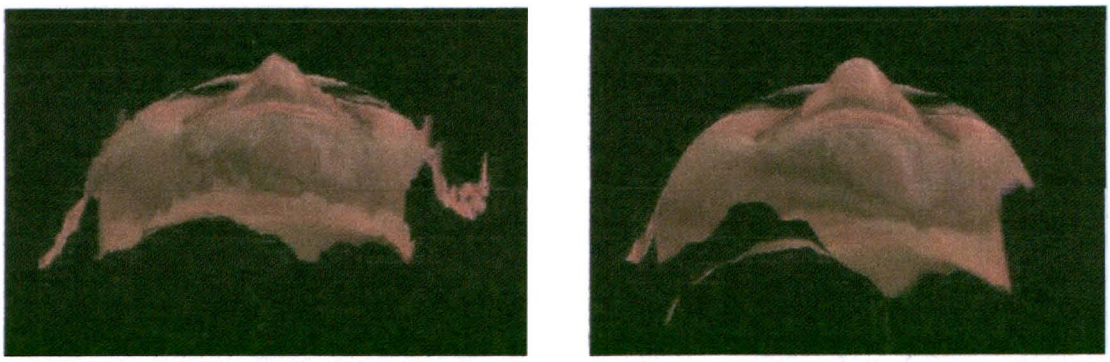


Figure 6.26: The Bottom View comparison for Result set 2. Left: the result 3D face. Right: the original 3D face.

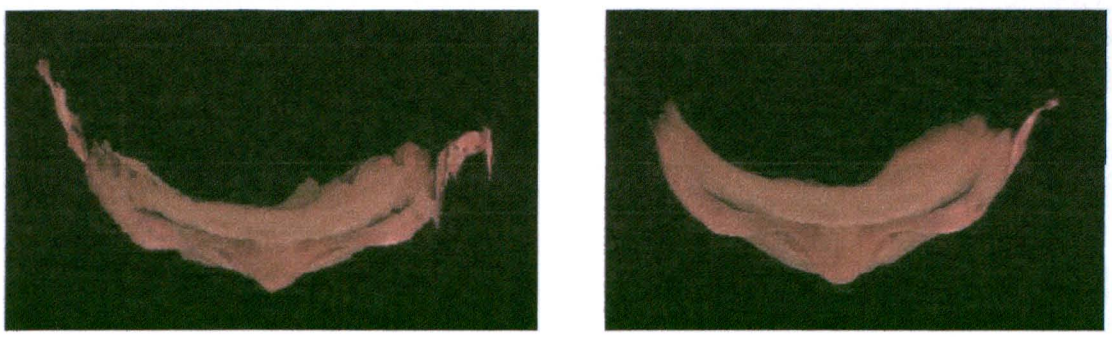


Figure 6.27: The top view comparison for Results set 2. Left: the result 3D face. Right: the original 3D face.

# Chapter 7

# Conclusion and Future Work

In the thesis, several problems about 3D object reconstruction were discussed. Here we give our conclusions about our research and the possible future work.

## 7.1 Discussions and Conclusions

Automatically deriving 3D scene information from (multi-view) images is an extremely difficult research problem that has engage countless researchers over three decades.Many ideas and techniques were proposed and much admirable progress have been made. Still there are still outstanding problems yet to be addressed before a fully automated 3D reconstruction algorithm can be reached.

In this thesis, we set out to look into several problems on 3D multi-view reconstruction. To achieve self-calibration, there must be automatically found reliable points. We try to make more robust existing state-of-the-art corner detectors, basically by re-examining these and find out their weak points. From this, we proposed our pattern based enhancement over SUSAN. Furthermore, for robustness, our corner detector is integrated with self-calibration. Another issue that we address is

stereo matching. We find that two image patches should be allowed to be related by small rotations and added to the translation and scale parameters that was used by Zhang [58]. The inclusion of this added degree of freedom, rotation, has been verified to show some significant improvements. Finally, the usual correlation approach to stereo matching has inherent problems. The key factors are size of the image patches, perspective distortions orientation of surface patches and occlusions... We address this by starting instead from hypothesized 3D object patches as have been done by Voxel coloring [111] and space carving [114]. We proposed a hierarchical, relaxation and information-based technique to address these outstanding issues.

