3D RECONSTRUCTION OF UNDERWATER SCENES FROM UNCALIBTARED VIDEO SEQUENCES

MUSTAFA YAVUZ KIRLI

AUGUST 2008

3D RECONSTRUCTION OF UNDERWATER SCENES FROM UNCALIBRATED VIDEO SEQUENCES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF
NATURAL AND APPLIED SCIENCES OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MUSTAFA YAVUZ KIRLI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2008

Approval of the thesis:

3D RECONSTRUCTION OF UNDERWATER SCENES

FROM UNCALIBRATED VIDEO SEQUENCES

submitted by **MUSTAFA YAVUZ KIRLI** in partial fulfillment of the degree of **Master of Science in Electrical and Electronics Engineering, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, **Graduate School of Natural and Applied Sciences**

Proj. Dr. İsmet ERKMEN _____
Head of Department, **Electrical and Electronics Engineering**

Assist. Prof. İlkay ULUSOY _____
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Kemal LEBLEBİCİOĞLU _____
Electrical and Electronics Engineering Dept., METU

Assist. Prof. İlkay ULUSOY _____
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Aydın ALATAN _____
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Çağatay CANDAN _____
Electrical and Electronics Engineering Dept., METU

Metin AKTAŞ _____
Electrical and Electronics Engineer, M.Sc., ASELSAN

Date: 28.08.2008

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Mustafa Yavuz KIRLI

Signature :

ABSTRACT

3D RECONSTRUCTION OF UNDERWATER SCENES

FROM UNCALIBRATED VIDEO SCENES

KIRLI, Mustafa Yavuz

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Assist. Prof. İlkay ULUSOY

August 2008, 144 pages

The aim of this thesis is to reconstruct 3D representation of underwater scenes from uncalibrated video sequences. Underwater visualization is important for underwater Remotely Operated Vehicles and underwater is a complex structured environment because of inhomogeneous light absorption and light scattering by the environment. These factors make 3D reconstruction in underwater more challenging.

The reconstruction consists of the following stages: Image enhancement, feature detection and matching, fundamental matrix estimation, auto-calibration, recovery of extrinsic parameters, rectification, stereo matching and triangulation.

For image enhancement, a pre-processing filter is used to remove the effects of water and to enhance the images. Two feature extraction methods are examined: 1. Difference of Gaussian with SIFT feature descriptor, 2. Harris Corner Detector with grey level around the feature point. Matching is performed by finding similarities of SIFT features and by finding correlated grey levels respectively for each feature extraction method. The results show that SIFT performs better than

Harris with grey level information. RANSAC method with normalized 8-point algorithm is used to estimate fundamental matrix and to reject outliers. Because of the difficulties of calibrating the cameras in underwater, auto-calibration process is examined. Rectification is also performed since it provides epipolar lines coincide with image scan lines which is helpful to stereo matching algorithms. The Graph-Cut stereo matching algorithm is used to compute corresponding pixel of each pixel in the stereo image pair. For the last stage triangulation is used to compute 3D points from the corresponding pixel pairs.

Keywords: Underwater, 3D Reconstruction, SIFT, Rectification, Stereo Mapping, Underwater Image Enhancement, Triangulation, Kruppa Equations

ÖZ

KALİBRE EDİLMEMİŞ SUALTI VİDEO SERİSİNDEN

3 BOYUTLU SAHNE GERİ ÇATIMI

KIRLI, Mustafa Yavuz

Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr İlkay ULUSOY

Ağustos 2008, 144 sayfa

Bu tezin amacı, kalibre edilmemiş sualtı videosu kullanarak 3B sahne geri çatımı yapmaktır. Sualtı görüntüleme, uzaktan kumanda edilebilen sualtı araçları için çok önemlidir ve sualtı ışık soğrulması açısından homojen olmaması ve ortamdaki ışık yansımaları nedeniyle görüntü işleme açısından karmaşık bir ortamdır. Bu sınırlayıcı koşullar sualtı sahnesinin 3B geri çatımını daha da zorlaştırmaktadır.

Geri çatım işlemi şu adımlardan oluşmaktadır: Görüntü iyileştirme, köşe nokta bulunması ve eşlenmesi, temel matris kestirimi, oto-ölçümleme, dışsal parametrelerin bulunması, doğrultma, stereo eşleştirme ve üçgenleme.

Görüntü iyileştirme için suyun etkisini kaldıran ve görüntüyü iyileştiren bir filtre yönetimi kullanılmıştır. Köşe nokta bulunması için SIFT ve Harris Köşe Bulma yöntemleri karşılaştırılmıştır ve SIFT yönteminin daha iyi sonuç verdiği

görülmüştür. Köşe noktalarını eşleştirmek için korelasyon yöntemi kullanılmıştır. Epipolar ilişkiyi gösteren temel matris kestirimi için RANSAC ve normalize edilmiş 8-nokta algoritması kullanılmıştır. Sualtında kamera ölçümlemesi zor bir işlem olduğu için kalibrasyon objesi gerektirmeden sahne görüntülerini kullanarak ölçümleme yapabilen oto-ölçümleme yöntemi Kruppa Denklemleri kullanılmıştır. Görüntü üzerindeki epipolar doğruları görüntü koordinat sisteminin yatay eksenine göre paralelleştiren ve aynı zamanda stereo eşleştirme algoritması için gerekli olan doğrultma yöntemi ile görüntüler stereo eşleştirme algoritmaları için uygun hale getirilmiştir. Graph-Cut stereo eşleştirme algoritması ile görüntüdeki her bir pikselin diğer görüntü de karşılığı bulunmuştur. Son aşama olarak üçgenleme yöntemi kullanılarak eşleniği bulunan her bir pikselin karşılık geldiği noktanın 3 boyutlu uzayda koordinatları bulunmuştur.

Anahtar Kelimeler: Sualtı, 3 Boyutlu Geri Çatım, SIFT, Görüntü Doğrultma, Stereo Eşleştirme, Sualtı Görüntü İyileştirme, Üçgenleme, Kruppa Denklemleri

ACKNOWLEDGMENTS

The author wishes to express his deepest gratitude to his supervisor Assist. Prof. Dr. İlkay ULUSOY for her guidance, advice and support throughout the research.

The author would like to thank to his mother Mrs. Nuran KIRLI, his father Mr. Ali KIRLI and his twin sisters Ms. Serap and Ms. Meltem KIRLI for their support, encouragement and patience.

The technical assistance of Mr. Emre EGE is gratefully acknowledged.

The author would also like to thank to his colleagues at Aselsan Inc. for their support, guidance and valuable advice throughout this thesis work and to Aselsan Inc. for the facilities provided for the completion of this thesis.

The author is grateful to Ms. Ebru KÖSEALİOĞLU for her support, encouragement and patience.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

3D reconstruction of the scenes has been studied for 20 years in computer vision literature. Many methods have been developed such as, structured lighting, ultrasonic and laser range finders… etc. Some of the methods mentioned above give high accuracy but cost more, some of them give inaccurate results beside provides no flexibility. By the development in computer vision, it becomes possible to reconstruct a 3D model of the scene from just the video sequence of that scene. Camera moves through the scene while making an arbitrary motion, from the frames of the video, the 3D model is reconstructed via related computer vision algorithms.

Considering the underwater environment, the methods such as structured lighting or laser range finders need more equipment and provide no flexibility and they are hard to apply. Ultrasonic or sonar is widely used in underwater researches. These methods perform perfectly in long range distances. But in short range, they do not provide detailed results like cameras do. For that reason, studies are performed to combine the data extracted from these two type sensors, optical (e.g. camera) and acoustic sensors (e.g. sonar) [27, 28].

Several methods have been developed to reconstruct 3D model from underwater images. Most of them uses calibrated stereo cameras, where the positions of the cameras are known and fixed. In recent years, the stereo systems leave their places to mono cameras. In this situation the motion of the camera becomes more important. In [13] a robot arm is used to move the calibrated camera in a predefined trajectory. In [33] a calibrated camera is used with a system which provides the position, orientation of AUV with magneto-inductive compass, so

that the motion of the camera is known. Fusiello proposed a method in [34] which uses an uncalibrated camera with an arbitrary motion, achieving to an Euclidean reconstruction with a-priori information about the positions of five identifiable scene points.

The aim of this thesis is to reconstruct the 3D model of the underwater scenes using pictures captured from an uncalibrated camera moving in an arbitrary path.

1.2 System Overview

The process of 3D reconstruction is composed of the following sub-steps:

1.2.1 Image Enhancement

Underwater is a complex structured environment for computer vision algorithms. Underwater images suffer from many factors: inhomogeneous environment, limited range, non-uniform lighting, important blur, back-scattering and little particles floating in the water like marine snow. It is necessary to enhance the images before using the image processing algorithms [20]. There are two approaches to solve this problem; physical approach (mount a polarizer to the front part of the camera) [28] and software approach (develop a preprocessing filter) [20]. Since software approach brings more flexibility than the physical approach, the preprocessing filter proposed by [20] is used in this thesis.

1.2.2 Feature Detection and Matching

After preprocessing the images, the first step of the reconstruction process is to determine the corresponding points from the images. The points which are distinctive according to their neighbors, called features, are detected. Many methods have been developed on feature detection. Harris Corner Detector is one of the most famous one. Because of the blurred structure of the water, corner detection does not perform well. For that reason, it is compared with another method called SIFT, provided by [3].

1.2.3 Estimation of Fundamental Matrix

Given the putative correspondences of the feature points, it is possible to estimate the epipolar geometry between the two images. Epipolar geometry

between two images can be represented with a *fundamental matrix*. Fundamental matrix can be estimated with at least 8 corresponding points. But all of the corresponding points may not be matched correctly due to the noise in the images. Therefore false matches in the corresponding points, called *outliers*, cause mistakes in fundamental matrix estimation. The outliers can be removed during the estimation process of fundamental matrix by combining the normalized 8-point algorithm and RANSAC method [7, 32].

1.2.4 Auto-Calibration

The forth step is to recover the camera parameters. Camera calibration is one of the most important steps of reconstruction for a more accurate 3D model. The cameras can be calibrated by using known structured calibration objects. Also methods which do not need calibration objects are developed. This type of methods is called *auto-calibration* or *self-calibration*. The cameras can be calibrated using the images that are captured by themselves. Because of the difficulties in calibrating cameras in underwater and the possibility of change in the calibration of the camera during functioning (e.g. zooming), auto-calibration is used in this thesis. The only required data in auto-calibration is the captured images from different locations and orientations. Many methods are developed for the auto-calibration problem. The most famous one is *Kruppa equations* [30]. Fundamental matrices are used to construct Kruppa equations and minimization algorithms are used to solve the Kruppa equations to determine the unknown calibration parameters. Unknown calibration parameters can also be determined by using the relation between the virtual conic and the calibration parameters. Since the virtual conic is invariant of Euclidean transformations (rotation and translation), its image on the camera only depends on the intrinsic parameters of the camera. Two algorithm representing the two approaches, simplified Kruppa equations [22] based on Kruppa equations and calibration by absolute quadric based on the virtual conic [1] are examined.

1.2.5 Estimation of Camera Motion

Once the fundamental matrix is estimated and the intrinsic parameters of the camera are determined by auto-calibration, the extrinsic parameters of the cameras can be determined. The extrinsic parameters of the cameras include the

rotation and translation of the cameras according to each other. Fundamental matrix is transformed to essential matrix which is only defined by the rotation and the translation of the camera by using the intrinsic parameters. Using the relation between the rotation and translation matrices, first the unit translation matrix is computed and finally the rotation matrix is estimated.

1.2.6 Rectification

Rectification is the process of determining new camera geometry such that the epipolar lines of the cameras are parallel to each other and horizontal axis. Since rectification reduces the search area and computation time, any stereo matching algorithm require rectified images. Two algorithms are examined in rectification: one for calibrated images [9] and the other for the uncalibrated images [10].

1.2.7 Stereo Matching

Stereo matching is the process of finding the correspondence pixel of each pixel in the image. From these correspondences, disparity map is constructed. Using the relation between the disparity and depth, the depth of each pixel is determined. There are several algorithms for stereo matching. The major problem in stereo matching is the homogenous regions (in term of texture) in the image. Finding the corresponding pixel in homogenous parts of the image is very difficult and error-prone. Because of the presence of homogenous areas in underwater images, stereo matching algorithm based on graph cut [14] which perform well in homogenous images is used in this thesis.

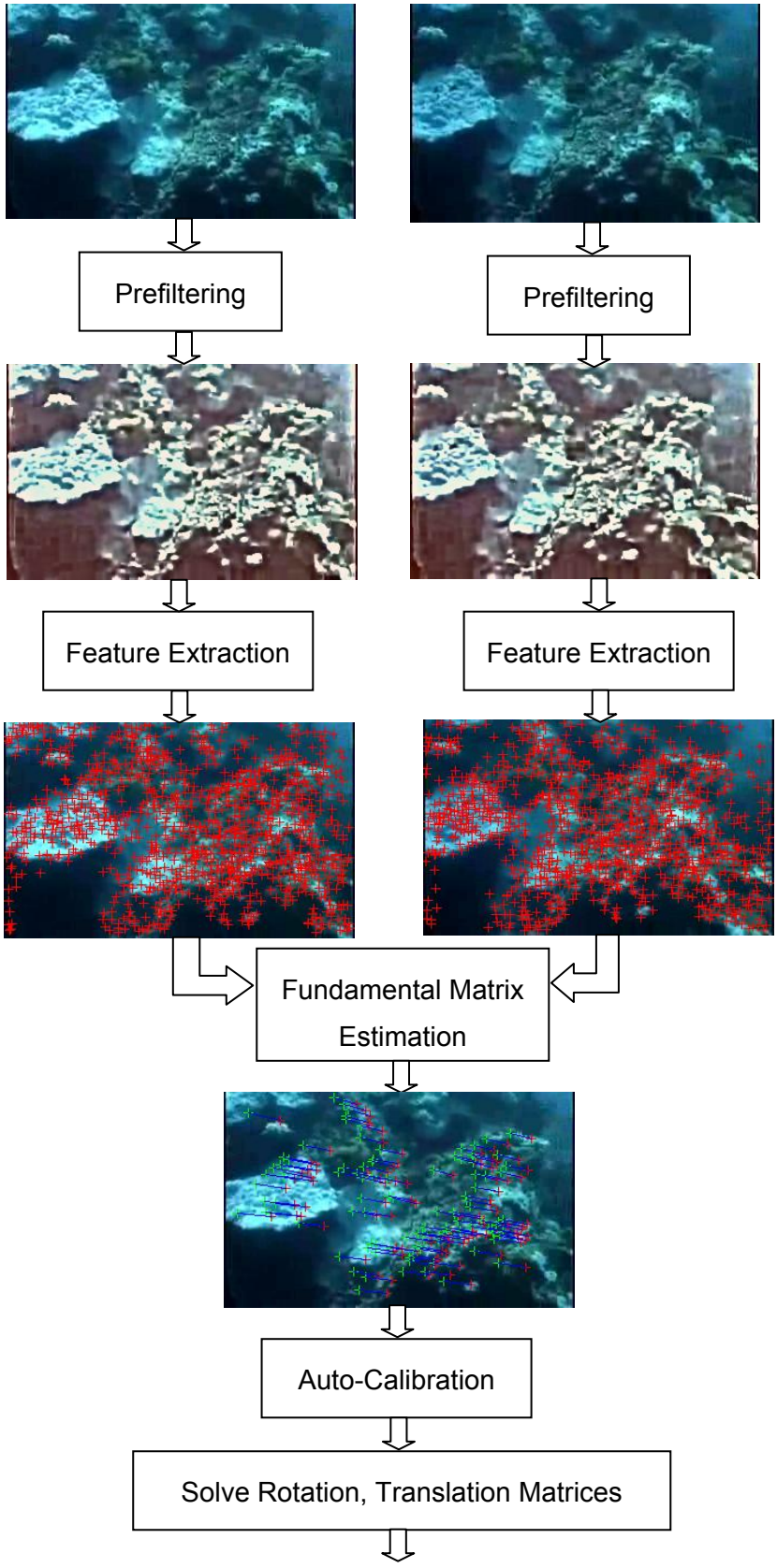
1.2.8 Triangulation

The final step of the reconstruction process is triangulation. Once the calibration matrices, rotation and translation matrices of the image pair are known, 3D coordinate of the point projected to the two images can be determined by back-projecting the rays of the corresponding points. Because of the noise on the images, the rays do not intersect. For that reason an optimal point must be estimated and this estimation process is called *triangulation*. A linear and a robust triangulation algorithm [23] are examined in this chapter.

With triangulation of feature points only, a sparse 3D reconstruction can be achieved. Sparse means that depth is not known for all of the points in the image, but for the corresponding points only. In order to perform dense reconstruction, the projection matrix of the camera and the disparity map constructed in stereo matching are required. A method for dense reconstruction is examined in this chapter and the final 3D model is visualized by VRML (Virtual Reality Modeling Language).

1.3 Scope of the Thesis

The scope of this thesis, summarized in Figure 1, is to develop the blocks of a complete 3D reconstruction system for underwater images. Rather than proposing a new algorithm this thesis is devoted to develop the sub-blocks of the complete 3D reconstruction process and compare different algorithms proposed for each sub-block in order to identify the best performing algorithm for underwater applications.


Prefiltering
Prefiltering
Feature Extraction
Feature Extraction
Fundamental Matrix Estimation
Auto-Calibration
Solve Rotation, Translation Matrices

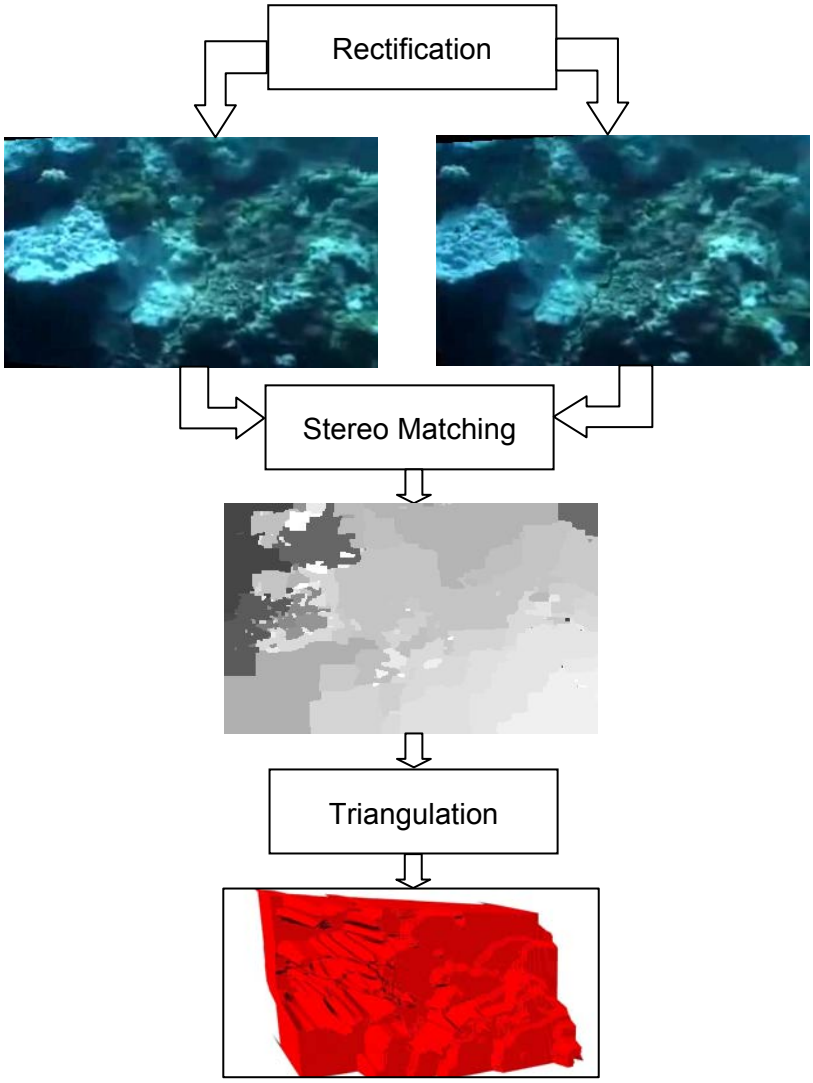


Figure 1 The summary of the 3D reconstruction process.

1.4 Organization

In the first chapter, problem definition, system overview and scope of the thesis are defined.

In Chapter 2 background information about the projective geometry and the camera model used throughout the thesis is provided.

In Chapter 3, underwater image enhancement is discussed and a preprocessing method is examined.

Feature detection and matching is discussed in Chapter 4. Two methods, Harris Corner Detector and SIFT are examined and compared using underwater images.

In Chapter 5, estimation of the fundamental matrix is given. Normalized 8-point algorithm and RANSAC method are examined for fundamental matrix estimation and outlier removal process.

Chapter 6 is devoted to auto-calibration. Camera calibration matrix is one of the important parts to be estimated for an accurate 3D reconstruction. The most famous algorithm, Kruppa equations, and the auto-calibration with virtual conic are examined and compared.

Chapter 7 gives brief information about how to extract rotation and translation matrices of the camera from given fundamental matrix and camera calibration matrix. A linear and a robust algorithm are examined and compared for rotation and translation matrix estimation.

In Chapter 8, rectification is discussed. Rectification is the process of determining new camera geometry such that the epipolar lines of the cameras are parallel to each other and horizontal axis that is necessary for stereo matching algorithms since all stereo matching algorithms require rectified images. Two algorithms, calibrated rectification and uncalibrated rectification, are examined and compared.

Chapter 9 is devoted to stereo matching. After the rectification of the images, the correspondence of each pixel in the image can be determined by stereo matching. Once the correspondence of each pixel of the image is determined, the dense 3D reconstruction of the scene can be computed. Graph-cut based stereo matching algorithms are examined and compared with a traditional stereo matching algorithm.

Chapter 10 gives the last step of the reconstruction which is triangulation. Once the projection matrices of the cameras and the corresponding points are found,

the coordinates of the 3D points are computed by back-projecting the rays from corresponding points. Because of the noise in the images, the rays do not intersect. An optimal solution is computed via triangulation. Triangulation gives sparse 3D model of the scene. Also a method for dense reconstruction is given in this chapter.

Chapter 11 gives the experimental results of the sub-blocks of the reconstruction process on different data sets.

Chapter 12 concludes the thesis with the remark about the sub-blocks and the whole 3D reconstruction process and provides some ideas about future work.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Introduction

In this chapter the geometry behind the projective geometry is discussed to better understand the uncalibrated scene reconstruction. This chapter gives brief information about the Euclidean geometry, the most general geometry without any constraint, Projective geometry, and the relation between the two views, epipolar geometry and the camera model used throughout this thesis. The following topics and definitions mostly follow the text [6, 24]; the details can be followed from these references.

Euclidean geometry is the most familiar geometry to us, since it describes our world. The basic properties of Euclidean geometry are intersecting lines determine angles between them, and two lines are said to be parallel if they lie in the same plane and never meet. Moreover, these properties do not change when the Euclidean transformations (translation and rotation) are applied. But these properties become insufficient when the imaging process of a camera is considered. Lengths and angles are no longer preserved, and parallel lines may intersect.

Euclidean geometry is actually a subset of what is known as *projective geometry*. The following section gives brief information about Projective geometry

2.2 Projective Geometry

N-dimensional projective space, P^n, is defined by a (n+1) vector with coordinates: $x = [x_1 \dots x_{n+1}]^T$. One of these coordinates must be non-zero. This coordinate representation is known as *homogenous coordinates*. In this

representation, two points represented by (n+1) vectors x and y are equal if there exists a non-zero scale factor λ such that $x_i = \lambda y_i$ where $1 \le i \le n+1$. This equality is shown as $x \sim y$.

Projective spaces can be transformed to each other, which is called collineation. This transformation from P^m to P^n is done with a $(m+1) \times (n+1)$ matrix H. Points are transformed linearly $x \rightarrow x' = Hx$.

2.3 Projective Plane

Projective plane is the projective space, P^2. A point in P^2 is defined with 3 dimensional vector $m = [u \;\; v \;\; w]^T$. Also a line is defined with 3 dimensional vector. A point m is located on a line l if $l^T m = 0$. This equation shows that there is no formal difference between points and lines in projective plane. This is known as the principle of *duality* [24].

2.4 Projective 3D-Space

Projective 3D space is the projective space, P^3. A point in P^3 is defined with 4 dimensional vector $M = [X \;\; Y \;\; Z \;\; W]^T$. The dual of a point in P^3 is a plane, Π. If a point is on a plane, then $\Pi^T M = 0$.

2.5 Transformations

Transformations in images, $P^2 \rightarrow P^2$, is performed with homographies represented by 3x3 matrix H. A point and a line are transformed as follows:

$$\begin{aligned} m &\rightarrow m' = Hm \\ l &\rightarrow l' = H^{-1} l \end{aligned} \tag{2.1}$$

In P^3, points and planes are transformed with a 4x4 matrix T:

$$\begin{aligned} M &\rightarrow M' = TM \\ \Pi &\rightarrow \Pi' = T^{-1}\Pi \end{aligned} \tag{2.2}$$

2.6 Conics and Quadrics

A *conic* in P^2 is the locus of all points m satisfying the equation:

$$S(m) = m^T C m = 0 \tag{2.3}$$

where C is a 3x3 symmetric matrix only defined up to a scale.

A *dual conic* in P^2 is the locus of all lines l satisfying the equation:

$$l^T C^* l = 0 \tag{2.4}$$

where C^* is a 3x3 symmetric matrix only defined up to a scale.

When m varies along the conic C, it satisfies $m^T C m = 0$, also at the same time the tangent line l to the conic at m satisfies $l^T C^{-1} l = 0$. This relation shows that the tangents of a conic C belong to a dual conic $C^* \sim C^{-1}$ [24].

The transformations of conics and dual conics can be written as:

$$\begin{aligned} C &\rightarrow C' \sim H^{-T} C H^{-1} \\ C^* &\rightarrow C^{*'} \sim H C^* H \end{aligned} \tag{2.5}$$

A *quadric* in P^3 is the locus of all points M satisfying the equation:

$$M^T Q M = 0 \tag{2.6}$$

where Q is a 4x4 symmetric matrix only defined up to a scale.

A *dual quadric* in P^3 is the locus of all planes Π satisfying the equation:

$$\Pi^T Q^* \Pi = 0 \tag{2.7}$$

When M varies along the quadric Q, it satisfies $M^T Q M = 0$, also at the same time the tangent plane Π to the quadric at M satisfies $\Pi^T Q^{-1} \Pi = 0$. This relation shows that the tangents of a quadric Q belong to a dual quadric $Q^* \sim Q^{-1}$ [24].

The transformations of quadric and dual quadric can be written as:

$$\begin{aligned} Q &\rightarrow Q' \sim T^{-T} Q T^{-1} \\ Q^* &\rightarrow Q^{*'} \sim T Q^* T \end{aligned} \tag{2.8}$$

2.7 The stratification of 3D Geometry

World is described by the Euclidean geometry, but in computer vision it is sometimes not desired to use to the full Euclidean structure. Instead of it, a less structured and simple projective geometry is used. Affine and metric geometry forms the intermediate layers.

The vision geometry is stratified to the stratums (layers) related to the transformation actions on geometric entities and invariants belong to that group. Projective stratum is the group of projective transformations; the affine stratum is the group of affine transformations; the metric stratum is the group of similarities and the Euclidean stratum is the group of Euclidean transformations and these groups are subgroups of each other [24].

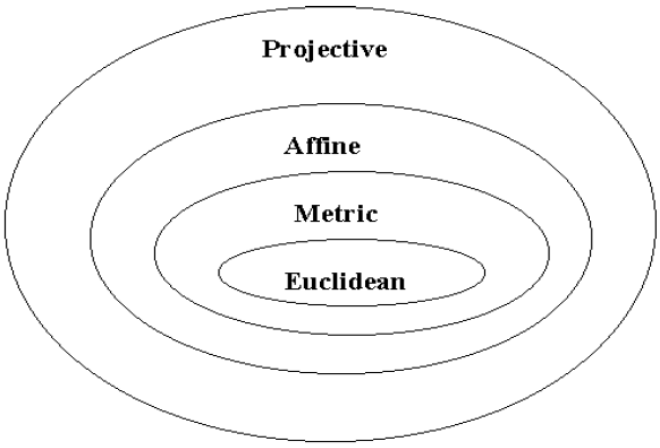


Figure 2 Hierarchy of geometries. [19]

As explained above, one of the important properties of these groups are their *invariants*. An *invariant* is a property of a configuration of geometric entities that is not altered by a transformation belonging to a specific group. The structure of a geometry can be upgraded to a higher geometry by computing there invariants. In the following section, each stratum, its invariants, transformations are discussed in detail.

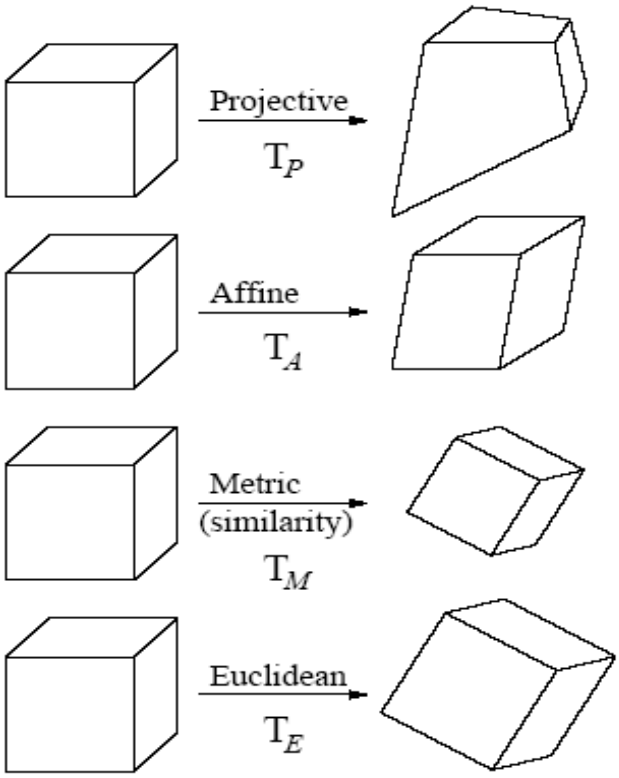


Figure 3 Shapes which are equivalent to a cube for the geometric strata projective, affine, metric and Euclidean. The reason of the deformation on the shape of the cube depends on the ambiguity of the related geometric stratum [24].

2.7.1 Projective Stratum

The first stratum is the projective stratum. It is the less structured one and therefore includes the least number of invariants. In contrary projective stratum has the largest number of transformations. The projective transformation of a 3D space point can be represented with a 4x4 invertible matrix [24].

$$T_p \sim \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} \tag{2.9}$$

Since this transformation is defined up to non-zero scale factor, it has 15 degrees of freedom. The invariants of projective stratum are relations of incidence,

collinearity, tangency and cross-ratio. The cross-ratio is defined as follows: assume four points, M_1, M_2, M_3, M_4 are collinear. They can be defined as $M_i = M + \lambda_i M'$.

$$\{M_1, M_2; M_3, M_4\} = \frac{\lambda_1 - \lambda_3}{\lambda_1 - \lambda_4} : \frac{\lambda_2 - \lambda_3}{\lambda_2 - \lambda_4} \tag{2.10}$$

2.7.2 Affine Stratum

The affine stratum lies between the projective and metric stratum. It contains more structure and invariants according to the projective one, but less than the metric one [24]. The specialty of affine stratum is defining the *plane at infinity*, $\Pi_\infty = [0 \;\; 0 \;\; 0 \;\; 1]^T$. The affine transformation is defined as, $M' \sim T_A M$:

$$T_A \sim \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.11}$$

Since the plane at infinity is the invariant of affine stratum, the affine transformation can not change the plane at infinity: $\Pi_\infty \sim T_A \Pi_\infty$. Also another invariant is added with affine transformation, *parallelism*. Lines or planes having intersection at infinity are called *parallel* [24].

The transformation from projective representation to affine representation is achieved by finding the plane at infinity. The position of plane at infinity can be computed by using the parallelism constraint.

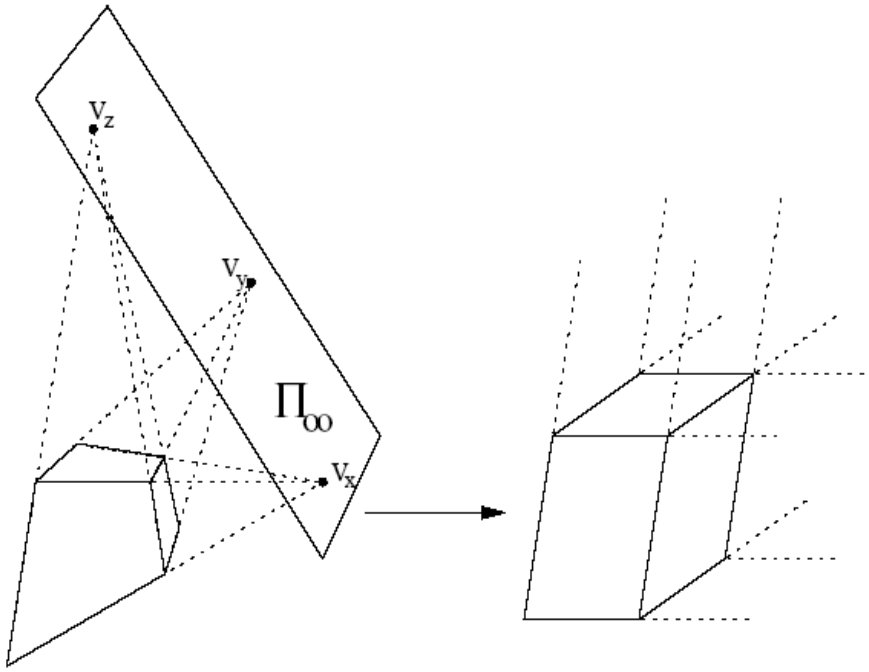


Figure 4 A cube defined in projective (left) and affine (right) stratum. Each figure is equal to a cube under their ambiguities. The plane at infinity can be found by using the vanishing points which are the points where the parallel sides of the cube intersects. This is used to transform from projective to affine stratum [24].

Once the location of the plane at infinity is found, the transformation from projective to affine is defined as:

$$T_{PA} \sim \begin{bmatrix} I_{3x3} & 0_3 \\ \pi_\infty & 1 \end{bmatrix} \tag{2.12}$$

where π_∞ is the plane at infinity.

2.7.3 Metric Stratum

Metric stratum can be defined as the group of similarities. Metric transformations correspond to the Euclidean transformations (orthonormal transformation and translation) up to a scale factor. The metric transformation can be defined as:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \sigma \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_{14} \\ t_{24} \\ t_{34} \end{bmatrix} \tag{2.13}$$

The coefficients r_{ij} define the rotation matrix that has the properties: $RR^T = I$. The metric transformation can be rewritten as:

$$T_M \sim \begin{bmatrix} \sigma r_{11} & \sigma r_{12} & \sigma r_{13} & t_X \\ \sigma r_{21} & \sigma r_{22} & \sigma r_{23} & t_Y \\ \sigma r_{31} & \sigma r_{32} & \sigma r_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

The new invariants with metric stratum are *relative lengths* and *angles*, which corresponds to a new geometric entity, *absolute conic*. It can be defined as an imaginary circle located in the plane at infinity. The absolute conic can be defined as:

$$\Omega = X^2 + Y^2 + Z^2 = 0 \text{ with } W = 0 \tag{2.15}$$

and also it can be defined as a 2D conic:

$$\omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \omega^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$

The dual entity of absolute conic, absolute dual quadric Ω^* is defined as:

$$\Omega^* \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.17}$$

Since the absolute quadric is the invariant of metric stratum, metric transformations must leave it unchanged.

$$\begin{bmatrix} I_{3x3} & 0_3 \\ 0_3^T & 0 \end{bmatrix} \sim \begin{bmatrix} A & b \\ c^T & d \end{bmatrix} \begin{bmatrix} I_{3x3} & 0_3 \\ 0_3^T & 0 \end{bmatrix} \begin{bmatrix} A & c \\ b^T & d \end{bmatrix} \sim \begin{bmatrix} AA^T & Ac \\ c^T A^T & c^T c \end{bmatrix} \quad (2.18)$$

Hence, $AA^T \sim I_{3x3}$ and $c = 0_3$ which are constraints for metric transformation.

The transformation from affine to metric is achieved by finding the absolute conic. Every angle and ratio of length gives a constraint about the absolute conic. Once an affine reconstruction is done, there must be an affine transformation which brings the absolute quadric to its canonical position satisfying the relation: [19, 24]

$$\Omega^* \sim \begin{bmatrix} A & a \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} I_{3x3} & 0_3 \\ 0_3^T & 0 \end{bmatrix} \begin{bmatrix} A^T & 0_3 \\ a^T & 1 \end{bmatrix} = \begin{bmatrix} AA^T & 0_3 \\ 0_3^T & 0 \end{bmatrix} \quad (2.19)$$

The 2D representation of absolute conic and its dual can be rewritten as:

$$\omega_\infty \sim A^{-T} A^{-1} \qquad \omega_\infty^* \sim AA^T \quad (2.20)$$

The transformation from affine to metric is defined as:

$$T_{AM} = \begin{bmatrix} A^{-1} & 0_3 \\ 0_3^T & 0 \end{bmatrix} \quad (2.21)$$

where A can be calculated from Ω^* via Cholesky decomposition or singular value decomposition [24].

The transformation from projective to metric directly can be written as:

$$T_{PM} = T_{AM} T_{PA} = \begin{bmatrix} A^{-1} & 0_3 \\ \pi_\infty^T & 0 \end{bmatrix} \quad (2.22)$$

2.7.4 Euclidean Stratum

The only difference between metric and Euclidean stratum is the scale factor is fixed in Euclidean stratum. Hence in Euclidean stratum, *absolute lengths* are measured not the *relative ones*. The transformation of the Euclidean stratum is defined as [24]:

$$T_E \sim \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.23}$$

In Table 1, all of the properties of the stratums are summarized. It is seen from the Table 1, while moving from Projective to Euclidean stratum, the ambiguity in the model decreases.

Table 1 The transformations, invariants and degrees of freedom of projective, affine, metric and Euclidean stratum [24].

Ambiguity	DOF	Transformation	Invariants
Projective	15	$T_p \sim \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$	Cross-ratio
Affine	12	$T_A \sim \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$	Relative Distances along direction Parallelism Plane at infinity

Metric	7	$T_M \sim \begin{bmatrix} \sigma r_{11} & \sigma r_{12} & \sigma r_{13} & t_X \\ \sigma r_{21} & \sigma r_{22} & \sigma r_{23} & t_Y \\ \sigma r_{31} & \sigma r_{32} & \sigma r_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	Relative distances Angles Absolute Conic
Euclidean	6	$T_E \sim \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	Absolute Distances

2.8 Camera Model

In this section, the camera model used in this thesis is discussed. The following topics and definitions mostly follow the text [6, 24]; the details can be followed from these references.

The most basic camera model, *pinhole camera model*, is used in this thesis. In pinhole camera model, a 3D point in space is projected onto a 2D image plane by drawing a line from 3D point to the center of the camera. Where the line intersects with the image plane is the location of the 2D projection of 3D point.

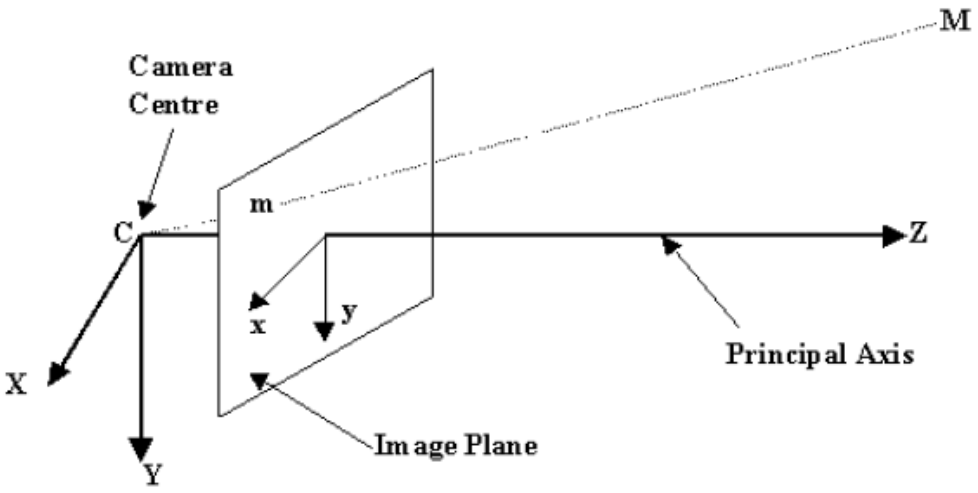


Figure 5 Pinhole Camera Geometry

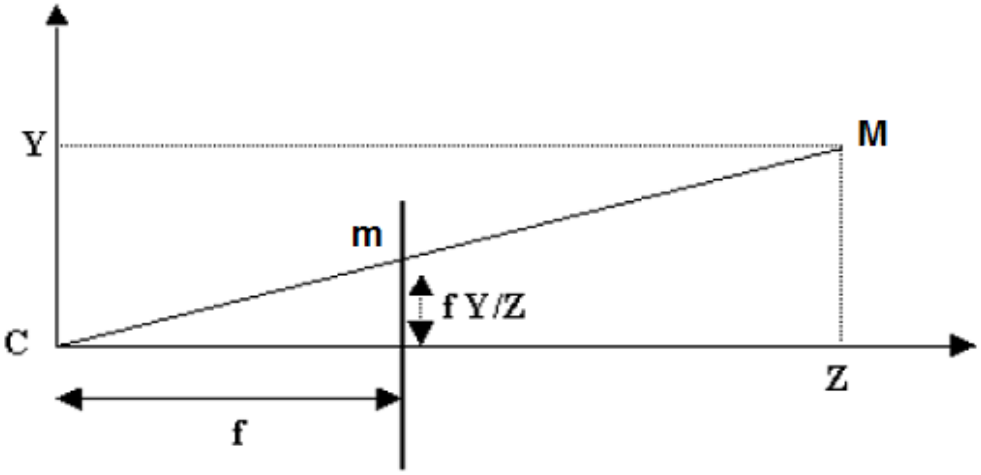


Figure 6 Side-view of the projection of a 3D point M onto the image plane.