One issue that we did not include is to optimize the search for the correct spatial surface. This has been recently and successfully be tackled by graph cut [131] [132].

### 7.1.1 Corner Detection and Corner Matching

Our idea here is to consider corner detection and corner matching in an integrated manner. Following it, we redesign both corner detector and corner matching algorithm, and archived reduction on the overall computational complexity and improvement on the robustness.

#### The Improvement on SUSAN Corner Detector

To accurately obtain the camera relationship, corners that are matched have to be detected both accurately and robustly. We then tried to improve the robustness of SUSAN corner detector [46] which has accurate detection. The local topology of image is then analyzed and is taken as a detector of false corners by SUSAN. In this way, we archiveed our objective for more accurate and robust corners. And of course, the amount of corners found by our method will be less but this does not matter, as they are more than sufficient to be used to compute the epipolar geometry accurately.

**The Corner Matching**

Even with special design in corner detection, a straightforward application of similarity check, such as correlation, will result in noisy corner matching. Our second task is to identify the true matches in some way. We then tried to improve on the landmark work in [58] by integrating more necessary information and proposing a new core function into the scheme. Together with our corner detector, we notably reduced the computation time on non-linear processing step of our corner matching method, which is the most computationally intensive step, and achieved a more robust and accurate matching results.

### 7.1.2 Reconstruction with Adaptive 3D Correlation Window

There are several variations between images from different viewpoints needed to be addressed during the establishment of dense correspondence and reconstruction. The most difficult ones are perspective distortion, changes by pose variation and occlusion. The 2D method, which search in image space to find correspondence, have the inherent weakness on dealing with the variations. We proposed to use 3D adaptive correlation window to work in 3D space and formulate a scheme to address the above variations. At the same time, layered depth image is used to track the occlusion in our proposed scheme. Our experiment demonstrated the strength of our method.

## 7.2 Future Works

In last section, we gave our accomplishments on the three key problems in 3D reconstruction. With these solutions, we are closer to an automatic reconstruction

system. However, the research we have done is just open another door for the 3D reconstruction problem.

The first possible way to go is automatically set the parameters during corner detection and corner matching. Our parameters are currently decided by user according to the image noise, resolution and object characteristics. So one set of parameters may be suitable for one group of tasks but not for others. If we can analyze the images before we start the whole process, we may be able to design a method to automatically select parameters.

The second improvement can be made for the pre-setting of work space for reconstruction. Currently, the work space is set up by user. However, a straightforward idea is whether it can be determined by our matched corners. The matched corners can be triangulated to 3D space so that we can roughly know the position of our target object, and we can then decide the workspace based on the sparse 3D points.

Lastly, although our 3D reconstruction by 3D adaptive window is feasible for 360 degree reconstruction theoretically, our implementation, as a proof of principle is only a part of the full 360 degree reconstruction. A straightforward idea is to apply our method from 6 directions of the object. However, the computational complexity is apparently very high. On the other side, our method need maintaining a frontal-to-backward order for occlusion tracking. So, in a full 3D reconstruction method, how to take all the images into account while keep tracking the visibility from each view point will be valuable problem to study.

# Appendix: Computing Pseudo-Fundamental Matrix by LMedS

To avoid the sensitivity inherent in linear methods, a robust nonlinear method should be used to compute the fundamental matrix. The method used in this section is the least-median-of-squares(LMeds) [?].

The principle of LMedS method is to minimize the median of the squared residuals:

$$\min med_i r_i^2 \tag{7.1}$$

That is, the LMedS must yield the smallest value for the median of squared residuals computed for the entire data set. In the context of computing fundamental matrix, $r_i^2$ is defined as:

$$r_i^2 = d^2(\tilde{\mathbf{m}}_i', \mathbf{F}_{sJ}\tilde{\mathbf{m}}_i) + d^2(\tilde{\mathbf{m}}_i, \mathbf{F}_{sJ}^T\tilde{\mathbf{m}}_i') \tag{7.2}$$

where $(\tilde{\mathbf{m}}_i, \tilde{\mathbf{m}}_i')$ is the $i$th matching pair. The $\mathbf{F}_{sJ}$ is defined as followed. Given $n$ matching pairs $(\tilde{\mathbf{m}}_i, \tilde{\mathbf{m}}_i')$, $i = 1 \sim n$, a Monte Carlo type technique is used to draw $m$ random subsamples of $p = 8$ different matching pairs. For the $J$th subsample, the fundamental matrix $\mathbf{F}_{sJ}$ is determined. Then, the median of the squared residuals

with repect to the whole set of matching pairs is defined as:

$$\mathbf{M}_J = med_{i=1,\cdots,n}(r_i^2) \tag{7.3}$$

Obviously, the searching space of possible estimates is very large. To reduce the computation complexity, only a randomly chosen subset of the whole data set can be analyzed. Suppose the outliers cover $\epsilon$ percent of the whole data set. In addition, a subsample is a "good" one if it contains $p$ good matching pairs. Then at least there should be one of the $m$ subsamples is good, the probability should be [?]:

$$P = 1 - [1 - (1 - \epsilon)^p]^m \tag{7.4}$$

The $P$ should be close to 1 and p=8 so that $m$ can be determined given $\epsilon$. In practice, the $\epsilon$ is assumed equal to 40% and p=0.99, then m=272.

Another problem that have to be noted is the randomly chosen subsample may contain the matching pairs that make the computation of $\mathbf{F}_s$ be degenerated. For example, the eight matching pairs are very near to each other. The degenerated subsample should be avoid because the fundamental matrix computed from them will be highly unstable and then be useless. In [58], a method that can avoid degeneracy of subsample is introduced. The basic principle is as follow: given a set of matching pairs, the max and min of the coordinates of the corners in the first image are found. Then the region between them is divided into $l \times l$ patches. The patches that no corner falls into are discarded, furthermore, eight patches are selected from the remained patches, and then one match in each patch is chosen randomly to generate the subsample for LMedS.

# Publication List

1. Wenbo Zhang, Xinting Gao, Sung, E. Sattar, F. Venkateswarlu, R. "A Robust Feature-based Matching of Two Uncalibrated Images," *Control, Automation, Robotics and Vision Conference*, Kunming, China, Dec 6-9, 2004.

2. Wenbo Zhang, Xinting Gao, Sung, E. Sattar, F. Venkateswarlu, R. "A Robust Feature-based Matching of Two Uncalibrated Images," *Pattern Recognition Letters*, Pages: 1222-1231. Volume 28, Issue 10.

# Bibliography

[1] B. Sabata and J. Aggarwal, "Estimation of motion from a pair of range images: A review," *CVGIP*, vol. 54(3), pp. 309–324, 1991.

[2] B. Sabata and J. K. Aggarwal, "Surface correspondence and motion computation from a pair of range images," *Computer Vision and Image Understanding*, pp. 232–250, 1996.

[3] G. Pajares, J. de la Cruz, and J. Aranda, "Stereo matching based on the self-organizing feature-mapping algorithm," *Pattern Recognition Letters*, pp. 319–330, 1998.

[4] C. Jawahar and P. J. Narayanan, "An adaptive multifeature correspondence algorithm for stereo using dynamic programming," *Pattern Recognition Letters*, pp. 549–556, 2002.

[5] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature, 293(10):133-135, September 1981*, vol. 293, pp. 133–135, 1981.

[6] E. Kaliva and A. Delopoulos, "An iterative method for 3d reconstruction under orthography," in *Visualization, Imaging and Image Processing 2005*, Benidorm, Spain, 2005.