The projection is shown in Figure 5, and Figure 6. In the figures **C** is the camera center; **p** is the principal point; **f** is the focal length. The ray, which is perpendicular to the image plane and passing through the camera center, is the principal axis. The intersection of the principal axis and the image plane is the principal point.

A 3D point $M = [\, X \;\; Y \;\; Z \,]^T$ is projected to the 2D point m . Using the similar triangles the projected coordinates are calculated as:

$$m = [\, f \frac{X}{Z} \;\; f \frac{Y}{Z} \,]^T \tag{2.24}$$

Using the homogeneous coordinates the transformation can be rewritten as:

$$\begin{bmatrix} fX \\ fY \\ f \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.25}$$

2.8.1 Principal Point Offset

In Equation 2.24 it is assumed that the origin of the coordinates in the image plane is the principal point. But in practice, instead of it lower left corner is used and the mapping is updated as accordingly:

$$[\, X \;\; Y \;\; Z \,]^T \rightarrow [\, f \frac{X}{Z} + p_x \;\; f \frac{Y}{Z} + p_y \,]^T \tag{2.26}$$

In homogenous coordinates

$$\begin{bmatrix} fX + p_x Z \\ fY + p_y Z \\ f \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.27}$$

$$m = K [\, I \;\; 0 \,] M_{cam} \quad K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \tag{2.28}$$

The matrix K is called the *camera calibration matrix*. In Equation 2.28 it is assumed that the camera is located in the origin of the Euclidean coordinate system with the principal axis, and the point M_{cam} is expressed in this system [6].

2.8.2 Camera Rotation and Translation

Up to now two coordinate systems are mentioned, camera coordinate system and world coordinate system. In general the points in space are represented with the world coordinate system. The transformation between these systems is based on the rotation and translation as seen in Figure 7.

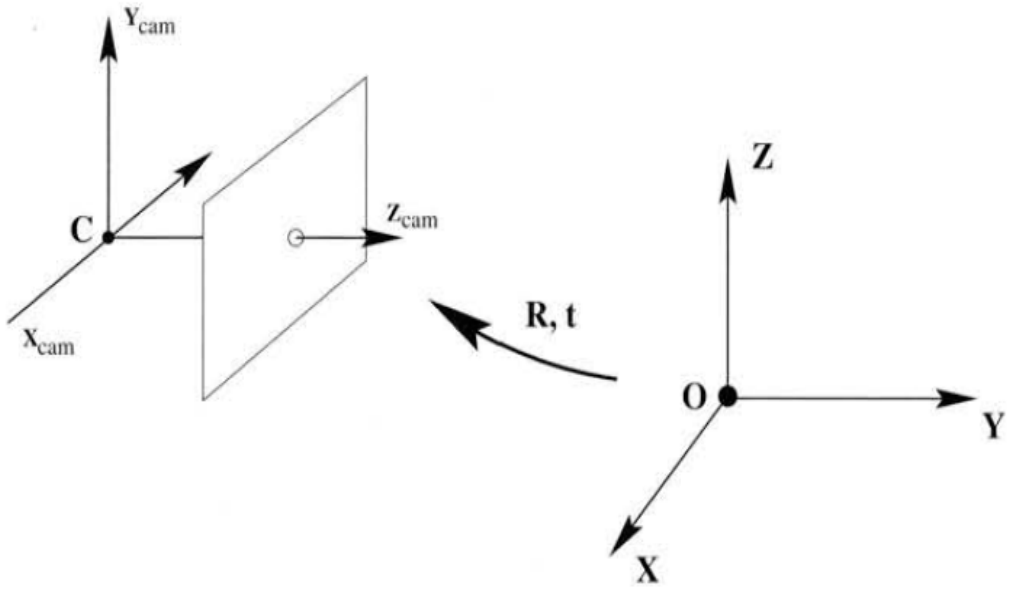


Figure 7 The Euclidean transformation from world coordinate system to camera coordinate system [6].

The transformation between the 3D points $\tilde{M}_{cam}$ and $\tilde{M}$ can be written as $\tilde{M}_{cam} = R(\tilde{M} - \tilde{C})$ where $\tilde{C}$ is the camera center in the world coordinate system and R is the 3x3 rotation matrix.

$$M_{cam} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} M \tag{2.29}$$

Substituting the Equation 2.29 with Equation 2.28 results as:

$$m = K[I\ 0]M_{cam} = K[I\ 0] \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} M = K[R\ -R\tilde{C}]M \tag{2.30}$$

The above equation can be rewritten as:

$$\begin{aligned} m &= K[R\ -R\tilde{C}]M \\ m &= K[R\ t]M \end{aligned} \tag{2.31}$$

where the translation t is equal to $-R\tilde{C}$. The projection matrix of the camera which projects the 3D points in space to 2D plane can be defined as:

$$m = PM \Rightarrow P = K[R\ t] \tag{2.32}$$

CHAPTER 3

UNDERWATER IMAGE ENHANCEMENT

3.1 Introduction

Underwater is a complex structured environment for image processing algorithms. Optical cameras are used for short-range operations in remotely operated underwater vehicles. But the underwater visualization suffers from limited range, non uniform lighting, low contrast, blurring and marine snow, floating small white particles [20]. So a pre-processing step that will remove the effects of water and enhance the image is necessary before processing the underwater images. Usually methods need parameter tuning or human interaction during processing, but in [20] a parameter-free pre-processing method is proposed which reduces the water effects and improves image quality and the method proposed by [20] is used in this thesis.

The algorithm is composed of the following successive independent steps:

3.2 Removing Moire Effect

Moire effect is a wavy repetitive pattern on the image. The reason of this effect is the aliasing in the digital camera. It is removed by detecting the peaks of the Fourier transform and deleting them [20]. The importance of removing the moiré effect is that the following steps increase the contrast, also the moiré effect, and this increases the chance of degraded results.

3.3 Resizing the Image

Resizing the image into a square form enables the usage of fast Fourier transform, fast wavelet transform algorithms and speeds up the process. In [20] the images are resized to a square form.

3.4 Color Space Conversion

The color space of the image is converted from RGB to YCbCr. In YCbCr color space, only the luminance channel (Y) is processed. This step increases the speed of the process by processing only one channel instead of processing the three channels in RGB color space [20].

3.5 Homomorphic Filtering

The homomorphic filter is used to correct the non-uniform lighting and enhance the contrast [20]. An image can be modeled as a product of illumination and reflectance.

$$f(x, y) = i(x, y)r(x, y) \tag{3.1}$$

where $f(x, y)$ is the image, $i(x, y)$ is the illumination factor and $r(x, y)$ is the reflectance factor. Since the illumination changes slowly throughout the image, it is assumed as the low frequency component in the Fourier transform and reflectance is the high frequency component. The non-uniform illumination is suppressed via high pass filtering.

First the logarithm of the image is taken. So the multiplicative effects turn to additive ones.

$$g(x, y) = \ln(\ f(x, y)) = \ln(\ i(x, y)) + \ln(\ r(x, y)) \tag{3.2}$$

The Fourier transform of the image is:

$$G(w_x, w_y) = I(w_x, w_y) + R(w_x, w_y) \tag{3.3}$$

The high pass filter is defined as:

$$H(w_x, w_y) = (r_H - r_L)(1 - \exp(-(\frac{w_x^2 + w_y^2}{2\sigma_w^2}))) + r_L \tag{3.4}$$

where r_H is 2.5, r_L is 0.5, (the maximum and minimum coefficients values) and σ_w is 32 (the factor that controls the cut-off frequency). These values are selected empirically [20].

3.6 Wavelet De-noising

The Gaussian noise present in the image is amplified by the previous step, homomorphic filtering. A further de-noising step is necessary to remove the amplified noise. Wavelet based filtering is used to remove noise, because of its performace compared to other similar algorithms [20].

3.7 Anisotropic filtering

Anisotropic filter smoothes the image in the homogeneous regions, while preserving the edges. The intensities of the pixels are re-calculated considering their 4-neighbor pixels [20]. For every pixel, the nearest-neighbor differences and the diffusion coefficients in the four directions North, South, East, West are computed [20].

$$\nabla_N I_{i,j} = I_{i-1,j} - I_{i,j} \qquad \nabla_S I_{i,j} = I_{i+1,j} - I_{i,j}$$

$$\nabla_E I_{i,j} = I_{i,j+1} - I_{i,j} \qquad \nabla_W I_{i,j} = I_{i,j-1} - I_{i,j}$$

$$c_{N_{i,j}} = g\left(\left|\nabla_N I_{i,j}\right|\right) \qquad c_{S_{i,j}} = g\left(\left|\nabla_S I_{i,j}\right|\right)$$

$$c_{E_{i,j}} = g\left(\left|\nabla_E I_{i,j}\right|\right) \qquad c_{W_{i,j}} = g\left(\left|\nabla_W I_{i,j}\right|\right)$$

where $\nabla_d I$ is the nearest-neighbor difference in direction d, $c_{d_{i,j}}$ is the diffusion coefficient and $g(\nabla I) = \exp\left(-\left(\left\|\frac{\nabla I}{\mathrm{K}}\right\|\right)^2\right)$ and K is 0.1.

The updated pixel intensity is:

$$\hat{I}_{i,j} = I_{i,j} + \lambda[c_N \nabla_N I + c_S \nabla_S I + c_W \nabla_W I + c_E \nabla_E I]_{i,j}$$

3.8 Image Intensity Adjustment

Adjusting the image intensity provides increase in contrast. After the anisotropic filtering some pixels intensity values can exceed the valid range or a non-uniform distribution of pixel values can occur considering the full valid range. The range of intensity is stretched so that the intensity of the pixels is spread to a full range. The following condition is applied to all pixels:

$$I_{i,j} = \frac{I_{i,j} - \min_I}{\max_I - \min_I} \quad if \;\; 0 < I_{i,j} < 1$$
$$I_{i,j} = 0 \qquad\qquad if \;\; 0 > I_{i,j}$$
$$I_{i,j} = 1 \qquad\qquad if \;\; I_{i,j} > 1$$

where $I_{i,j}$ is the pixel value of image I in the coordinates (i, j), $\max_I$ and $\min_I$ are the maximum and minimum pixel values in image I respectively.

3.9 Re-converting the Color Space and Equalizing Color Mean

The image is converted from YCbCr color space back to RGB. Also the image is resized to its original size. The color channels are not balanced because of the nature of the underwater images [20]. For each RGB channel, the difference between the mean and median of the channel is added to each pixel.

3.10 Results and Conclusion

The performance of preprocessing method is examined on real images and the results are given in Figure 8, 9 and 10. The red crosses in the images show the features detected by SIFT in the images before and after preprocessing.

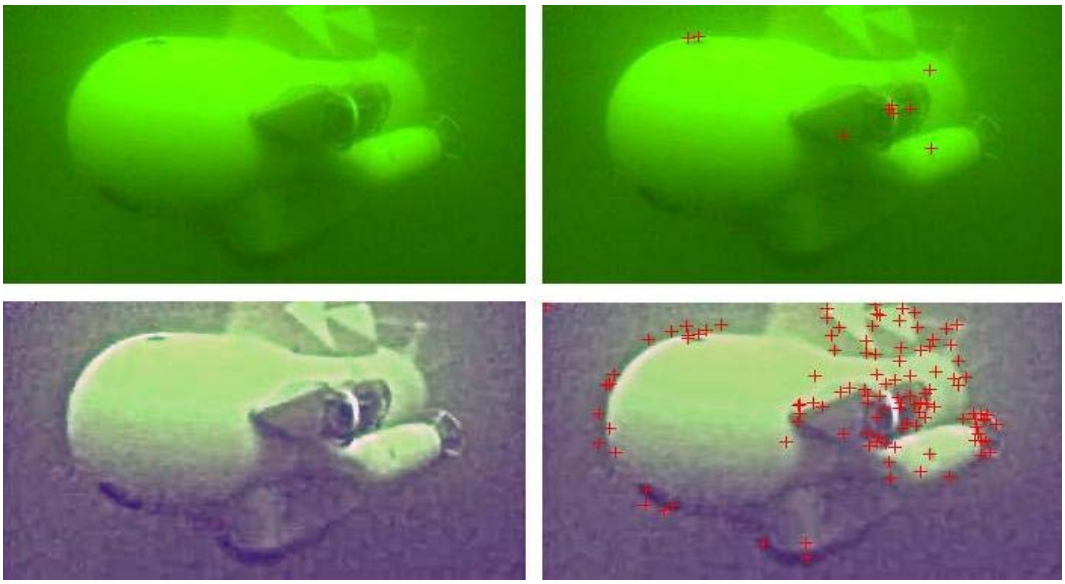


Figure 8 The Boat Sequence. The upper left is the image before preprocessing, lower left is the image after preprocessing.

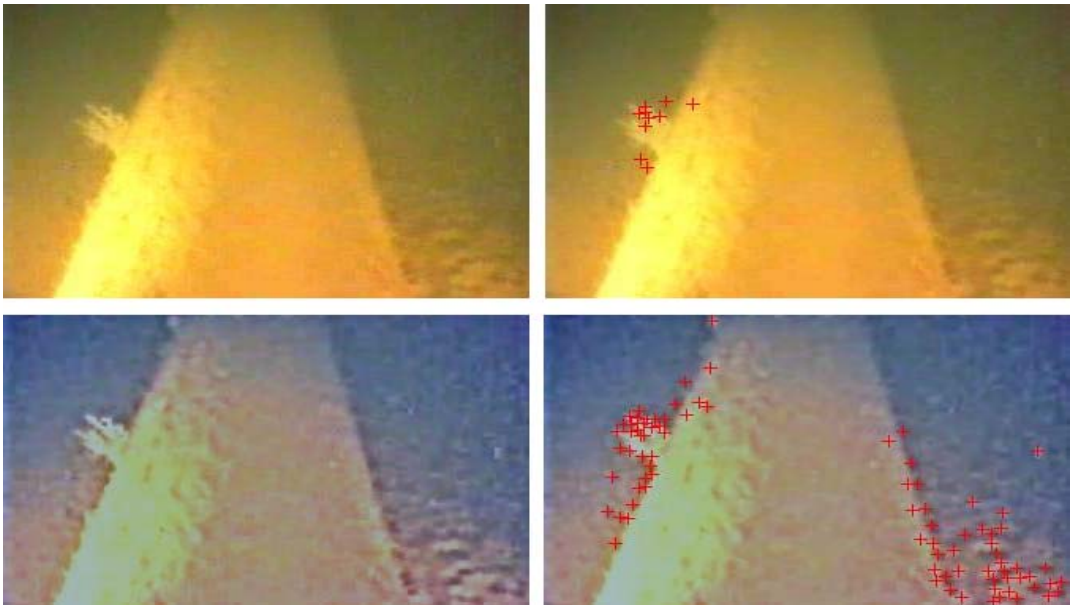


Figure 9 The Pipe Sequence. The upper left is the image before preprocessing, lower left is the image after preprocessing.

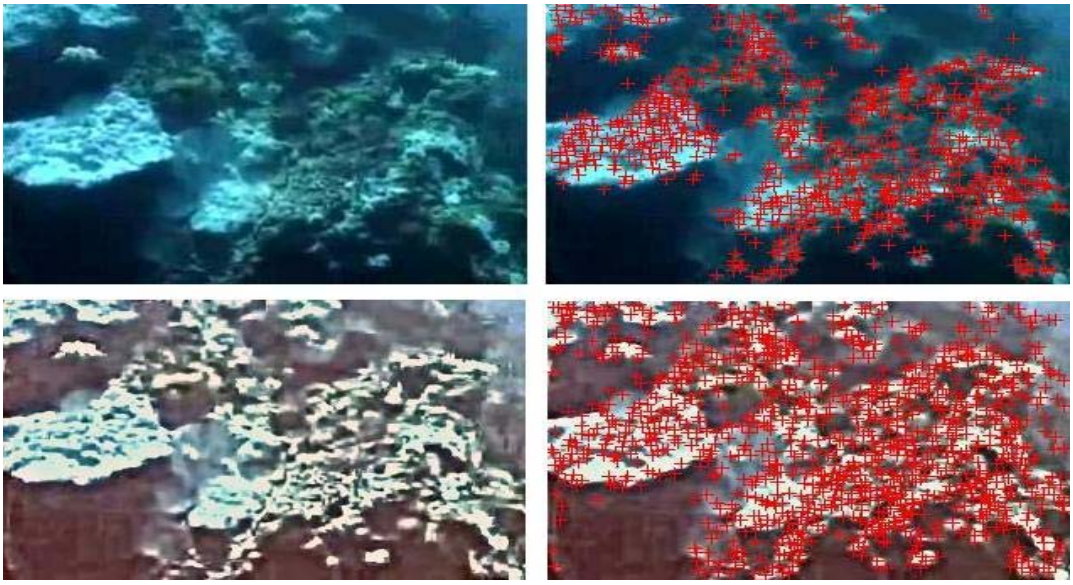


Figure 10 The Coral Sequence. The upper left is the image before preprocessed, lower left is the image after preprocessed.

As it is seen in Figure 8, 9 and 10, the preprocessing method removes the water effects and enhances the images. The edges are clearer and there is a significant improvement in color histogram. The goal of the preprocessing method is to enhance the image and increase the number of detected edges which is crucial for the detection of the epipolar geometry.

In all sequences the number of detected features is significantly increased by the preprocessing method. For comparison purposes, the number of detected feature before and after preprocessing is listed in Table 2 for 3 image sets.

Table 2 The number of detected features with SIFT method, before and after applying the preprocessing method to the input sequences.

Input Sequence	# of Features	
	Before	After
Boat	11	127
Pipe	11	103
Coral	934	1305

CHAPTER 4

FEATURE DETECTION AND MATCHING

4.1 Introduction

The first stage of reconstruction process is feature detection and matching in order to recover the relative geometry between the images. Feature points are the special points that can be differentiated from their neighboring pixels, so that it is possible to match them uniquely with the corresponding pixels in the other image [1].

Since underwater is a complex environment with poor illumination and the camera movement is not deterministic, features between two images suffers from illumination change, rotation and translation. This makes the feature detection and matching process more difficult and decreases its accuracy. For that reason, the feature detection and matching method should cope with these difficulties.

Many methods are developed to find feature points in different ways and one of the most famous one is Harris corner detector [2]. Harris corner detector is based on image gradient evaluation and this makes the method illumination invariant and insensitive to the transformations defined above. Another feature detection method, SIFT, is also illumination, transformation and scale invariant [3].

Harris corner detector and SIFT method and Keypoint descriptor [3], an improved version of correlation-based feature matching method [4] are examined in this thesis.

4.2 Harris Corner Detector

Harris corner detector is based on image gradient evaluation. The points which have high gradient through x and y directions are defined as feature points.

Image regions are typically compared using sum-of-squared-differences (SSD) for matching purposes. Considering a window *W* in image *I* and a corresponding region **T**(W) in image *J*, the *dissimilarity* between two image regions based on SSD is given by [1]:

$$D = \iint_W \left[J(T(x, y)) - I(x, y) \right]^2 w(x, y)\, dxdy \tag{4.1}$$

The approximation of *dissimilarity* between an image window **W** and a slightly translated image window is represented as [2] :

$$D(\Delta x, \Delta y) = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} M \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \tag{4.2}$$

where M is defined as:

$$M = \iint_W \begin{bmatrix} \dfrac{\partial I}{\partial x} \\ \dfrac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} \dfrac{\partial I}{\partial x} & \dfrac{\partial I}{\partial y} \end{bmatrix} w(x, y)\, dxdy \tag{4.3}$$

where $\partial I / \partial x$ and $\partial I / \partial y$ represents the derivative of image *I* in x and y directions respectively and $w(x, y)$ represents the Gaussian smoothing filter:

$$w(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \tag{4.4}$$

The typical value of σ is 0.7. The window size is 7x7.