[7] M. A. Fahiem, S. A. Haq, and F. Saleemi, "A review of 3d reconstruction techniques from 2d orthographic line drawings," in *The Geometric Modelling and Imaging*, Washington, DC, USA, 2007, pp. 60–66.

[8] T. Moons, V. Gool, M. V. Diest, and E. Pauels, "Affine reconstruction fro perspective image pairs," in *The DARPA-ESPRIT workshop on Applications of Invariants in Computer Vision*, Azores, Portugal, 1993, pp. 249–266.

[9] T. Moons, L. V. 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*, pp. 77–83, 1996.

[10] M. Obeysekera, *Affine Reconstruction from multiple views using Singular Value Decomposition.* The University of western Australia, 2003.

[11] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, "Visual modeling with a hand-held camera," *International Journal of Computer Vision*, pp. 207–232, 2004.

[12] A. Z. Richard Hartley, *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2002.

[13] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach.* Upper Saddle River, New Jersey 07458: Prentice-Hall, 2002.

[14] S. Soatto and P. Perona, "Reducing structure from motion:a general framework for dynamic vision part 1: Modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 9, pp. 933–942, 1998.

[15] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15(4), pp. 353–363, 1993.

[16] Z. zhang, "A new and efficient iterative approach to image matching," in *Computer Vision and Image Processing., Proceedings of the 12th IAPR International Conference on*, 1994, pp. 563 –565.

[17] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Approach.* Cambridge, MA, USA: MIT press, 1993.

[18] K. Kanatani, *Geometric Computation for Machine Vision.* New York, NY, USA: Oxford University Press, USA, 1993.

[19] O. Faugeras, "Stratification of 3-d vision: Projective, affine, and metric representations," *Journal of the Optical Society of America A*, pp. 465–484, 1995.

[20] J. L. Mundy and A. Zisserman, *Geometric Invariance in Computer Vision.* MIT Press, Cambridge, Massachusetts, USA, 1992.

[21] S. C. Radu Horaud and F. Dornaika, "Object pose: The link between weak perspective, para perspective, and full perspective," INRIA, Techinical Report 2356, September 1994. [Online]. Available: http://www.inria.fr/rrrt/rr-2356.html

[22] Y. Ohta, K. Maenobu, and T. Sakai, "Obtaining surface orientation from texels under perspective projection," in *Proc. Seventh International Joint Conf. Artificial Intelligence*, 1981, pp. 746–751.

[23] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, pp. 7–42, 2004.

[24] C. Zitnick and T. Kanade, "A cooperative algorithm for stereo matching and occlusion detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 2000.

[25] J. Louchet, "Stereo analysis using individual evolution strategy," in *International Conference on Pattern Recognition*, pp. 908–911.

[26] H. Ishikawa, "Multi-scale reature selection in stereo," in *Computer Vision and Pattern Recognition*, pp. 132–137.

[27] M. Ouali, H. Lange, and C. laurgeau, "An energy minimization approach to dense stereovision," in *International Conference on Image Processing*, pp. 841–845.

[28] P. Eisert, E. Steinbach, and B. Girod, "Automatic reconstruction of stationary 3-d objects from multiple uncalibrated camera view," *IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on 3D Video Technology*, pp. 261–277, 2000.

[29] S. Roy and I. J. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem," in *International Conference on Computer Vision*, Bombay, Indin, January 1998, pp. 492–499.

[30] J. Kannala and S. Brandt, "Quasi-dense wide baseline matching using match propagation," in *International Conference on Pattern Recognition*, Minnesota, USA, June 2007, pp. 1–8.

[31] R. Szeliski and S. Kang, "Recovering 3d shape and motion from image streams using nonlinearleast squares," in *Computer Vision and Pattern Recognition*, Los Alamitos, CA, 1993, pp. 752–753.