The desired result is to have large eigenvalues for *M* matrix. The magnitude of eigenvalue represents the intensity change around the pixel. If the two eigenvalues of *M* matrix are small, this means that the windowed image region is approximately constant intensity. If one eigenvalue is high and the other one is low, this indicates an edge. If both of the eigenvalues are high, this indicates a corner [2].

The method Harris cornerness measure [2] to find corners without calculating the eigenvalues is given below:

$$R(x, y) = \det(\hat{C}) - k\ trace^{2}(\hat{C}) \tag{4.5}$$

where $\hat{C}$ is defined as

$$\hat{C} = \begin{bmatrix} \hat{I}_x & \hat{I}_x\hat{I}_y \\ \hat{I}_x\hat{I}_y & \hat{I}_y \end{bmatrix} \tag{4.6}$$

where $\hat{I}$ represents the Gaussian smoothing of grey-level image $I(x, y)$. I_x and I_y represents the derivates in the x and y direction respectively.

k is set to 0.04 according to the suggestion of Harris [2].

Since the original Harris cornerness measure did not give satisfactory results, the Harris cornerness measure defined in [5] is used in the thesis:

$$R(x,y) = \frac{\det\left(\hat{C}\right)}{[\hat{I}_x^2 + \hat{I}_y^2 + e]} \tag{4.7}$$

For comparison purposes, feature points are detected by the original Harris corner measure and corner measure modified by Nobel in [5] are shown in Figure 11. The false feature points detected by the original Harris corner detector are eliminated by the modified Harris corner detector.

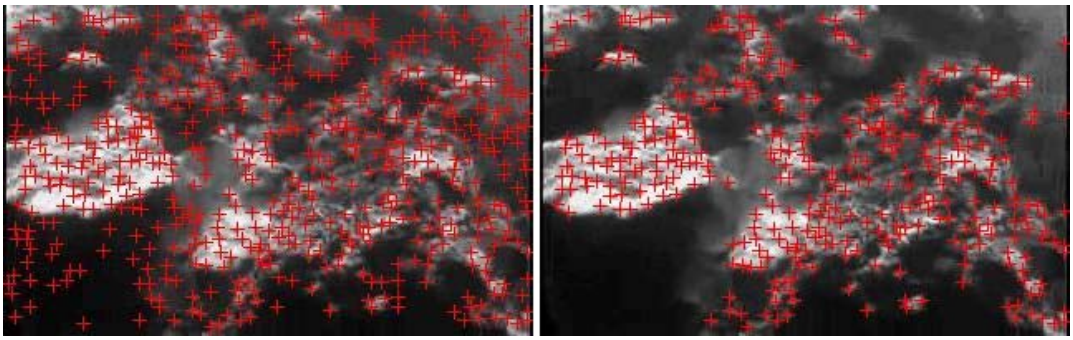


Figure 11 The left image shows the feature points detected by original Harris corner measure with false feature points. The right image shows the feature points detected by modified Harris corner measure by Nobel [5].

4.3 Normalized Cross Correlation

Normalized cross correlation is widely used in matching feature points. The correspondence of given feature point is searched in a determined area, where the pixel can traverse, with a pre-defined NxN square window. The center of the search area is the coordinates of the given feature point. Normalized Cross Correlation is defined as [1]:

$$S(x,y)=\frac{\iint_W (J(T(x,y))-\bar{J})(I(x,y)-\bar{I})w(x,y)dxdy}{\sqrt{\iint_W (J(T(x,y))-\bar{J})w(x,y)dxdy}\sqrt{\iint_W (I(x,y)-\bar{I})w(x,y)dxdy}} \quad (4.8)$$

where

$$\bar{J}=\iint_W J(T(x,y))dxdy \quad \bar{I}=\iint_W I(T(x,y))dxdy \quad (4.9)$$

are the mean image intensity for I and J in the search window. $S(x,y)$ defines the similarity matrix of the feature point searched in the defined search area of the other image.

The performance of the Normalized Cross Correlation (NCC) is good at the images which are slightly transformed. If we consider adjacent frames in a video, small translation constraint is met. But the complex structure of underwater increases the false matches of NCC.

Figure 12 represents the feature points and the feature matches computed with Normalized Cross Correlation. Although the features make a horizontal translation between the two frames, NCC found a few vertical translated feature matches which are false matches and are called as outliers.

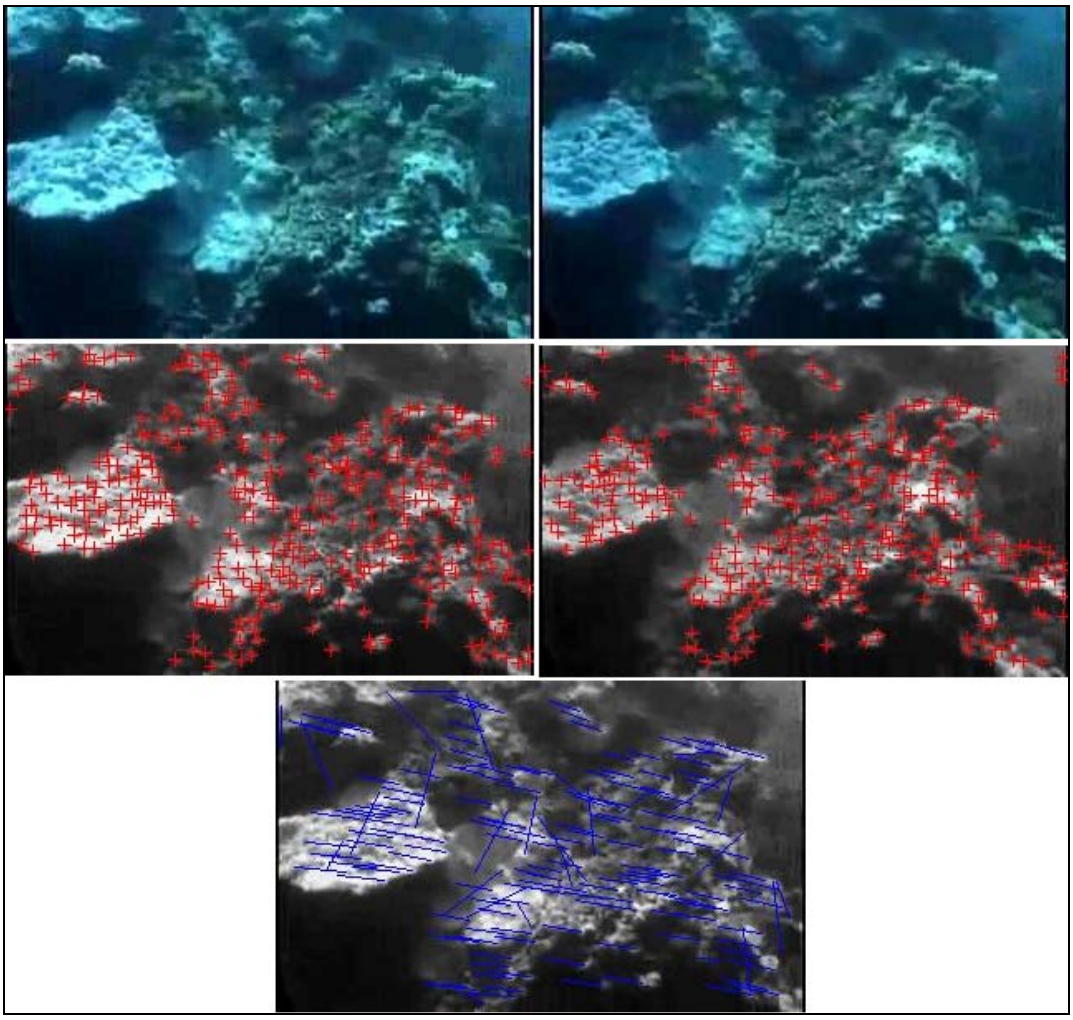


Figure 12 The upper left and upper right images are the two frames from coral sequence. The red crosses show the feature points with Harris Corner Detector in the middle left and middle right images. The last image shows the matches with normalized cross correlation between the feature points of the image above. The blue line represents the translation between the matched features. All of the features in the images make a horizontal translation. The vertical blue lines are the false matches (outliers).

4.4 Scale Invariant Feature Transform (SIFT)

SIFT is a method of extracting distinctive invariant features from images. These features are invariant to illumination, scale, rotation and change in 3D viewpoint, which provide robust matching [3].

SIFT method is performed under 4 stages:

4.4.1 Scale-space extrema detection

The first stage is the interest point detection. Interest points are called keypoints in SIFT framework. Keypoints invariant to scale and orientation are detected by extremum points of Difference of Gaussian. It is shown that under reasonable assumptions the only possible scale-space kernel is the Gaussian function. [3] Therefore, the scale-space of an image is defined as:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{4.10}$$

where $L(x, y, \sigma)$ is the scale-space of the image $I(x, y)$ and $G(x, y, \sigma)$ is the Gaussian function with scale σ :

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{4.11}$$

The difference of Gaussian function is defined as:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{4.12}$$

The computation of $D(x, y, \sigma)$ is shown in Figure 13. The input image is incrementally convolved with Gaussian to produce images separated by a constant scale factor k in scale space shown as stacked layers in the left column. Each octave of scale space is divided into an integer number, s, of intervals, so $k = 2^{1/s}$. S + 3 images must be produced in the stack of blurred images for each octave, so the final extrema detection covers a complete octave.

Adjacent image scales are subtracted to produce the difference of Gaussian images shown on the right. Once a complete octave has been processed, the Gaussian image that has twice the initial value of σ is resampled by taking every second pixel in each row and column.

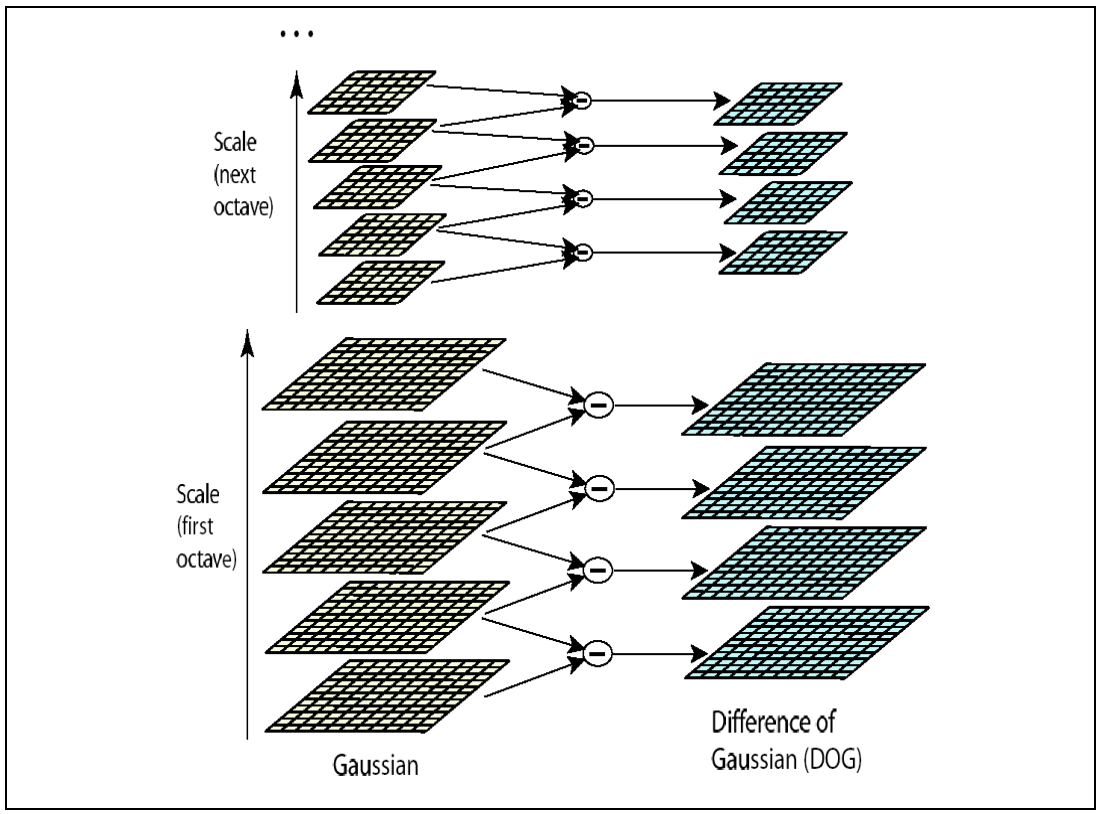


Figure 13 The input image is incrementally convolved for each octave as shown in the left. Adjacent image scales are subtracted to produce the difference of Gaussian images (shown on the right) [3].

The keypoints are identified as the local maxima and minima of $D(x, y, \sigma)$ and the keypoint detection is done by comparing each pixel in the DoG images to its eight neighbor pixels in the same scale and the nine neighbors in the scale above and below. If the pixel is the maximum or minimum of its 26 neighbor pixels, then it is selected as keypoint candidate.

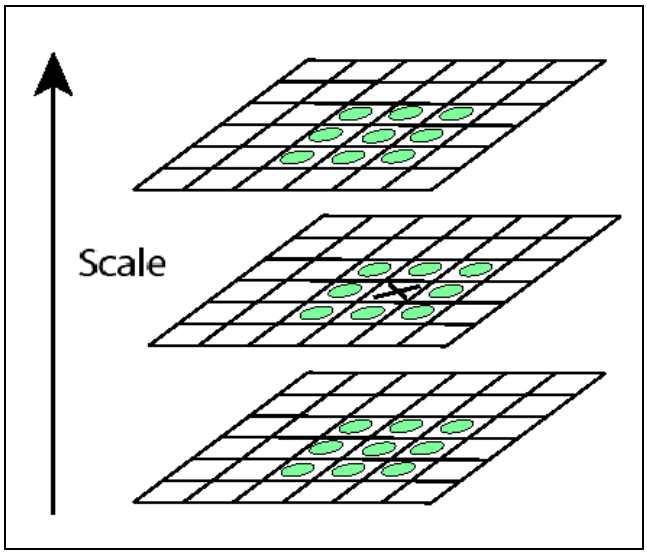


Figure 14 Adjacent image scales are subtracted to produce the difference of Gaussian images [3].

4.4.2 Keypoint localization:

After the detection of the keypoint candidates, the next step is the accurate localization of each keypoint candidate while removing the low contrasted and poorly localized ones.

A method is developed by Brown and Lowe et al [6] to fit a 3D quadratic function to the keypoint candidate to determine the interpolated location of the maximum. This method uses the Taylor expansion of scale-space function, $D(x, y, \sigma)$:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \qquad (4.13)$$

where D and its derivatives are evaluated at the same point and $x = (x, y, \sigma)^T$ is the offset from this point. The location of maximum/minimum point, $\hat{x}$, is computed by taking the derivative of this function with respect to x and setting the result to zero. Solving this equation, $\hat{x}$ is estimated as follows [3]:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2}\frac{\partial D}{\partial x} \tag{4.14}$$

If $\hat{x}$ is smaller than 0.5 in any dimension, the offset is added to its keypoint candidate to compute the interpolated estimate for the location of extremum.

To reject the keypoint candidates with low contrast $D(\hat{x})$ is calculated:

$$D(\hat{x}) = D + \frac{1}{2}\frac{\partial D^T}{\partial x}\hat{x} \tag{4.15}$$

If the value of $\left|D(\hat{x})\right|$ is less than 0.03, the corresponding keypoint candidate is rejected.

Besides rejecting the low contrasted keypoint candidates, Difference of Gaussian (DoG) function has strong responses along the edges, which creates unstable keypoints.

Edges create poor peaks in DoG function which have a large principal curvature across the edge but a small one in the perpendicular direction [3]. To find these principal curvatures Hessian matrix is used:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{4.16}$$

where D_{xx} and D_{yy} represents the secondary derivates in the x and y direction respectively.

Consider α as the eigenvalue with the largest magnitude and β be the smaller one of this Hessian matrix [3], then

$$\begin{aligned} Tr\,(H) &= D_{xx} + D_{yy} = \alpha + \beta \\ Det\,(H) &= D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \end{aligned} \tag{4.17}$$

Let $\alpha = r\beta$ and

$$R = \frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \tag{4.18}$$

It is suggested in [3] that the keypoints that have the ratio R greater than $\frac{(r+1)^2}{r}$ with $r = 10$ should be eliminated.

4.4.3 Orientation assignment

After the keypoints are detected and localized, the next step is to assign orientation to each keypoint based on local image gradient directions. In the keypoint descriptor identification step, the keypoint descriptors are related with this orientation values and therefore achieve rotation invariant property [3].

The scale of the keypoint is used to select the Gaussian smoothed image, L , so that scale-invariant property is achieved. The gradient magnitude, $m(x, y)$ and the orientation, $\theta(x, y)$, for image $L(x, y, \sigma)$ are computed as:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$
$$\theta(x,y) = \tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \tag{4.18}$$

An orientation histogram with 36 bins where each bin represents 10 degrees, covering 360 degree is computed. Each sample in the neighboring window added to a histogram bin is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a variance σ that is 1.5 times of the scale of the keypoint [3].

Peaks in the orientation histogram represent the dominant direction of local gradients. The highest peak and any other peak that is 80% of the highest peak is used to create another keypoint with that orientation. This approach creates multiple keypoints with same location and scale but different orientations. [3]

Multiple keypoints with different orientations provide a signification contribution to the matching stability.

4.4.4 Keypoint descriptor:

Once the location, scale and orientation of the keypoints are detected, the next step is to compute the descriptors of these keypoints that are highly distinctive and invariant to illumination and 3D viewpoint.

This step is pretty similar to the Orientation Assignment step. The feature descriptor is computed as a set of orientation histograms on (4 x 4) pixel neighborhoods. The orientation histograms are relative to the keypoint orientation and the orientation data comes from the Gaussian image closest in scale to the keypoint's scale. Just like before, the contribution of each pixel is weighted by the gradient magnitude, and by a Gaussian with σ 1.5 times the scale of the keypoint. Histograms contain 8 bins each, and each descriptor contains a 4x4 array of 16 histograms around the keypoint. This leads to a SIFT feature vector with (4 x 4 x 8 = 128 elements). This vector is normalized to enhance invariance to changes in illumination.

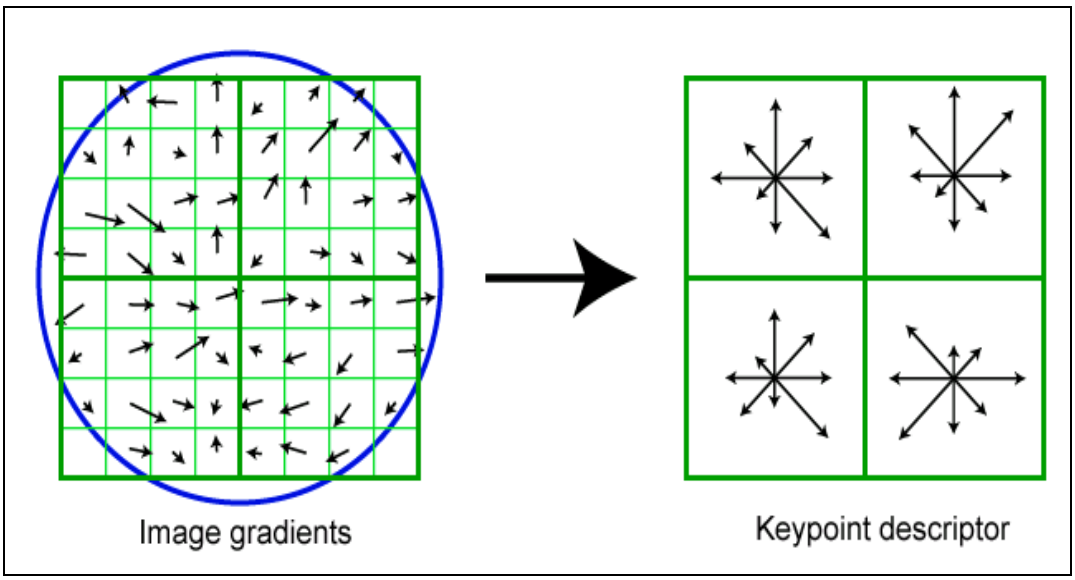


Figure 15 The computation of 2x2 descriptor. The computed gradient magnitude and orientation is weighted by a Gaussian window indicated by the circle as shown on the left. On the right side, keypoint descriptor is shown. It allows for significant

shift in gradient positions by creating orientation histograms over 4x4 sample regions [3].

A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left part of figure 3.5. These are weighted by a Gaussian window with a σ that is 1.5 times of the scale of the keypoint, indicated by the circle. Gaussian windowing is performed in orientation assignment step. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 sub-regions, as shown on the right of figure 3.5 with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, 4x4 descriptors computed from a 16x16 sample array is generally used [3]. A histogram with 8 bins where each bin represents an orientation is formed, and each descriptor contains a 4x4 array of 16 histograms around the keypoint. This leads to a SIFT feature vector with (4 x 4 x 8 = 128 elements). Then this vector is normalized to remove illumination effect.

4.5 Feature Matching with SIFT Keypoint Descriptors

Feature matching with keypoint descriptors is performed by comparing the descriptor of the keypoint with the descriptors of match candidates on the other image and this is done by Euclidean-distance based nearest neighbor approach.

Euclidean distance between the descriptor of selected keypoint and the descriptors of match candidates are computed. If the ratio of the nearest neighbor distance to the second nearest neighbor distance is greater than d then the corresponding match is rejected. The value of d is implementation dependent. Its valid range is between 0.0 and 1.0. The smaller the value of d , the more matches are found, also with more outliers. The larger the value of d , the lesser matches are found, but with more robust matches. The value of d is 0.8 in [3]. In this thesis d is taken as 0.97.

Figure 16 shows the 2 frames from Coral sequence with detected features via SIFT method and the last image in figure 16 shows the translation of matched features with a blue line.

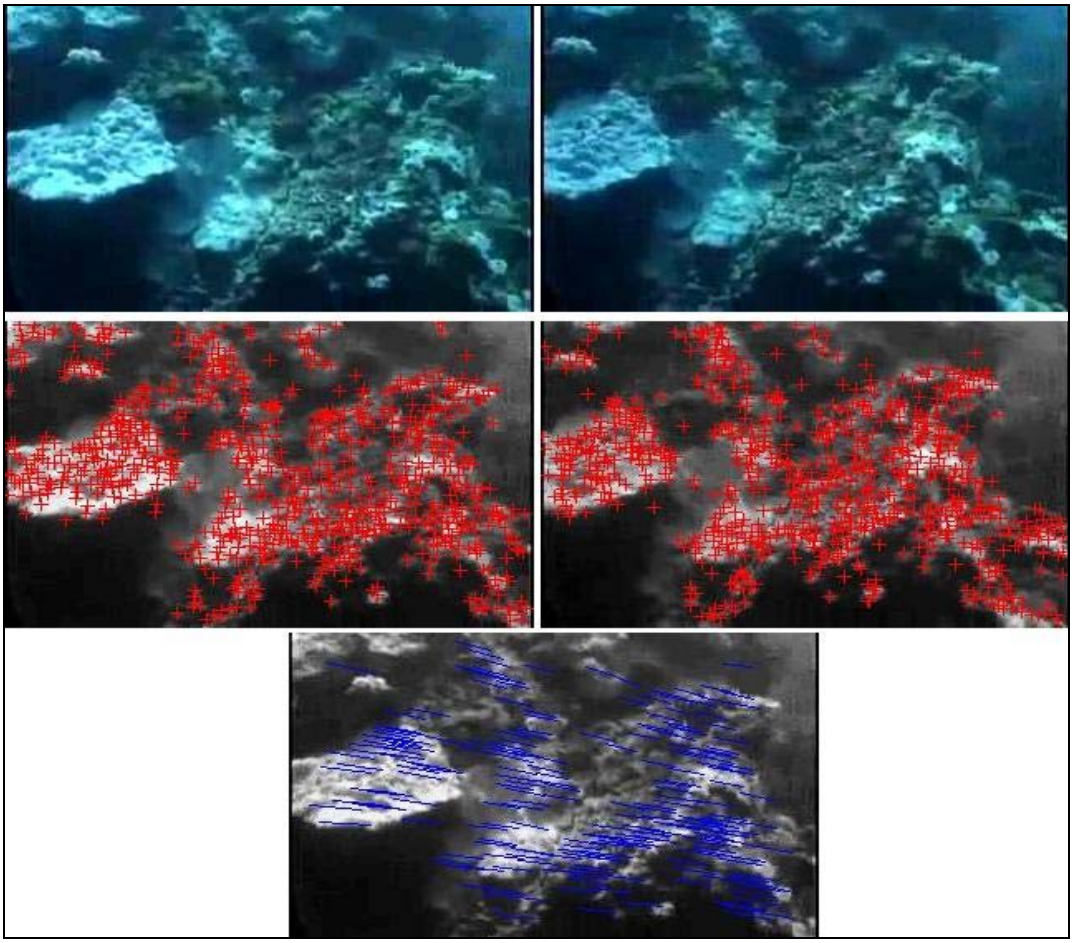


Figure 16 The upper left and upper right images are the two frames from coral sequence. The red crosses show the feature points with SIFT method in the middle left and middle right images. The last image shows the matches between the feature points of the image above using the Key Descriptor method.

4.6 Results and Conclusion

In Figure 11, two frames from coral sequence are shown with the features detected by Harris Corner Detector and the matching result of normalized cross correlation is also shown. In Figure 15 the features detected by SIFT and the matching result of keypoint descriptor is seen. For comparison purposes, the

number of detected features and matched pairs are listed in Table 3 for Harris corner detector and SIFT method.

Table 3 The Comparison of SIFT and Harris Corner Detector.

	Feature Detection & Matching Method	
	SIFT & Keypoint Descriptor	**Harris Corner Detector & Normalized Cross Correlation**
Features Detected in left image	926	359
Features Detected in right image	866	323
Corresponding Pairs	241	178

In Coral sequence, SIFT method gives a better performance than Harris Corner Detector and Normalized Cross Correlation. Nearly 3 times more features are detected with SIFT compared to Harris. Keypoint Descriptor used in SIFT gives a better performance than normalized cross correlation method. When the last images of Figure 12 and Figure 16 are compared, normalized cross correlation gives more false matches than Keypoint Descriptor.

In order to compare the computation time of these algorithms, processing times are presented in Table 4.

Table 4 The execution times of feature detection and matching algorithms

Algorithm	**Process Time (in seconds)**

SIFT	1.062
Keypoint Detector	0.437
Harris Corner Detector	0.172
Normalized Cross Correlation	0.281

From these results, it is seen that SIFT is more suitable for underwater applications. Although Harris Corner Detector and Normalized Cross Correlation performs faster than SIFT and Keypoint Descriptor, these algorithms suffers from the effects of water, blurring and scattering, which decreases the edge detection performance and accuracy.

CHAPTER 5

FUNDAMENTAL MATRIX ESTIMATION

5.1 Introduction

Two images of a scene are represented by epipolar geometry which is independent of scene structure and depends on camera's intrinsic and extrinsic parameters. The geometric relation of a stereo image pair can be related with a 3x3 singular matrix. If internal parameters of the camera are known, the epipolar geometry is defined with an essential matrix [6]. Essential matrix consists of the extrinsic parameters of the stereo system. If the internal parameters of the camera are not known, the epipolar geometry is defined with a fundamental matrix [7]. Fundamental matrix contains both internal parameters and relative poses of the cameras. Since a video sequence is used in this thesis, two consecutive frames of the video are considered as two images, taken from the same camera, which means that they have the same internal parameters, with an undetermined rotation and translation.

By the estimation of the fundamental matrix, the epipolar geometry of the stereo image pair is found. It is known that corresponding points between the two images are enough to estimate the fundamental matrix. Many algorithms are developed to estimate fundamental matrix using its properties. One of the most popular algorithms is *8-Point Algorithm* that Longuet-Higgins introduced to estimate the essential matrix for calibrated cameras [8]. It is linear, fast and easy to implement but very sensitive to noise. Hartley introduced a method based on 8-Point Algorithm with a slight modification by normalizing the input data before constructing the equations which increased the performance significantly [7]. Hartley's method is called normalized 8-point algorithm. In this thesis, normalized

8-point algorithm is used for estimating fundamental matrix because of its performance and implementation ease.

5.2 8-Point Algorithm

Once at least 8 corresponding matches are known, fundamental matrix can be estimated up to a scale factor. The epipolar relation between the two matching points and the fundamental matrix is:

$$m_i'^T \; F \; m_i = 0 \tag{5.1}$$

where $m_i = \left[u_i, v_i, 1\right]^T$, $m_i' = \left[u_i', v_i', 1\right]^T$ and F is the fundamental matrix. If this matrix multiplication is extended for two points, the following linear equation is formed:

$$u_i'u_i f_{11} + u_i'v_i f_{12} + u_i' f_{13} + v_i'u_i f_{21} + v_i'v_i f_{22} + v_i' f_{23} + u_i f_{31} + v_i f_{32} + f_{33} = 0 \tag{5.2}$$

The linear equation for *n* corresponding points is:

$$Af = \begin{bmatrix} u_1'u_1 & u_1'v_1 & u_1' & v_1'u_1 & v_1'v_1 & v_1' & u_1 & v_1 & 1 \\ : & : & : & : & : & : & : & : & : \\ : & : & : & : & : & : & : & : & : \\ u_n'u_n & u_n'v_n & u_n' & v_n'u_n & v_n'v_n & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0 \tag{5.3}$$

The eigenvector corresponding to the smallest eigenvalue of matrix **A** is the estimated fundamental matrix. The fundamental matrix is a rank-2 matrix. Because of the errors while determining the corresponding points and the noise on point coordinates, estimated fundamental matrix can not meet rank-2 constraint and epipolar lines do not meet in a single point and the algorithms

which depend on the fundamental matrix fails. For these reasons, the estimated fundamental matrix F is corrected by a singular matrix F' which minimizes the Frobenius norm $\|F - F'\|$. By Singular Value Decomposition method, the eigenvalues of estimated fundamental matrix is computed and the smallest one is set to 0. By this method rank-2 constraint for fundamental matrix is provided.

$$F = UDV^T \tag{5.4}$$

where $D = diag(r,s,t)$ and $r \geq s \geq t$. Then $t = 0$ and

$$F' = Udiag(r,s,0)V^T \tag{5.5}$$

where F' is the corrected rank-2 fundamental matrix.

5.3 Normalized 8-Point Algorithm

Normalized 8-Point Algorithm is the modified version of *8-Point Algorithm* with normalized corresponding points. The biggest advantage of *8-Point Algorithm* is being linear and easy to implement, but it is very sensitive to noise. It is shown that using *8-Point Algorithm* with normalized corresponding points significantly increases the performance of the algorithm [7].

The *normalized 8-point algorithm* is summarized as follows:

- First, the center of corresponding points are calculated as follows:

$$m_i = [u_i, v_i, 1]^T \text{ and } m'_i = [u'_i, v'_i, 1]^T$$

$$u_{av} = \frac{1}{n}\sum_{i=1}^{n} u_i \quad v_{av} = \frac{1}{n}\sum_{i=1}^{n} v_i$$

$$u'_{av} = \frac{1}{n}\sum_{i=1}^{n} u'_i \quad v'_{av} = \frac{1}{n}\sum_{i=1}^{n} v'_i$$

where m_i and m_i' are the i-th corresponding pair in left and right image respectively. u_i, v_i are the X and Y coordinates of the point m_i; u_i' and v_i' are the X and Y coordinates of the point m_i'.

- The center of the corresponding point is transformed to the center of the reference and the corresponding points are scaled so that the root-mean-square (RMS) distance of the points to the origin is $\sqrt{2}$.

$$\hat{u}_i = \frac{\sqrt{2}\,(u_i - u_{av})}{s_1} \text{ and } \hat{v}_i = \frac{\sqrt{2}\,(v_i - v_{av})}{s_1}$$

$$\hat{u}_i' = \frac{\sqrt{2}\,(u_i' - u_{av}')}{s_2} \text{ and } \hat{v}_i' = \frac{\sqrt{2}\,(v_i' - v_{av}')}{s_2}$$

where

$$s_1 = \frac{1}{n}\sum_{i=1}^{n} \sqrt{\left((u_i - u_{av})^2 + (v_i - v_{av})^2\right)}$$

$$s_2 = \frac{1}{n}\sum_{i=1}^{n} \sqrt{\left((u_i' - u_{av}')^2 + (v_i' - v_{av}')^2\right)}$$

- The resultant transformation matrices are:

$$T_1 = \begin{bmatrix} \frac{\sqrt{2}}{s_1} & 0 & \frac{-u_{av}}{s_1} \\ 0 & \frac{\sqrt{2}}{s_1} & \frac{-v_{av}}{s_1} \\ 0 & 0 & 1 \end{bmatrix} \text{ and } T_2 = \begin{bmatrix} \frac{\sqrt{2}}{s_2} & 0 & \frac{-u_{av}'}{s_2} \\ 0 & \frac{\sqrt{2}}{s_2} & \frac{-v_{av}'}{s_2} \\ 0 & 0 & 1 \end{bmatrix}$$

- Normalized corresponding points are:

$$\hat{m}_i = T_1 m_i \text{ and } \hat{m}'_i = T_2 m'_i$$

- The corrected fundamental matrix $\hat{F}$ is computed with the normalized corresponding point set $\hat{m}_i$ and $\hat{m}'_i$ by 8-point algorithm and inverse translation is applied to get the fundamental matrix F :

$$F = T_2^T \hat{F} T_1$$

5.4 RANSAC

The input of the fundamental matrix estimation algorithm is the corresponding points and the noise in the image causes false matches, called outliers, which is mentioned in Chapter 5.2 and Chapter 5.5. Since an input set with outliers does not provide correct results for the fundamental matrix estimation and is the source of error in the estimation process, the outliers, which does not fit the fundamental matrix model, should be rejected. RANSAC method is used for this outlier rejection process.

The fundamental matrix estimation with RANSAC is summarized as follows:

Repeat for N times

- Select a random sample of 8 corresponding points and compute the fundamental matrix F with the *normalized 8-point algorithm*

- Calculate the Sampson error e_{samp} for each putative match for the estimated fundamental matrix.

$$e_{samp,i} = \frac{(m_i'^T F m_i)^2}{(Fm_i)_1^2 + (Fm_i)_2^2 + (F^T m'_i)_1^2 + (F^T m'_i)_2^2}$$

- If $e_{samp\ ,i}$ is below the threshold, then that match is considered as inlier, otherwise it is considered as outlier. The threshold value is the distance of

the point to the epipolar line, which is 0.02 for normalized corresponding points.

- Choose F with the largest number of inliers, and reject the outliers.
- Refresh the number of iterations N

$$N = \frac{\log(1-p)}{\log(1-(1-e)^s)}$$

where p is the probability that at least one of the random samples of points is free from outliers, usually it is as 0.99, e is the probability that any selected point is an outlier. s is the size of the sample, 8 for *normalized 8-point algorithm*.

5.5 Results and Conclusion

The accuracy and constraint satisfaction in fundamental matrix estimation is crucial. The performance of fundamental matrix estimation determines the performance of the rest of the reconstruction process.

Figure 17 shows the epipolar lines in the two images of coral sequence via the fundamental matrix computed by normalized 8-point algorithm and RANSAC method.

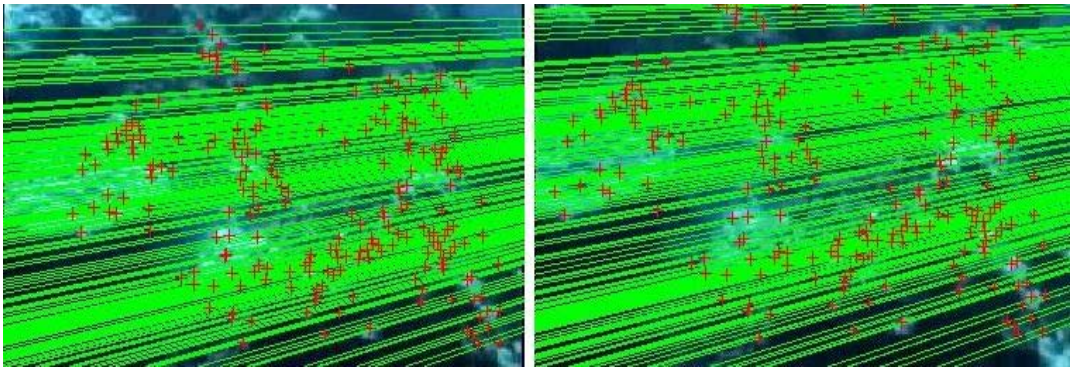


Figure 17 The corresponding pairs and the epipolar lines in coral sequence.

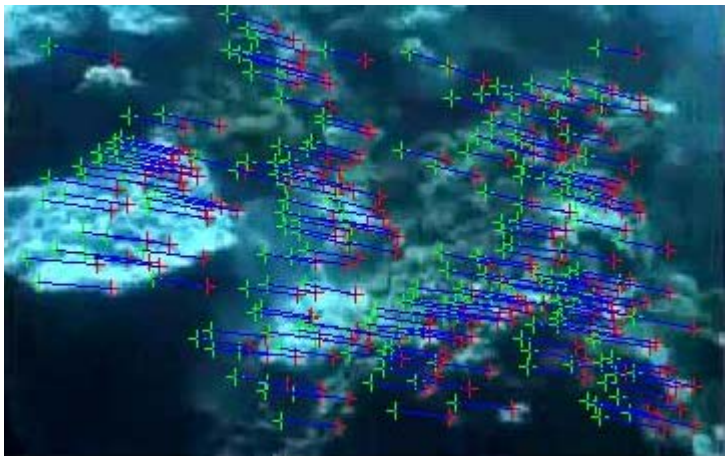


Figure 18 The inliers computed by RANSAC method. The red cross show the position of the corresponding point in the left image. The green cross show the position of the corresponding point in the right image. The blue line represents the route of the point between the two frames.

In coral sequence, 231 inliers are found among 241 corresponding pairs with a Sampson error of 0.0217.

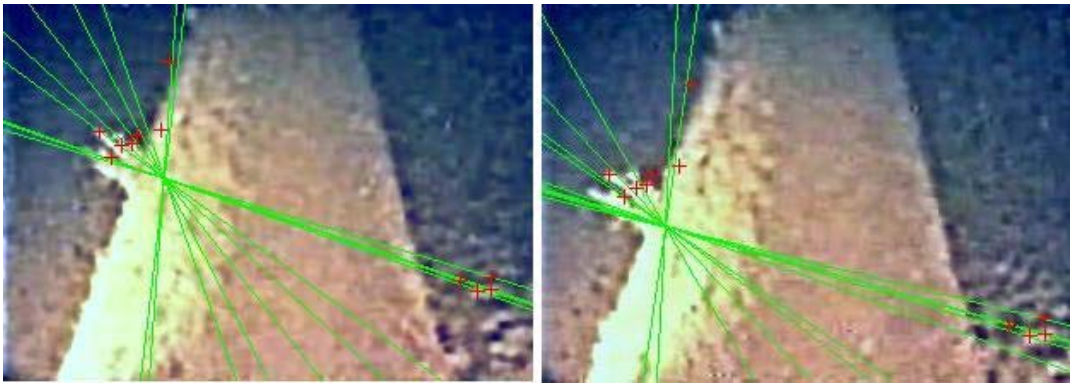


Figure 19 The corresponding pairs and the epipolar lines in pipe sequence.

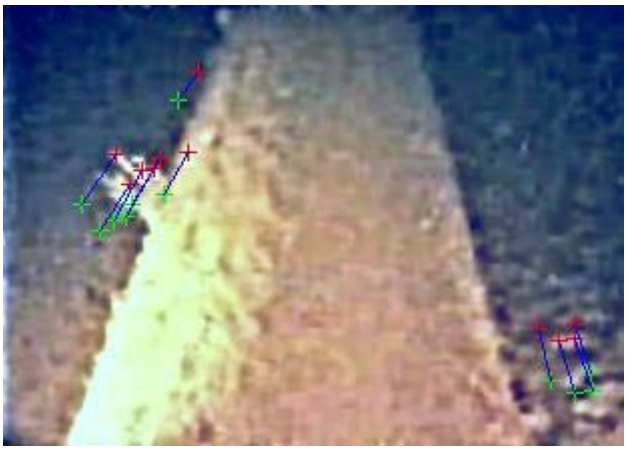


Figure 20 The inliers in the pipe sequences. Unlike in coral sequence, camera makes a translational motion in this sequence.

In pipe sequence, 14 inliers are found among 14 corresponding pairs with a Sampson error of 0.9788.

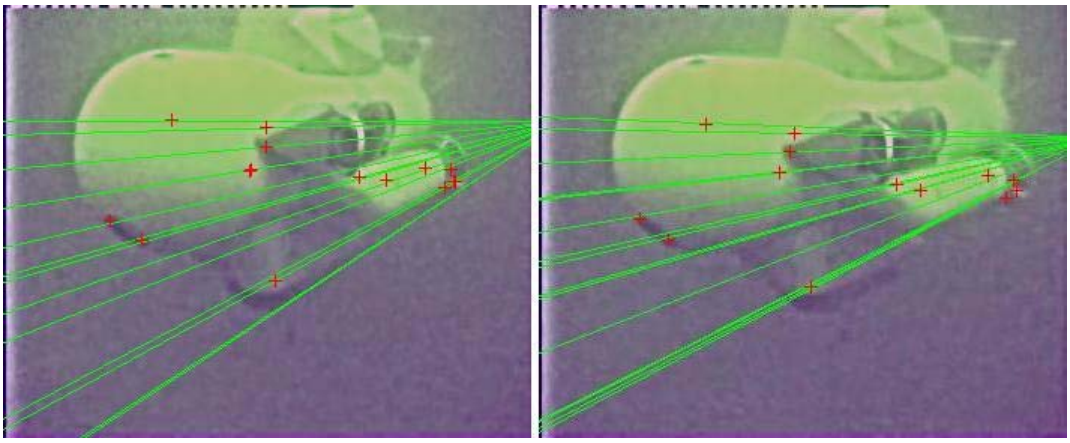


Figure 21 The corresponding pairs and the epipolar lines in boat sequence.