[32] U. Dhond and J. Aggarwal, "Structure from stereo-a review," *Systems, Man and Cybernetics, IEEE Transactions on*, pp. 1489–1510, 1989.

[33] T. Huang and A. Netravali, "Motion and structure from feature correspondences: a review," in *IEEE*, Feb. 1994, pp. 252 – 268.

[34] C. Park and H. Park, "A robust stereo disparity estimation using adaptive window search and dynamic programming search," *Pattern Recognition*, pp. 2573–2576, 2001.

[35] S. Gutierrez and J. Marroquin, "Robust approach for disparity estimation in stereo vision," *Image adn Vision Computing*, pp. 183–195, 2004.

[36] J. Oh, S. Ma, and C. Kuo, "Stereo matching via disparity estimation and surface modeling," in *Computer Vision and Pattern Recognition*, Minneapolix, MN, June 2007, pp. 1–8.

[37] H. Lim and H. W. Park, "A dense disparity estimation method using color segmentation and energy minimization," in *The International Conference on Image Processing*, Atlanta, GA, USA, October 2006, pp. 1033–1036.

[38] H. P. Moravec, "Towards automatic visual obstacle avoidance," in *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, August 1977, p. 584.

[39] H.Moravec, "Visual mapping by a robot rover," in *Proceedings of the 6th International Joint Conference on Artificial Intelligence*, August 1979, pp. 599–601.

[40] C. G. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of the 4th Alvey Vision Conference*, 1988, pp. 147–151.

[41] Z. Zheng, H. Wang, and E. K. Teoh, "Analysis of gray level corner detection," *Pattern Recognition Letters*, pp. 149–162, 1999.

[42] L. Kitchen and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognition Letters*, pp. 95–102, 1982.

[43] W. H. and B. M. A., "A practical solution for corner detection," in *International Conference on Image Processing*, pp. 919–923.

[44] P. R. Beaudet, "Rotational invariant image operators," in *Fourth International Conference on Pattern Recognition*, 1978, pp. 579–583.

[45] R. Deriche and G. Giraudon, "A computational approach for corner and vertex detection," *International Journal of Computer Vision*, pp. 101–124, 1993.

[46] S. Smith and J. Brady, "Susan - a new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.

[47] M. Trajkovic and M. Hedley, "Fast corner detection," *Image and Vision Computing*, pp. 75–87, 1998.

[48] C. D. Y. Sun Cheol bae, In So Kweon, "Cop: a new corner detector," *Pattern Recognition Letters*, pp. 1349–1360, 2002.

[49] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, pp. 79–116, 1998.

[50] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 91–110, 2004.

[51] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, pp. 63–86, 2004.

[52] T. Lindeberg and J. Garding, "Shape-adapted smoothing in estimation of 3-d shape cues from affine deformations of local 2-d brightness structure." *Image and Vision Computing*, vol. 15, no. 6, pp. 415–434, 1997.

[53] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proceedings of the 7th European Conference on Computer Vision*. Springer, 2002, pp. 128–142, copenhagen.

[54] F. Schaffalitzky and A. Zisserman, "Viewpoint invariant texture matching and wide baseline stereo," in *Proceedings of the Eighth IEEE International Conference on Computer Vision*, Vancouver, BC, Canada, pp. 636–643.

[55] F. Mokhtarian and F. Mohanna, "Performance evaluation of corner detectors using consistency and accuracy measures," *Computer Vision and Image Understanding*, pp. 81–94, 2006.

[56] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3d objects," *International Journal of Computer Vision*, pp. 263–284, 2006.

[57] C. Schmid, R. Mohr, and C. Bauckhage, "Comparing and evaluating interest points," in *Proceedings of the 6th International Conference on Computer Vision*. Bombay, India: IEEE Computer Society Press, January 1998.

[58] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. LUONG, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intellegence Journal*, pp. 87–119, 1995.

[59] P. Viola and W. M. III, "Alignment by maximization of mutual information," *International Journal of Computer Vision*, p. 137C154, 1997.