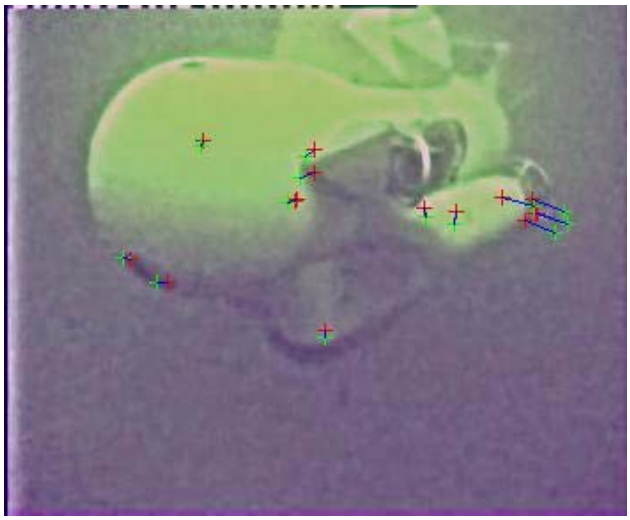


Figure 22 The inliers in the boat sequence.

In pipe sequence, 19 inliers are found among 21 corresponding pairs with a Sampson error of 0.6374.

Table 5 The percentage of inliers on corresponding points and the computed Sampson errors for different video sequences.

Sequences	# of Inliers	Inliers / Corresponding Points	Sampson Error
Coral	231	231/241 ~ %96	0.0217
Pipe	14	14/14 ~ %100	0.9788
Boat	19	19/21 ~ %90	0.6374

CHAPTER 6

AUTO CALIBRATION

6.1 Introduction

Auto-calibration or *Self-calibration* is the process of determining the intrinsic parameters directly from multiple uncalibrated images. Once auto-calibration is done, the extrinsic parameters, rotation and translation matrices, can be computed and it is possible to compute the metric reconstruction of the scene. Auto-calibration provides the calibration of the camera directly from an image sequence despite the unknown motion, instead of calibrating the camera with a special calibration object [6].

6.2 Algebraic Framework

Consider that a projective reconstruction is performed with a camera projection matrix P_i and 3D point coordinate M_j is computed. This 3D point is projected to the 2D point via $m^i_j = P_i M_j$, where i represents the number of views and j represents the number of corresponding points. A 4x4 homography matrix H is used to upgrade the projection reconstruction to a metric one. The projection matrix of the camera and the 3D point coordinate is transformed as following, but the coordinates of the 2D projected point does not change:

$$m^i_j = (P_i H)(H^{-1} M_j) \tag{6.1}$$

$$P_{Mi} = P_i H \quad M_{Mj} = H^{-1} M_j \tag{6.2}$$

The goal is to determine the homography matrix H , where $H = \begin{bmatrix} A & t \\ v^T & k \end{bmatrix}$. Since the left camera coordinate system is assumed as the world coordinate system, the projection matrix of the first camera in the projection reconstruction is $P_1 = [I \;\; 0]$. In metric reconstruction once the calibration matrix is estimated, the projection matrix is transformed to $P_{M1} = K_1[I \;\; 0]$. The transformation from projective to metric reconstruction is $P_{M1} = P_1 H \Rightarrow K_1[I \;\; 0] = [I \;\; 0]H$. This relation shows that $A = K_1$ and $t = 0$. Since H is non-singular, k must be non-zero and it is set to 1 to fix the scale of the reconstruction. The vector v with K_1 determines the plane at infinity in projective reconstruction satisfying the following relation:

$$\pi_\infty = H^{-1}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} K_1^{-T} & -K_1^{-T}v \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -K_1^{-T}v \\ 1 \end{bmatrix} \quad (6.3)$$

The plane at infinity can be represented as $\pi_\infty = [p \;\; 1]^T$ where $p = -K_1^{-T}v$. The homography matrix can be also be represented as:

$$H = \begin{bmatrix} K_1 & t \\ -p^T K_1 & k \end{bmatrix} \quad (6.4)$$

If the plane at infinity in the projective frame and the calibration matrix of the first camera are known, the homography H that upgrades the projective reconstruction to the metric one can be computed. To do this, it is sufficient to specify the 8 parameters, 3 for p and 5 for K_1 .

Let the projective matrices of the other cameras be $P_i = [A_i \;\; a_i]$. From Equation 6.2,

$$K_i R_i = (A_i - a_i p^T) K_1 \quad \text{for i=2…m (number of cameras)} \qquad (6.5)$$

which may be rearranged as

$$R_i = K_i^{-1}(A_i - a_i p^T) K_1 \qquad (6.6)$$

Remembering the rotation matrix is an orthogonal matrix $R_i R_i^T = I$, the rotation matrix can be eliminated by multiplying both sides with R_i^T and the result is:

$$K_i K_i^T = (A_i - a_i p^T) K_1 K_1^T (A_i - a_i p^T)^T \qquad (6.7)$$

The dual image of the absolute conic is $\omega_i^* = K_i K_i^T$ and by making this substitution the basic auto-calibration equations are derived [6]:

$$\omega_i^* = (A_i - a_i p^T) \omega_1^* (A_i - a_i p^T)^T \qquad (6.8)$$

$$\omega_i = (A_i - a_i p^T)^{-T} \omega_1 (A_i - a_i p^T)^{-1} \qquad (6.9)$$

All self-calibration methods are the variations of solution to Equation (6.8) and (6.9). The first step is to compute ω_i^* or ω_i , then the calibration matrix is calculated by Cholesky Decomposition of $\omega_i^* = K_i K_i^T$.

A counting argument can be developed to determine the number of view required to solve the 8 unknown parameters. Each image except the first one imposes 5 constraints. Since these constraints are independent for each view, a solution is determined providing $5(m-1) \geq 8$, where m is the number of views. At least 3 views are required to solve the auto-calibration problem.

6.3 Auto-calibration by Dual Absolute Quadric

The absolute dual quadric, Ω_{∞}^{*}, is a dual quadric represented by a 4x4 homogenous matrix of rank 3 [6]. In Euclidean coordinate system, the absolute dual quadric is defined as $\Omega_{\infty}^{*} = diag\ (1,1,1,0)$. The importance of the absolute dual quadric is that it is invariant to transformations. According to this property a transformation that transforms the dual absolute conic to its form in Euclidean frame, $\Omega_{\infty}^{*} \rightarrow diag\ (1,1,1,0)$, will transform the projective reconstruction to the metric one.

The projection of the dual absolute quadric in the image is defined as:

$$w_i^{*} \approx P_i \Omega_{\infty}^{*} P_i^{T} \tag{6.10}$$

Once the dual absolute quadric is computed, using Equation (6.10) the dual image of the absolute conic can be calculated and with Cholesky Decomposition the calibration matrix can be computed via $\omega_i^{*} = K_i K_i^{T}$ .The algorithm examined below is known as *linear auto-calibration*.

The first step is to normalize the projection matrix.

$$P_N = K_N^{-1} P \quad K_N = \begin{bmatrix} w+h & 0 & w/2 \\ & w+h & h/2 \\ & & 1 \end{bmatrix} \tag{6.11}$$

where w and h are the width and height of the image, respectively. After the normalization the focal length is in the order of 1, the principal point is close to origin. For practical purposes, the skew is assumed as 0 and the aspect ratio as 1. Considering the standard deviation on the intrinsic parameters, $f \approx 1 \pm 3, u_0 \approx v_0 \approx \pm 0.1$, the following relation can be composed: [1]

$$\omega^* \approx KK^T = \begin{bmatrix} f^2+u_0^2 & u_0v_0 & u_0 \\ u_0v_0 & f^2+v_0^2 & v_0 \\ u_0 & v_0 & 1 \end{bmatrix} = \begin{bmatrix} 1\pm 9 & \pm 0.01 & \pm 0.1 \\ \pm 0.01 & 1\pm 9 & \pm 0.1 \\ \pm 0.1 & \pm 0.1 & 1 \end{bmatrix} \quad (6.12)$$

and $\omega_{22}^* / \omega_{11}^* \approx 1 \pm 0.2$. Using the Equation (6.9) the uncertainty can be handled by weighting the equations as [1]:

$$\begin{aligned}
&\frac{1}{9v}(P_1\Omega_\infty^* P_1^T - P_3\Omega_\infty^* P_3^T) = 0 \\
&\frac{1}{9v}(P_2\Omega_\infty^* P_2^T - P_3\Omega_\infty^* P_3^T) = 0 \\
&\frac{1}{0.2v}(P_1\Omega_\infty^* P_1^T - P_2\Omega_\infty^* P_2^T) = 0 \\
&\frac{1}{0.1v}(P_1\Omega_\infty^* P_2^T) = 0 \\
&\frac{1}{0.1v}(P_1\Omega_\infty^* P_3^T) = 0 \\
&\frac{1}{0.01v}(P_2\Omega_\infty^* P_3^T) = 0
\end{aligned} \quad (6.13)$$

where P_i is the i-th row of P and v is a scale factor, initially set to 1 but updated with $P_3\tilde{\Omega}_\infty^* P_3^T$ where $\tilde{\Omega}_\infty^*$ is the result of previous iteration [1]. Since Ω_∞^* is a symmetric 4x4 matrix, it is defined with 10 coefficients. Once the absolute dual quadric is computed, the dual image of absolute quadric can be calculated via Equation (6.10) and the calibration matrix can be calculated by the Cholesky Decomposition of the dual image of absolute quadric.

6.4 Auto-calibration by Kruppa Equations

The absolute conic is a special conic lying at the plane at infinity and having the property that its projection depends on the intrinsic parameters of the camera

[21]. This property is expressed mathematically by Kruppa equations. If the intrinsic parameters of the camera do not change between the frames, the image of the absolute conic ω will be same.

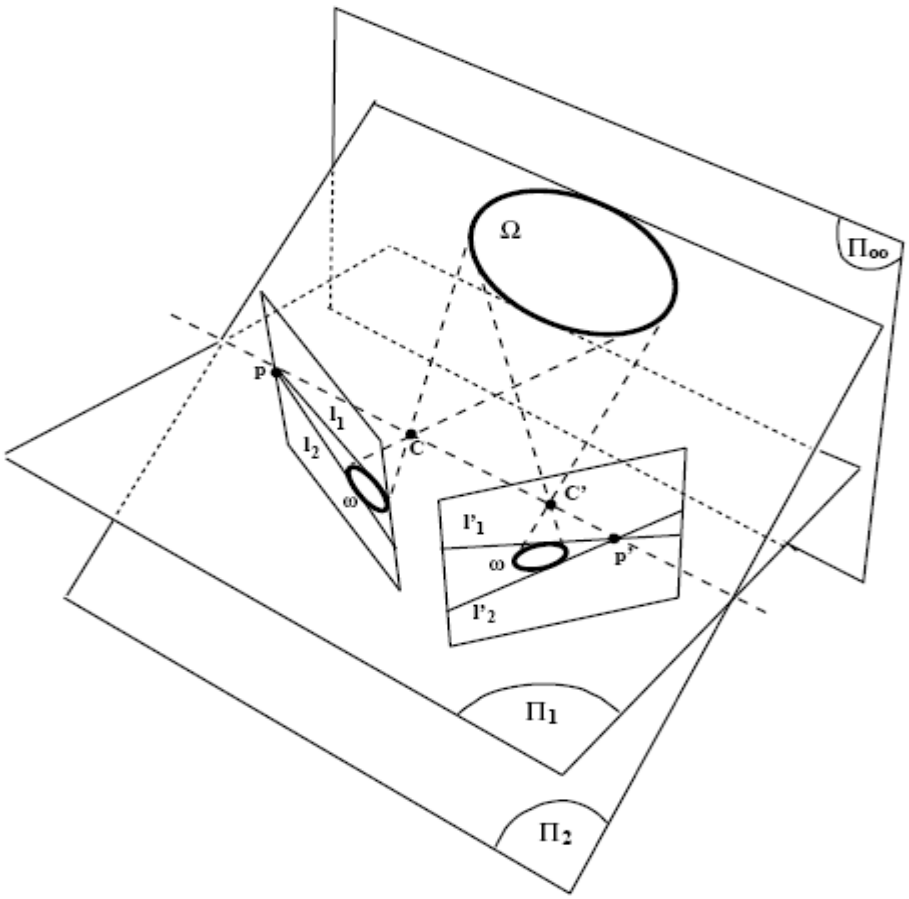


Figure 23 The Absolute Conic

The derivation of classical Kruppa equation is as follows:

Given that $F^T e' = 0$, the epipole of the right camera e' must satisfy the following equation:

$$K^{-T} R^T [t]_x^T K'^{-1} e' = 0 \tag{6.14}$$

Remembering that $[t]_x^T t = 0$, the following solution for e' is obtained:

$$e' = \lambda K' t \tag{6.15}$$

where λ is a non-zero scalar. The translation can be obtained as:

$$t = \frac{1}{\lambda} K'^{-1} e' \tag{6.16}$$

Equation (6.16) leads to the following equation for $[t]_x$:

$$[t]_x = \frac{1}{\lambda} \det(K'^{-1}) K'^{T} [e']_x K' \tag{6.17}$$

Substitution of Equation (6.17) into Equation (6.18) yields the Kruppa equations in the matrix form as shown in Equation (6.19) [21].

$$FKK^{T}F^{T} = K^{-T}[t]_x ([t]_x)^T K'^{T} \tag{6.18}$$

$$FKK^{T}F^{T} = \gamma [e']_x K' K'^{T} ([e']_x)^T \tag{6.19}$$

where γ is non-zero scalar. Since $FKK^{T}F^{T}$ is a symmetric matrix, by eliminating the γ , the Kruppa equations are formed:

$$\begin{aligned}
&\frac{(FKK^TF^T)_{11}}{([e']_x K'K'^T([e]_x)^T)_{11}} = \frac{(FKK^TF^T)_{12}}{([e']_x K'K'^T([e]_x)^T)_{12}} = \\
&\frac{(FKK^TF^T)_{22}}{([e']_x K'K'^T([e]_x)^T)_{22}} = \frac{(FKK^TF^T)_{13}}{([e']_x K'K'^T([e]_x)^T)_{13}} = \\
&\frac{(FKK^TF^T)_{23}}{([e']_x K'K'^T([e]_x)^T)_{23}} = \frac{(FKK^TF^T)_{33}}{([e']_x K'K'^T([e]_x)^T)_{33}}
\end{aligned} \tag{6.21}$$

The Equations (6.20) are linearly dependent since

$$(FKK^{T}F^{T} - \gamma [e']_x K' K'^{T} ([e']_x)^T) e' = 0 \tag{6.22}$$

There are two independent equations among the set of six equations. A special parameterization of epipolar geometry is used in order to choose which two equations are selected or by randomly selecting one equation for estimating the scale factor and then substituting the result into two others that are arbitrarily chosen among the remaining five ones [21].

Simplified Kruppa equations derive fewer equations than the classical Kruppa equations and there is no need to compute e' which suffers from the presence of noise and degenerate motion. Because of these reasons, Simplified Kruppa equations are used in this thesis.

6.5 Simplified Kruppa Equations

In [22] Hartley derived simplified Kruppa equations using Singular Value Decomposition of fundamental matrix. The SVD of F is:

$$F = UDV^T \tag{6.23}$$

Remembering that F is rank 2 matrix that the diagonal matrix D has the form:

$$D = \begin{bmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6.24}$$

where r and s are the eigenvalues of the matrix FF^T, U and V are the orthogonal matrices. The relation between the fundamental matrix F and the epipole e' can be rewritten as:

$$F^T e' = VD^T U^T e' = 0 \tag{6.25}$$

Since D is a diagonal matrix with a last element zero, the direct solution of e' is:

$$e' = Uo \tag{6.26}$$

where $o = [0 \ 0 \ 1]^T$. Hence the skew-symmetric matrix of e' is:

$$[e']_x = UOU^T \tag{6.27}$$

where $O = [o]_x$.

Substituting Equation (6.26) with Equation (6.19) new expression of Kruppa equation is obtained:

$$FKK^T F^T = \gamma UoU^T K'K'^T UO^T U^T \tag{6.28}$$

It is assumed that the calibration matrix of the cameras are the same, $K = K'$. Since U is an orthogonal matrix, left and right multiplication of the Equation (6.28) by U^T and U respectively, gives the following simple expression of Kruppa equation:

$$DV^T AVD^T = \gamma OU^T AUO^T \tag{6.29}$$

where $A = KK^T$ .Because of the simple forms of D and O Equation (6.29) can be written as:

$$\begin{aligned} DV^T AVD^T &= \begin{bmatrix} r^2 v_1^T A v_1 & rsv_1^T A v_2 & 0 \\ rsv_2^T A v_1 & s^2 v_2^T A v_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ OU^T AUO^T &= \begin{bmatrix} u_2^T A u_2 & -u_2^T A v_1 & 0 \\ -u_1^T A u_2 & u_1^T A u_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{6.30}$$

where u_1, u_2, u_3 are the column vectors of U and v_1, v_2, v_3 are the column vectors of V .The above expressions finally yield the following three linearly dependent equations [31]:

$$\frac{r^2 u_1^T A u_1}{v_2^T A v_2} = \frac{rs u_1^T A u_2}{-v_2^T A v_1} = \frac{s^2 u_2^T A u_2}{v_1^T A v_1} \tag{6.31}$$

Since two of these three equations are independent, by cross multiplication the following independent equations can be obtained:

$$\begin{aligned} P_1 &= (r^2 u_1^T A u_1)(-v_2^T A v_1) + (v_2^T A v_2)(rs u_1^T A u_2) \\ P_2 &= (rs u_1^T A u_2)(v_1^T A v_1) + (-v_2^T A v_1)(s^2 u_2^T A u_2) \end{aligned} \tag{6.32}$$

First by solving the above equations for matrix A and then by Cholesky Decomposition the calibration matrix K can be computed.

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_2 & a_4 & a_5 \\ a_3 & a_5 & 1 \end{bmatrix} \text{ and}$$

$$K = \begin{bmatrix} \sqrt{(a_1 - a_3^2 - \frac{(a_2 - a_3 a_5)^2}{a_4 - a_5^2})} & \frac{a_2 - a_3 a_5}{\sqrt{a_4 - a_5^2}} & a_3 \\ 0 & \sqrt{a_4 - a_5^2} & a_5 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.33}$$

The derived simplified Kruppa equations can be rewritten as in the form:

$$C(A) = \sum_{i=1}^{3} P_1^2(F_i, A) + P_2^2(F_i, A) \tag{6.34}$$

The non-linear least square equation, Equation (6.32), can be solved by Levenberg-Marquardt minimization algorithm.

6.6 Results and Conclusion

Two methods of auto-calibration are examined, auto-calibration by absolute dual conic and auto-calibration by simplified Kruppa equations.

First the auto-calibration algorithms are examined with “Ballet Sequence” images which provide ground truth data for the intrinsic and the extrinsic parameters of the cameras.

The examined auto-calibration algorithms are performed using Figure 24 and the output calibration matrix is compared with the ground truth calibration matrix.

Figure 24 Three chosen frames from Ballet Sequence, a scene captured by 8 cameras from predetermined positions and known intrinsic and extrinsic parameters.