[60] P. Fua, "A parallel stereo algorithm that produces dense depth maps and preserves image features," vol. 6, no. 1, 1993, pp. 35–49.

[61] C. Sun, "A fast stereo matching method," in *In Digital Image Computing: Techniques and Applications*, 1997, pp. 95–100.

[62] O. Faugeras, B. Hotz, H. Mathieu, T. Vieville, Z. Zhang, P. Fua, E. Theron, L. Moll, G. Berry, V. J., and C. Bertin, P.and Proy, "Real time correlation-based stereo: Algorithm, implementations and applications," in *Technical Report RR-2013, INRIA*, 1993.

[63] K. R. J. Ramesh and S. Brian, *Machine Vision*. McGraw Hill, 1995.

[64] S. Ullman, *High Level Vision*. MIT Press, 1996.

[65] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15(4), pp. 353–363, 1993.

[66] M. Levine, D. O'Handley, and G. Yagi, "Computer determination of depth maps," *Computer Graphics and Image Process*, pp. 131–150, 1973.

[67] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 920–932, 1994.

[68] H. Saito and M. Mori, "Application of genetic algorithms to stereo matching of images," *Pattern Recognition Letters*, pp. 815–821, 1995.

[69] S. Lloyd, "Stereo matching using intra- and inter-row dynamic programming," *Pattern Recognition Letters*, vol. 4, pp. 273–277, 1986.

[70] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 2, pp. 139–154, March 1985.

[71] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs, "A maximum likelihood stereo algorithm," *Computer Vision Image Understanding*, vol. 63, no. 3, pp. 542–567, 1996.

[72] P. N. Belhumeur, "A bayesian approach to binocular stereopsis," *International Journal of Computer Vision*, vol. 19, no. 3, pp. 237–260, 1996.

[73] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 269–293, 1999.

[74] D. Marr and T. Poggio, "Cooperative computation of stereo disparity," vol. 194, no. 4262, pp. 283–287, October 1976.

[75] N. Yokoya, "Dense matching of two views with large displacement," in *The First IEEE International Conference on Image Processing*, 1994, pp. 213–217.

[76] K. Do, Y. Kim, T. Uam, and Y. Ha, "Iterative relaxational stereo matching based on adaptive support between disparities," *Pattern Recognition*, vol. 31, no. 8, pp. 1049–1059, August 1998.

[77] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[78] P. H. S. Torr and D. W. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *International Journal of Computer Vision*, pp. 271–300, 1997.

[79] Z. Wu, F.C.and Hu and F. Duan, "8-point algorithm revisited: factorized 8-point algorithm," in *Tenth IEEE International Conference on Computer Vision*, Beijing, China, 3 2005, pp. 488–494.

[80] R. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, June 1997.

[81] H. Bian and J. Su, "Feature matching based on geometric constraints in stereo views of curved scenes," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug 2005, pp. 313–318.

[82] Y. Sheikh, A. Hakeem, and M. Shah, "On the direct estimation of the fundamental matrix," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–7.

[83] J. Barreto and K. Daniilidis, "Fundamental matrix for cameras with radial distortion," in *Tenth IEEE International Conference on Computer Vision*, 2005, pp. I: 625–632.

[84] X. Armangue and J. Salvi, "Overall view regarding fundamental matrix estimation," *Image and Visioin Computing*, vol. 21, no. 2, pp. 205–220, February 2003.

[85] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *International Journal of Computer Vision*, vol. 27, no. 2, pp. 161–195, March 1998.

[86] I. Jung and S. Lacroix, "A robust interest points matching algorithm," in *Eighth IEEE International Conference on Computer Vision*, Vancouver, Canada, July 2001, pp. II: 538–543.

[87] R. C. Paul Smith, D. Sinclair and K. Wood, "Effective corner matching," in *Proceedings of the British Machine Vision Conference*, pp. 545–556.