Each camera has its own intrinsic parameters. Since they are very close to each other, the intrinsic parameters of the left most camera is assumed to be the system’s calibration matrix which is:

$$K_{gt} = \begin{bmatrix} 1908.25 & 0.335 & 560.336 \\ 0 & 1914.16 & 409.596 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.35)$$

Solving Kruppa equations with Levenberg-Marquardt requires the initial estimation of the calibration matrix. The focal length in X and Y-axis is assumed to be same and equal to the summation of width and height of the image. The principal point is assumed to be in the middle of the image. Based on these assumptions the initial calibration matrix is:

$$K_{init} = \begin{bmatrix} 1792 & 0 & 512 \\ 0 & 1792 & 384 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.36}$$

The changes of the calibration parameters and the error to be minimized are shown in the following figure:

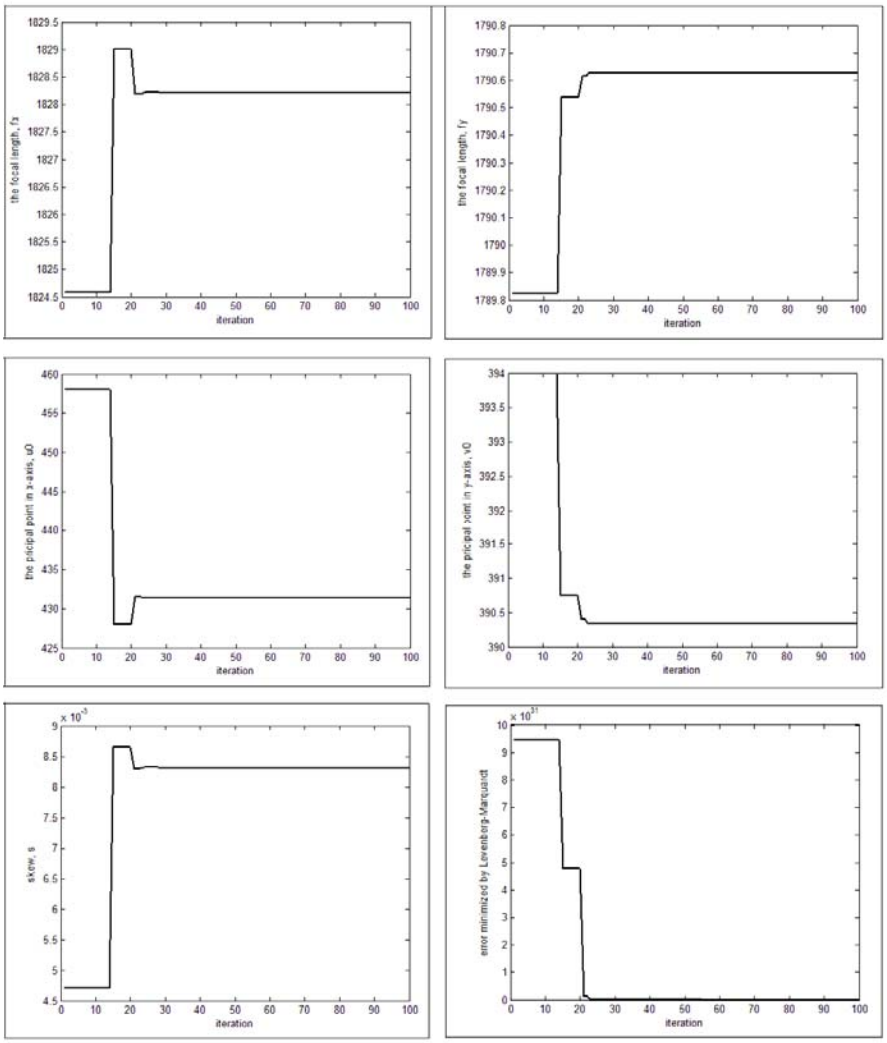


Figure 25 The estimated calibration parameters. The upper left graphic shows the focal length in x-axis, f_x, the upper right shows the focal length in y-axis, f_y. In the second row, the graph in the left shows the principal point in x-axis, u_0, the right graph shows the principal point in the y-axis, v_0. The lower right graph shows the skew s, and the right graph show the energy minimized by Levenberg-Marquardt.

As it is seen from the graphs above, the unknown parameters converge after 20-30 iterations and the final estimated calibration matrix is:

$$K_{est} = \begin{bmatrix} 1882 & 0.0083 & 431.4 \\ 0 & 1790.6 & 390.3 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.37)$$

Despite using the ground truth data, the estimated calibration matrix parameters are different from the ground truth ones. The difference in focal length in x-axis and the principal point in y-axis is in degrees of 10-20 pixels, which can be ignored. But the difference in focal length in y-axis and the principal point in x-axis is more than 100 pixels.

Another factor that affects the performance of solving the Kruppa equations with Levenberg-Marquart is the initial estimation. The more accurate initial estimation, the more successful results is computed. The simplified Kruppa algorithm is tested with different initial data sets.

Table 6 List of different data sets for the initial value of calibration parameters.

Initial Data Set No.	**f_x** **True = 1908**	**f_y** **True = 1914**	**u_0** **True=560**	**v_0** **True=409**	**s** **True=0.335**
1	1900	1900	500	400	0
2	1900	1900	0	0	0
3	1200	1200	500	400	0
4	2500	2500	500	400	0
5	500	800	50	50	0

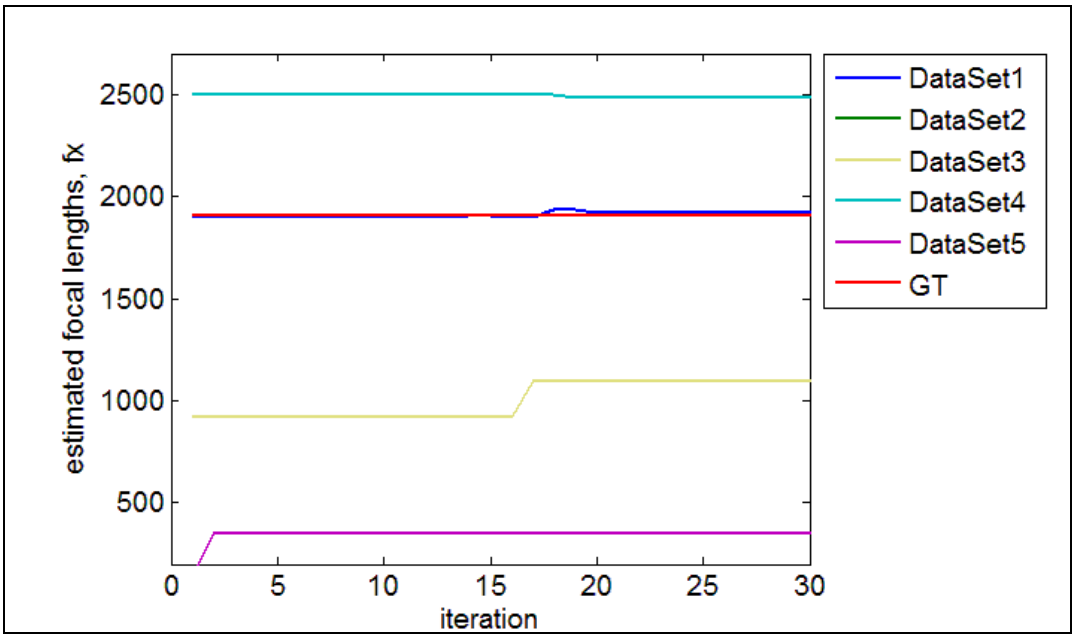


Figure 26 The effect of different initial data sets to the estimation of focal length in x-axis, f_x.

The graph above shows the effect of the data sets listed above on the estimation of focal length in x-axis, f_x. As it is seen from the graph, the accuracy of initial value determines the success of the estimation process. The estimations are given in Table 6. The most successful data set is Dataset2, with $f_x = 1907.5$, DataSet2 give the second best results with $f_x = 1925$. Although DataSet1 is closer to the true values than DataSet2, DataSet2 gives the best estimation in focal length. But when the other unknown calibration parameters are considered, DataSet1 gives the best estimation for all.

Table 7 The estimated calibration parameters with different initial data sets. Since DataSet1 is the closest data set to the true value, it gives the best estimations.

Initial Data Set No.	Est. f_x True = 1908	Est. f_y True = 1914	Est. u_0 True=560	Est. v_0 True=409	Est. s True=0.335
1	1925	1899	386	404	0.011

2	1908	1900	-35	32	0.00003
3	1094	1202	677	390	-0.04
4	2484.4	2500.5	572	396	-0.04
5	350	800	291	38	-0.0074

The auto-calibration with absolute dual conic is also tested with the ballet sequence. The required input data for the algorithm is the projection matrix of the camera and the width and height of the image. These data are provided by the ballet sequence. Equation (6.12) is computed iteratively until the parameter v converges and the absolute dual conic is computed with the projection matrix and parameter v . Finally the image of absolute dual conic is computed via Equation (6.9) and via Cholesky decomposition the calibration matrix is computed.

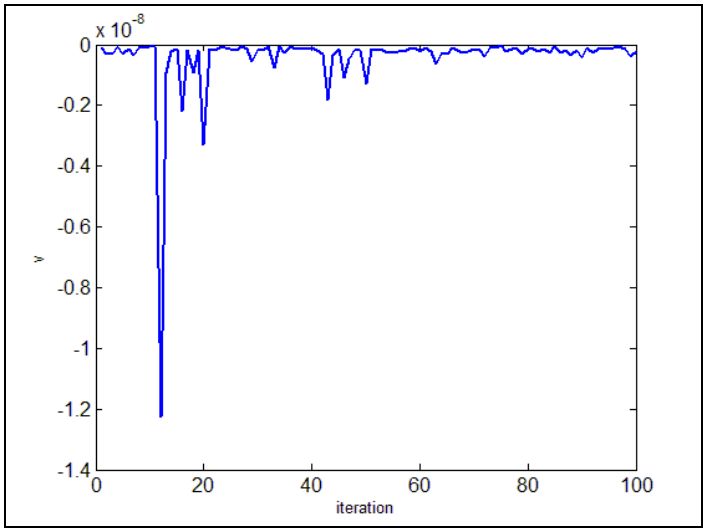


Figure 27 The parameter v.

It is expected that v to converge after a few iterations, but it does not converge after 100 iterations. Although the iteration number is increased, it continues to oscillate. The parameter v is set to the mean value of the last 20 iterations where

the oscillation decreases with respect to the beginning. After the parameter v is determined, the process described above is applied and the computed calibration matrix is:

$$K_{est} = \begin{bmatrix} 1792 & 0 & 512 \\ 0 & 1792 & 384 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.38}$$

The estimated calibration is exactly equal to the initial value of the calibration matrix in Kruppa equations, where the focal length is equal to the sum of width and height of the image and the principal point is in the middle of the image. Also it is equal to the calibration matrix K_N which is used to normalize the projection matrix in Equation (6.10). Although the estimated parameters satisfy the Equation (6.12), the algorithm fails to estimate the calibration matrix. For that reason the simplified Kruppa equations are used in this thesis.

The intrinsic parameters of the coral sequence are computed with the simplified Kruppa equations. Using the following initial estimation is:

$$K_{init} = \begin{bmatrix} 520 & 0 & 160 \\ 0 & 520 & 100 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.39}$$

the estimated calibration matrix is:

$$K_{est} = \begin{bmatrix} 495.75 & -0.0217 & 223.98 \\ 0 & 520.22 & 98.87 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.40}$$

CHAPTER 7

ESTIMATION OF CAMERA MOTION

7.1 Introduction

In the previous chapter, the intrinsic parameters of the camera are computed. However in order to compute 3D metric reconstruction of the scene, it is mandatory to compute the extrinsic parameters of the cameras, rotation and translation matrices.

Once the intrinsic parameters and the fundamental matrix are computed, rotation and translation matrix of the camera is estimated via *essential matrix*. Essential matrix is the special case of fundamental matrix for the case of normalized image coordinates [6]. Essential matrix is introduced before fundamental matrix. Essential matrix can be assumed as the special form of fundamental matrix, considering the case that the intrinsic parameters are known. The relation between the essential matrix and the fundamental matrix is:

$$E = K'^{T} F K \tag{7.1}$$

where K is the calibration matrix of the left camera and K' is the calibration matrix of the right camera. Since a video sequence is used in this thesis, the former frame is assumed as taken by the left camera and the latter frame taken by the right camera. It is assumed that the intrinsic parameters of the camera do not change between the two frames. Based on this assumption the intrinsic parameters of the left and the right camera are considered as equal.

$$K = K' \tag{7.2}$$

The normalized image coordinates is determined as $\hat{m} = K^{-1}m$ and the relation between the essential matrix and the normalized corresponding pairs is:

$$\hat{m}_i'^T E\ \hat{m}_i = 0 \tag{7.3}$$

If the left camera coordinate system is chosen as the world coordinate system, then the projection matrix of the left camera will be $P = K\begin{bmatrix} I & 0 \end{bmatrix}$ with zero translation and identity rotation matrix. The projection matrix of the right camera will be $P' = K\begin{bmatrix} R & t \end{bmatrix}$ with rotation R and translation t. Considering the left camera coordinate system as the world coordinate system, the rotation matrix R and the translation matrix t defines the transformation of the right camera coordinate system with respect to the left camera coordinate system and this is the case in this thesis.

With the rotation and the translation matrix the point $M = \begin{bmatrix} M_x & M_y & M_z \end{bmatrix}^T$ defined in the left camera coordinate system and the point $M' = \begin{bmatrix} M'_x & M'_y & M'_z \end{bmatrix}^T$ defined in the right camera coordinate system is related as:

$$M' = RM + t \tag{7.4}$$

The coordinates of the points on image plane is determined by normalizing the 3D coordinates of the points with the value of Z-axis, depth value.

$$\hat{m}_i = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{M_x}{M_z} \\ \frac{M_y}{M_z} \\ 1 \end{bmatrix}, \hat{m}'_i = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{M'_x}{M'_z} \\ \frac{M'_y}{M'_z} \\ 1 \end{bmatrix} \tag{7.5}$$

Equation 7.4 is revised:

$$M'_z \hat{m}' = M_z R \hat{m} + t \tag{7.6}$$

If there is a translation in the system, one can obtain:

$$\frac{M'_z}{\|t\|} \hat{m}' = \frac{M_z}{\|t\|} R \hat{m} + t_0 \tag{7.7}$$

where $t_0 = \frac{t}{\|t\|}$.

From the Equation 7.7, it is easily seen that with a given corresponding pairs, $\hat{m}_i$ and $\hat{m}'_i$, the rotation matrix R and the translation vector represented by unit vector t_0 can be computed. With the computation of unit vector t_0, only the direction of the translation is computed, but the magnitude is still missing. Because of the missing magnitude of translation vector, only the 3D coordinates of the points is computed up to a scale, which is the definition of metric reconstruction and this scale factor is the difference between the metric and Euclidean reconstruction.

Since the fundamental matrix can be computed from the projection matrix, the following equation can be obtained [6]:

$$F = K'^{-T} [t]_x R K^{-1} \tag{7.8}$$

By substituting the fundamental matrix value with the one in Equation 7.1, the relation between the essential matrix and the rotation and the translation matrix can be obtained as:

$$E = [t]_x R \tag{7.9}$$

7.2 Linear Algorithm for Determining R and t

Since the essential matrix is a rank-2 matrix like the fundamental matrix, SVD of the essential matrix is:

$$E = UDV^T \text{ where } D = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{7.10}$$

Camera matrices can be retrieved from the essential matrix up to a scale and four-fold ambiguity [6]. Define the matrix:

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.11}$$

t_0 is defined as $t_0 = U(0,0,1)^T = u_3$, the last column of U. The rotation matrix is defined as $R = UWV^T$ or $R = UW^TV^T$. Since the sign of E and t_0 can not be determined, two possible choices for R and two possible choices for t_0 and four possible choices of projection matrix rises. If the projection matrix of the left camera is $P = [I \ 0]$, the projection matrix of the right camera is one of the four possible choices:

$$\begin{aligned} P' &= [UWV^T \ u_3] \ or \ P' = [UWV^T - u_3] \\ P' &= [UW^TV^T \ u_3] \ or \ P' = [UW^TV^T - u_3] \end{aligned} \tag{7.12}$$

It is known that the 3D points projected by the cameras are in front of the cameras with positive depth values. The sign ambiguity of rotation and translation can be solved with a triangulation of a single point: Triangulate one of the corresponding points with the four possible projection matrices; the computed 3D point must be in front of the cameras, Z-axis value must be positive. Choose the projection matrix which satisfies this condition.

This method is easy to implement but it is not robust to noise which makes it impossible to be used in practical applications.

7.3 Robust Algorithm for Determining R and t

From the Equation 7.10, essential matrix and the translation matrix must satisfy the following constraint:

$$E^T t_0 = 0 \tag{7.13}$$

Since t_0 is a unit vector, the solution t_0 is the unit eigenvector of EE^T with the smallest eigenvalue. Once t_0 is found, the next step is to find its sign. The depth values of the points viewed by the cameras have to be positive because of being in front of the camera. The vectors $t_0 \times \hat{m}_i'$ and $E\hat{m}_i$ must have the same sign [19]. If the following condition hold the sign of t_0 is changed:

$$\sum_{i=1}^{n} (t_0 \times \hat{m}_i')(E\hat{m}_i) < 0 \tag{7.14}$$

The rotation matrix is determined by minimizing:

$$\min_R \left\| R^T \left[-t_0 \right]_x - E^T \right\| \tag{7.15}$$

Let $C = \left[-t_0 \right]_x$, $D = E^T$ and define a 4x4 matrix B.

$$B = \sum_{i=1}^{3} B_i^T B_i \text{ and } B_i = \begin{bmatrix} 0 & (C_i - D_i)^T \\ (D_i - C_i) & \left[D_i + C_i \right]_x \end{bmatrix} \tag{7.16}$$

where C_i and D_i are the column vectors of C and D.

$q = [q_0, q_1, q_2, q_3]^T$ is the unit eigenvector of B with the smallest eigenvalue. The rotation matrix R is related with q as:

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_2q_1 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_3q_1 - q_0q_2) \\ 2(q_1q_3 - q_0q_2) & 2(q_3q_2 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (7.17)$$

7.4 Results and Conclusion

The linear and robust algorithm are tested with the ballet sequence which provides ground truth rotation and translation matrices. The rotation and translation matrices are computed via robust and linear algorithm with ground truth fundamental matrix and calibration matrix values and the results are compared with the ground truth rotation and translation matrices.

Table 8 The calculated translation matrices and euler angles of the calculated rotation matrix via linear and robust algorithm and the ground truth values.

Method	**t_x**	**t_y**	**t_z**	**Θ_x (degree)**	**Θ_y (degree)**	**Θ_y (degree)**
Ground Truth	-3.9037	-0.0404	0.1687	-0.68	-4.6	1.49
Linear	0.999	0.0107	-0.0431	24.9	42.07	59.16
Robust	-72.5	-0.77	$3.5e^{-7}$	0.331	-2.46	-0.018

Both linear and robust algorithms give good result on translation matrix estimation. The ratio of translation among the axis is correct for both algorithms, but the direction of the translation estimated by linear algorithm is the inverse of the ground truth. It has to be corrected before triangulation. If not, the points are triangulated with negative depth values, which mean they are behind the cameras, but in real they are not.

The problem with the linear algorithm is the estimated rotation matrix and Euler angles. They are far from the ground truth.

Robust algorithm estimates the translation matrix in the same direction with the ground truth but up to a scale. As it is mentioned before, only the direction of the translation matrix can be estimated, the magnitude can be estimated up to a scale factor which leads to a scale-*ambiguity*. The estimated rotation matrix and Euler angles are nearly equal to the ground truth. The results show that robust algorithm performs better than the linear one.

CHAPTER 8

RECTIFICATION

8.1 Introduction

Rectification is the process of determining the transformation of a given stereo pair, which makes the epipolar lines of the images parallel to the horizontal axis. Rectification is the pre-step for stereo matching. Almost all stereo matching algorithms require rectified images as input.

Rectification decreases the 2D search area to 1D for stereo matching. By the transformation applied to the stereo image pair, the corresponding points have the same vertical coordinates. After the rectification, it is known that the corresponding point of each feature point is located at the same vertical coordinate, but shifted horizontally. So the search is performed on the same vertical axis in stereo matching algorithms.

Several algorithms are proposed for rectification, but a common criterion or a measure of the performance have not been proposed. The average distance between the vertical coordinates of the corresponding points is considered as a measure for the rectification algorithms examined in this thesis.

Two rectification algorithms are examined: 1. Rectification for calibrated stereo pairs [9], 2. Quasi-Euclidean Uncalibrated Epipolar Rectification [10].

8.2 Rectification for Calibrated Stereo Pairs

The rectification process in calibrated images is performed by rotating the retinal planes of the cameras until the epipolar lines coincide and the epipoles translates to infinity. The cameras are rotated while keeping the optical centers constant.

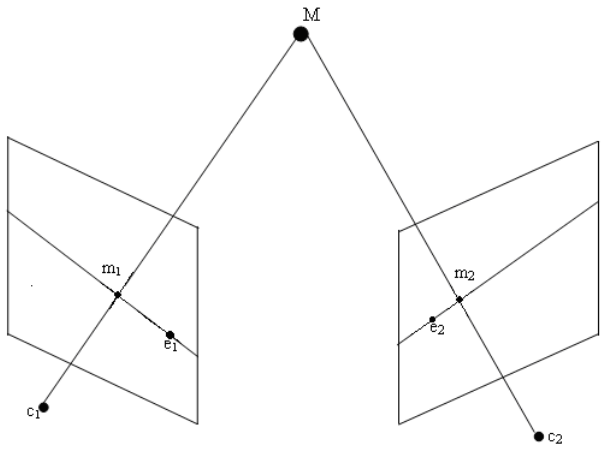


Figure 28 Epipolar Geometry

Figure 28 shows the epipolar geometry of a stereo pair. The 3D point M with the coordinates $[x, y, z, 1]^T$ is projected by a stereo camera pair with camera centers c_1 and c_2, and projection matrices $P_l = K[R \ t]$ and $P_r = K[R \ t]$, to the 2D points m_1 and m_2 with the coordinates $[u_1, v_1, 1]^T$ and $[u_2, v_2, 1]^T$.