[88] V. Gouet, P. Montesinos, and D. Pele, "A fast matching method for color uncalibrated images using differential invariants," in *British Machine Vision Conference*, 1998, pp. 367–376.

[89] G. Finlayson, M. Drew, and B. Funt, "Color constancy: generalized diagonal transforms suffic," *Journal of the Optical Society of America A*, pp. 3011–3019, 1994.

[90] M. Lourakis, A. Argyros, and K. Marias, "A graph-based approach to corner matching using mutual information as a local similarity measure," in *Proceedings of the 17th International Conference on Pattern Recognition*, 2004, pp. II: 827–830.

[91] M. Lhuillier and L. Quan, "Match propagation for image-based modeling and rendering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1140–1146, August 2002.

[92] M. Awrangjeb and G. Lu, "A robust corner matching technique," in *IEEE International Conference on Multimedia and Expo*, Beijing, China, July 2007, pp. 1483–1486.

[93] K.-J. Lee, Y.-T. Kim, H.-C. Myung, J.-M. Kim, and Z. Bien, "A corner matching algorithm with uncertainty handling capability," in *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, July 1997, pp. 1469 – 1474.

[94] S. Meshoul and M. Batouche, "A fully automatic method for feature-based image registration," in *2002 IEEE International Conference on Systems, Man and Cybernetics*, Hammamet, Tunisia., Oct 2002.

[95] M. Lhuillier, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE Transaction Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 418–433, 2005, senior Member-Long Quan.

[96] A. Kara, D. Wilkes, and K. Kawamura, "3d structure reconstruction from point correspondence between two perspetive projections," *Computer Vision, Graphics and Image Processing*, pp. 392–397, 1994.

[97] G. Pajares, J. M. de la Cruz, and J. A. Lopez Orozco, "Relaxation labeling in stereo image matching," *Pattern Recognition*, pp. 53–68, 2000.

[98] E. Vincent and R. Laganiere, "Detecting and matching feature points," *Journal of Visual Communication and Image Representation*, pp. 38–54, 2005.

[99] P. H. S. Torr and A. Zisserman, "Feature based methods for structure and motion estimation," in *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*. London, UK: Springer-Verlag, 2000, pp. 278–294.

[100] P. Eisert, E. Steinbach, and B. Girod, "Automatic reconstruction of stationary 3-d objects from multiple uncalibrated camera views," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 2, pp. 261–277, March 2000.

[101] P. Eisert, E. Steinbach, and Girod, "Multi-hypothesis, volumetric reconstruction of 3-d objects from multiple calibrated camera views," in *The International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, AZ, March 1999, pp. 3509–3512.

[102] M. Goesele, B. Curless, and S. M. Seitz, "Multi-view stereo revisited," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2402–2409.

[103] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques.* New York, NY, USA: ACM, 1996, pp. 303–312.

[104] P. Narayanan, P. Rander, and T. Kanade, "Constructing virtual worlds using dense stereo," in *the Sixth IEEE International Conference on Computer Vision*, January 1998.

[105] S. Vedula, P. Rander, H. Saito, and T. Kanade, "Modeling, combining, and rendering dynamic real-world events from image sequences," in *the 4th Conference on Virtual Systems and Multimedia*, November 1998, pp. 326 – 332.

[106] P. Fua and Y. G. Leclerc, "Object-centered surface reconstruction: combining multi-image stereo and shading," *International Journal of Computer Vision*, vol. 16, no. 1, pp. 35–55, 1995.

[107] P. Fua and P. Sander, "Reconstructing surfaces from unstructured 3d points," in *DARPA Image Understanding Workshop.*

[108] M. Pollefeys, R. Koch, and L. Van Gool, "Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters," *International Journal of Computer Vision*, vol. 32, no. 1, pp. 7–25, August 1999.

[109] A. Fitzgibbon and A. Zisserman, "Automatic 3d model acquisition and generation of new images from video sequences," in *Proceedings of European Signal Processing Conference*, Rhodes, Greece, 1998, pp. 1261–1269.

[110] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer, "A survey of volumetric scene reconstruction methods from photographs," in *The International Workshop on Vo.ume Graphics*, New York, USA, June 2001, pp. 81–100.

[111] S. Seitz and C. Dyer, "Photorealistic scene reconstruction by voxel coloring," *International Journal of Computer Vision*, vol. 35, no. 2, pp. 151–173, November 1999.

[112] A. Prock and C. Dyer, "Towards real-time voxel coloring," in *DARPA Image Understanding Workshop*, 1998, pp. 315–321.

[113] K. Kutulakos and Seitz, "What do n photographs tell us about 3d shape?" in *Technical Report*, 1998.

[114] K. Kutulakos and S. Seitz, "A theory of shape by space carving," *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199–218, July 2000.

[115] K. Kutulakos, "Approximate n-view stereo," in *The Sixth European Conference on Computer Vision*, Dublin, Ireland, June 2000, pp. I: 67–83.

[116] A. Broadhurst and R. Cipolla, "A statistical consistency check for the space carving algorithm," in *The 11th British Machine Vision Conference*, Bristol, UK, Sept 2000.

[117] W. B. Culbertson, T. Malzbender, and G. G. Slabaugh, "Generalized voxel coloring," in *Proceedings of the International Workshop on Vision Algorithms, ICCV*. London, UK: Springer-Verlag, 2000, pp. 100–115.

[118] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Transaction on Pattern Recognition and Machine Intelligence*, pp. 147–159, 2004.

[119] ——, "Multi-camera scene reconstruction via graph cuts," in *European Conference on Computer Vision*, May 2002.

[120] Z. R. G. S. Kolmogorov, V., "Generalized multi-camera scene reconstruction using graph cuts," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, 2003, pp. 501–516.

[121] P. H. S. T. George Vogiatzis, Carlos Hern an dez Esteban and R. Cipolla, "Multi-view stereo via volumetric graph-cuts and occlusion robust photo-consistency," *IEEE Transaction on Pattern Recognition and Machine Intelligence*, pp. 2241–2246, 2007.

[122] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of Computation Physics*, pp. 12–49, 1988.

[123] O. D. Faugeras and R. Keriven, "Complete dense stereovision using level set methods," in *Proceedings of the 5th European Conference on Computer Vision*, vol. I, London, UK, 1998, pp. 379–393.

[124] G. Slabaugh, R. Schafer, and M. Hans, "Multi-resolution space carving using level set methods," in *The International Conference on Image Processing*, 2002, pp. II: 545–548.

[125] J. Hailin, "Variational methods for shape reconstruction in computer vision," *PH.D Thesis*, August 2003.

[126] P. Tissainayagam and D. Suter, "Assessing the performance of corner detectors for point feature tracking applications," *Image and Vision Computing*, pp. 663–679, 2004.

[127] R. I. Hartley, "In defence of the 8-point algorithm," in *Proceedings of the Fifth International Conference on Computer Vision*, Washington, DC, USA, 1995, pp. 1064–1070.

[128] J. Geusebroek, G. Burghouts, and A. Smeulders, "The amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, January 2005.

[129] J. Y. Zheng, "Acquiring 3-d models from a sequence of contours," *IEEE Transaction on Pattern Recognition and Machine Intelligence*, pp. 163–178, 1994.

[130] A. Y. Mulayim, U. Yilmaz, and V. Atalay, "Silhouete-based 3d model reconstruction from multiple images," *IEEE Transaction on Systems, Man, and Cybernetics, B*, pp. 582–591, 2003.

[131] S. Paris, F. Sillion, and Q. L, "A surface reconstruction method using global graph cut optimization," *International Journal of Computer Vision*, pp. 141–161, 2006.

[132] S. Tran and L. davis, "3d surface reconstruction using graph cuts with surface constraints," in *European Conference on Computer Vision*, pp. 219–231.