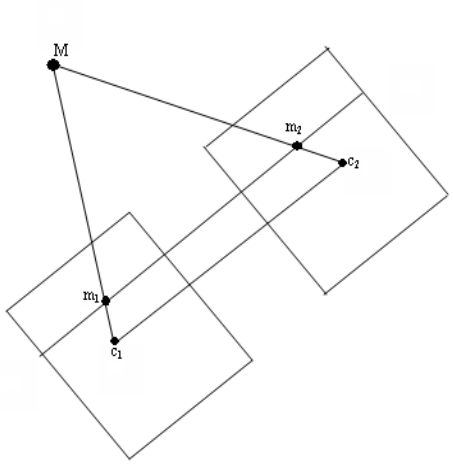


Figure 29 Rectified cameras. Epipolar lines are parallel to each other and baseline.

When the focal plane of the right camera coincides with the optical center c_1 of left camera, the right epipole is at infinity. The same concept holds for the left camera. When both epipoles are at infinity, the baseline $c_1 c_2$ is in both focal planes which means that the retinal planes are parallel to the baseline and the epipolar lines are parallel to the horizontal axis [9].

Rectification is the process of defining new projection matrices P_{nl} and P_{nr} by rotating the old ones P_{ol} and P_{or} around their optical centers until focal planes coincide [9]. In order to get epipolar lines parallel to the horizontal axis, the new X axis of the cameras must be parallel to the baseline. The camera matrices of the cameras must be same in order to have corresponding points with same vertical coordinates [9]. The new projection matrices are defined as:

$$P_{nl} = K\left[R_n - R_n c_1\right] \text{and } P_{nr} = K\left[R_n - R_n c_2\right] \quad (8.1)$$

The algorithm is summarized as follows:

- The old projection matrices P_{ol} and P_{or} are factorized to calibration, rotation and translation matrices:

$$P_{ol} = K_{ol}\left[R_{ol} \; t_{ol}\right] \text{ and } P_{or} = K_{or}\left[R_{or} \; t_{or}\right] \quad (8.2)$$

- The projection matrices can be represented as $P_{ol} = \left[Q_{ol} \;\; q_{ol} \right]$ and $P_{or} = \left[Q_{or} \;\; q_{or} \right]$. The optical centers of the cameras are:

$$c_1 = -Q_l^{-1} q_l \text{ and } c_2 = -Q_r^{-1} q_r \tag{8.3}$$

- The new rotation matrix is defined as:

$$R_n = \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix} \tag{8.4}$$

- The new X axis is parallel to the baseline:

$$r_1 = \frac{(c_1 - c_2)}{\left\| c_1 - c_2 \right\|} \tag{8.5}$$

- The new Y axis is orthogonal to X and k :

$$r_2 = k \Lambda r_1 \tag{8.6}$$

- The new Z axis is orthogonal to X and Y axis:

$$r_3 = r_1 \Lambda r_2 \tag{8.7}$$

k is an arbitrary vector to define the position of new Y axis and it is chosen as the Z unit vector of the old left rotation matrix.

- The choice of calibration matrices is arbitrary. The calibration matrix of the right camera is chosen as the calibration matrix of the new camera matrices [9].

$$K_{nl} = K_{nr} = K_{or} \tag{8.8}$$

- The new projection matrices are:

$$P_{nl} = K_{nl}\left[R_n - R_n c_l\right] \text{ and } P_{nr} = K_{nr}\left[R_n - R_n c_r\right] \quad (8.9)$$

If $P_{nl} = \left[Q_{nl} \quad q_{nl}\right]$ and $P_{or} = \left[Q_{or} \quad q_{or}\right]$, the transformation applied to the first camera projection matrix is $T_l = Q_{nl}Q_{ol}^{-1}$. It is also same for the second camera.

8.3 Quasi-Euclidean Uncalibrated Epipolar Rectification

Euclidean rectification is provided by a rotation of image planes providing that epipolar lines are parallel and horizontal. The image transformation is computed by a reference frame, plane at infinity in the calibrated case. Quasi-Euclidean rectification is an approximation to the plane at infinity as a reference plane [10]. As explained in the previous part, the transformation applied to the cameras in order to rectify them can be represented as $T_l = Q_{nl}Q_{ol}^{-1}$. Indeed T_l is the collineation referenced by plane at infinity between the old and the new cameras [10].

The transformation can be represented as:

$$T_l = K_{nl}R_l K_{ol}^{-1} \quad (8.10)$$

where K_{nl} and K_{ol} are the intrinsic parameters of the new and old cameras and R_l is the rotation applied to the old left camera to rectify it [10]. In the uncalibrated case, only the corresponding pairs are given. The aim is to compute the proper transformation in order to transform the corresponding points to satisfy the epipolar geometry of a rectified image pair.

The fundamental matrix of the rectified image pair can be represented as:

$$F_r = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} = \left[u_1\right]_x \quad (8.11)$$

where $u_1 = (1,0,0)$ and F_r is the fundamental matrix of the rectified image pair.

The relation between the corresponding pairs and the fundamental matrix, $x_i'^T \; F \; x_i = 0$, is refined as:

$$\left(T_r x_i'^T\right) F_r \left(T_l x_i\right) = 0 \tag{8.12}$$

T_l and T_r are the unknown transformations for the rectification and they must satisfy the epipolar constraint with the transformed corresponding point [10]. The transformation matrices are defined as:

$$T_l = K_{nl} R_l K_{ol}^{-1} \text{ and } T_r = K_{nr} R_r K_{or}^{-1} \tag{8.13}$$

The unknown parameters are the old intrinsic parameters, K_{ol} , K_{or} and the rotation matrices R_l , R_r . For the new intrinsic parameters, it is assumed that the vertical and horizontal focal lengths are the same and the principal point is the mid point of the image.

Consider the following equation that:

$$F = K_{or}^{-1} R_r^T F_r R_l K_{ol}^{-1} \tag{8.14}$$

The multiplication of R_l , R_r with F_r will eliminate the X components of the rotation matrices so that they can be set to zero. Using the above relations the uncalibrated rectification problem can be approximated as a least-squares problem with six unknown parameters, the rotation angles Y, Z for the left image, X, Y, Z for the right image and the focal length. Sampson error is used for the error measurement which is a first order approximation of the geometric re-projection error.

$$E_{samp}^{i} = \frac{\left(x_i'^T \; F \; x_i\right)^2}{\left\|\left[u_3\right]_x F x_i\right\|^2 + \left\|x_i'^T F \left[u_3\right]_x\right\|^2} \tag{8.15}$$

where $u_3 = (0,0,1)$

The algorithm is summarized as:

- The corresponding pairs and the fundamental matrix is given. The unknown parameters are:

  Y-left: the rotation angle around Y-axis for the left image.

  Z-left: the rotation angle around Z-axis for the left image.

  X- right: the rotation angle around X-axis for the right image.

  Y- right: the rotation angle around Y-axis for the right image.

  Z- right: the rotation angle around Z-axis for the right image.

  Focal length: the focal length of the camera pair.

  The rotation angle around X-axis for the left camera is set to zero.

- Set the initial values of the unknown parameters to zero.

- The cost function is defined as in Equation (8.15). The fundamental matrix is computed with the parameters defined in Equation (8.14).

$$K_{ol} = K_{or} = \begin{bmatrix} f & & w/2 \\ & f & h/2 \\ & & 1 \end{bmatrix}$$

  where w and h are the width and height of the image respectively. The rotation matrix R_l with Euler angles (0, Y-left, Z-left) and the rotation matrix R_r with Euler angles (X-right, Y-right, Z-right).

- Compute the average Sampson error for all of the N corresponding pair.

$$E_{samp} = \frac{1}{N}\sum_{i=1}^{N} E^{i}_{samp}$$

- Minimize the Sampson error with least-squares method.

- The translation T_l and T_r is computed with the rotation angles Y-left, Z-left, X-right, Y-right, Z-right and focal length which provide minimum Sampson error using Equation 8.13.

8.4 Results and Conclusion

Calibrated rectification and uncalibrated rectification algorithms are examined and compared. For the calibrated rectification algorithm only the projection matrices of the cameras are required. For the uncalibrated rectification fundamental matrix is required which is enough to provide the epipolar relation between the two cameras.

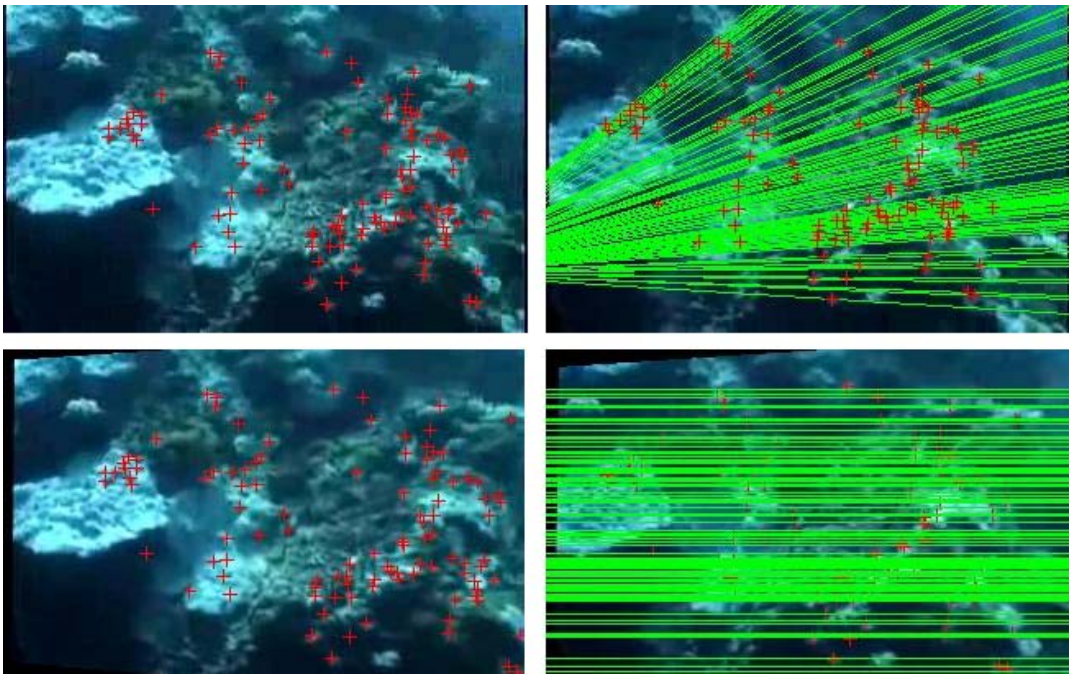


Figure 30 Coral sequence rectified with uncalibrated rectification algorithm. The upper left image is the image from left camera. The upper right is the image from

right camera with the epipolar lines and corresponding pairs. The lower left shows the rectified left image. The lower right is the rectified right image with the epipolar lines horizontal and parallel to the X-axis.

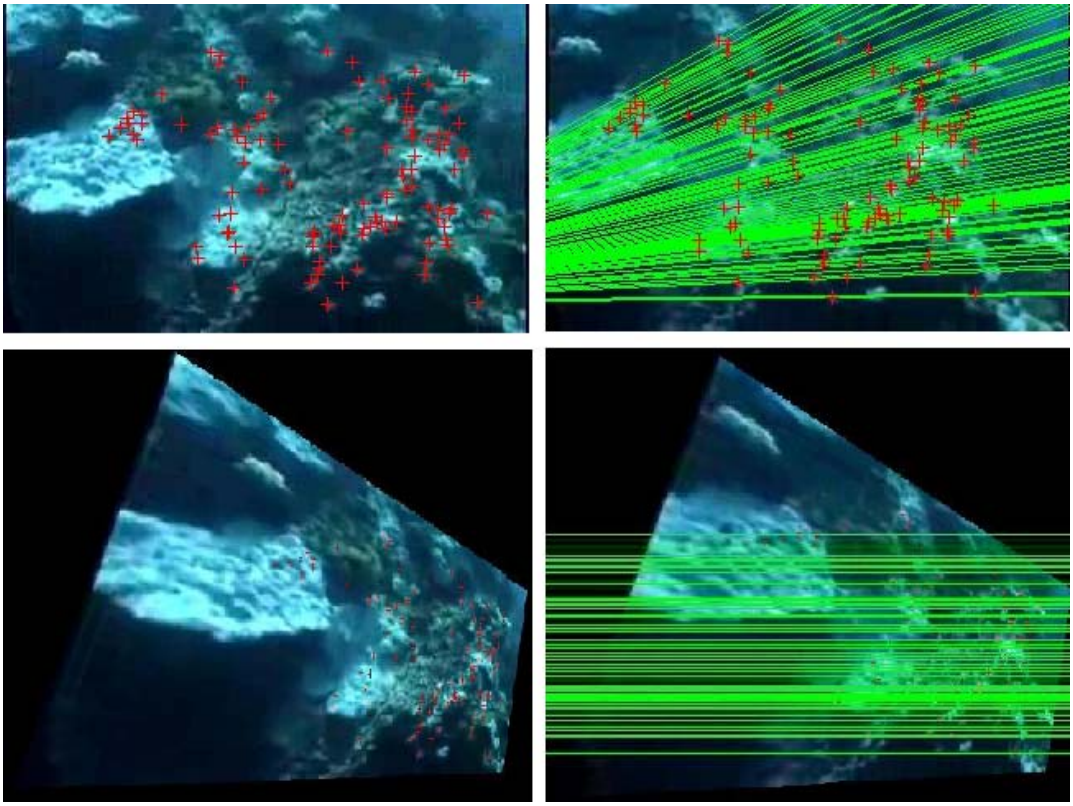


Figure 31 The coral sequence rectified with calibrated rectification algorithm.

In Figure 30 and Figure 31 the rectified images are seen with the two rectification algorithms. The biggest challenge in rectification is to rectify the images without much degeneration. Warping the images according to the computed new projection matrices is crucial because these images are the inputs of stereo matching algorithms and the performance of rectification algorithm directly affects the result of the stereo matching algorithm. The calibrated rectification gave a more distorted image according to the uncalibrated rectification. The reason is that in calibrated rectification collineations are derived directly from the camera matrices and the camera matrices are computed by auto-calibration, which is error-prone. But in uncalibrated rectification, rectification collineations are derived by minimizing the error function defined in Equation 8.15.

There is no a common criterion or performance measure of the rectification algorithms except visual comparison. But the rectified corresponding points can be compared using the properties of having the same horizontal coordinates. The mean absolute difference between the y- coordinates of the rectified corresponding points can be a measure of performance. The mean absolute difference of corresponding points after calibrated rectification is 12.8804 and the difference after uncalibrated rectification is 0.303865. The reason of the difference between the two algorithms is that the latter one naturally minimizes this error during its optimization step.

As an example for their performances, x and y coordinates of points before and after rectification are given in Table 8. In the uncalibrated rectification the y coordinates of the points (bold ones) are same which means they are parallel. But in calibrated rectification, the difference in the y coordinates of the points shows the distortion in the image.

Table 9 The corresponding points before and after the rectification.

Corresponding Pairs before Rectification		**Uncalibrated Rectification**		**Calibrated Rectification**	
m	m'	$\hat{m}$	$\hat{m}'$	$\hat{m}$	$\hat{m}'$
108.68 **74.49**	86.61 **69.46**	101.96 **72.96**	77.98 **72.49**	-70.50 **197.417**	17.11 **205.63**
77.21 **73.22**	52.32 **68.44**	155.86 **67.40**	131.82 **67.11**	69.085 **367.48**	180.58 **396.87**
…	….	….	….	…	….

CHAPTER 9

STEREO MATCHING

9.1 Introduction

Stereo matching is the process of computing the correspondence pixel for each pixel in a rectified image pair and produce a dense disparity map by which the depth map of the scene can be extracted easily. Each pixel $p = (p_x, p_y)$ in the left image must have a corresponding pixel $q = (q_x, q_y)$ in the right image. As a result of rectification the vertical coordinates of the corresponding pixels are same $p_y = q_y$. The distance between the horizontal components $(p_x - q_x)$ is called disparity and it is inversely proportional to the distance of the object to the camera [16]. So the distance of the object to the camera, depth of the scene, can be found by computing the disparity of the corresponding pixels which is possible by stereo matching.

Traditional dense stereo matching algorithms computes a dense disparity map and a depth map from the known camera motion, which is provided by rectification. The scene in stereo matching is assumed as Lambertian, without specularities, reflection or any transparency [15]. The factors that make stereo matching challenging are:

- Noise: Unavoidable light variations between the two frames, image blurring.
- Textureless areas: computing corresponding pixel pairs in textureless areas where there is no significant difference between the neighbor pixels.

- Depth Discontinuities: computing disparities and depth values near object borders.

- Occlusions: the pixels seen by only one of the cameras that causes correspondence problems.

Stereo matching is one of the most popular research topics in computer vision. A large number of stereo matching algorithms have been developed. In [11] Scharstein and Szeliski provided a good taxonomy of stereo matching algorithms. 20 stereo algorithms have been compared. Considering the overall performance of the algorithms, Graph Cut and Belief Propagation algorithms give the best performance in all regions of the sample images [11]. These two algorithms become the basis for new powerful vision algorithms. Tappens provided a comparison between these two algorithms and in his study he showed that Graph Cut performs better than Belief Propagation proving more smooth disparity map and better energy minimization [12]. It is proven that Graph Cut gives better results in underwater especially in textureless areas and near discontinuities [13]. Considering the reasons above, Graph Cut algorithm [14] is chosen as the stereo matching algorithm in this thesis.

9.2 Stereo Matching with Graph Cuts

The stereo problem is summarized as follows: compute the corresponding pixel in the right image for every pixel in the left image [14]. This problem fits a class of problems called *pixel labelling problem*. The aim is to assign each pixel a label from a set. The procedure is similar to image segmentation but it is more complex. In stereo matching problem this label set is the disparity.

Every pixel $p = (p_x, p_y) \in P$ must have a corresponding pixel $q = (q_x, q_y) \in P$ in the other image and must be assigned a label $f_p \in L$. In a labeling $f = \{f_p, p \in P\}$ the pixel p in the left image corresponds to

the pixel $p + f_p$ in the right image. During the labeling process the following factors have to be considered [16]:

- The pixel p in the left image and the pixel $p + f_p$ in the right image must have similar intensities.
- With this labeling p and $p + f_p$ must have similar labels f_p and f_q with their neighbor pixels.

The pixel labelling problem is formularized as an energy minimization problem. The goal is to find the labelling f which minimizes the energy defined as:

$$E(f) = E_{data}(f) + E_{smooth}(f) + E_{visibility}(f) \tag{9.1}$$

The energy of labeling f consists of two sub-energy forms. $E_{data}(f)$ is the cost of assigning labels. $E_{smooth}(f)$ is the smoothness term that measures the extent to which f is not piecewise smooth. Energy function must be minimized considering the criteria of finding the appropriate labeling for pixel p while pixel p must have a label which is also smooth with its neighbor pixels. $E_{visibility}(f)$ will encode the visibility constraint.

The data term is

$$E_{data}(f) = \sum_{p \in P} D_p(f_p) \tag{9.2}$$

The match penalty D_p provides the photoconsistency, the constraint of corresponding pixels to having similar intensities.

$$D_p(f_p) = \left\| I(p) - I'(p + f_p) \right\|^2 \tag{9.3}$$

The smoothness term is

$$E_{data}(f) = \sum_{\{p,q\} \in N} V_{\{p,q\}}(f_p, f_q) \tag{9.4}$$

Neighborhood constraint is introduced in the smoothness term. The neighborhood is defined as:

$$N \subset \{\{p, q\};\ p, q \in P\} \tag{9.5}$$

The pixels $p = (p_x, p_y)$ and $q = (q_x, q_y)$ are neighbors if they are in the same image and $|p_x - q_x| + |p_y - q_y| = 1$.

The smoothness penalty V provides the smoothness constraint through the neighbor pixels.

$$V(f_p, f_q) = \lambda T\lfloor f_p \neq f_q \rfloor \tag{9.6}$$

where $T[.]$ is 1 if its argument is true and 0 otherwise [14].

Figure 32 summarized the occlusion and visibility terms. The pixel p from Camera 1 (C1) and pixel q from Camera2 (C2) are shown. They are at the same disparity level, level 2 and will have the same label. The green square represents the pixel q in a different (more deep) level, behind the red dot with a different label. The pair (p,2) and (q,2) have visibility constraint, but the pair (p,2) and (q,3) do not have and they belong to I_{vis} [14].

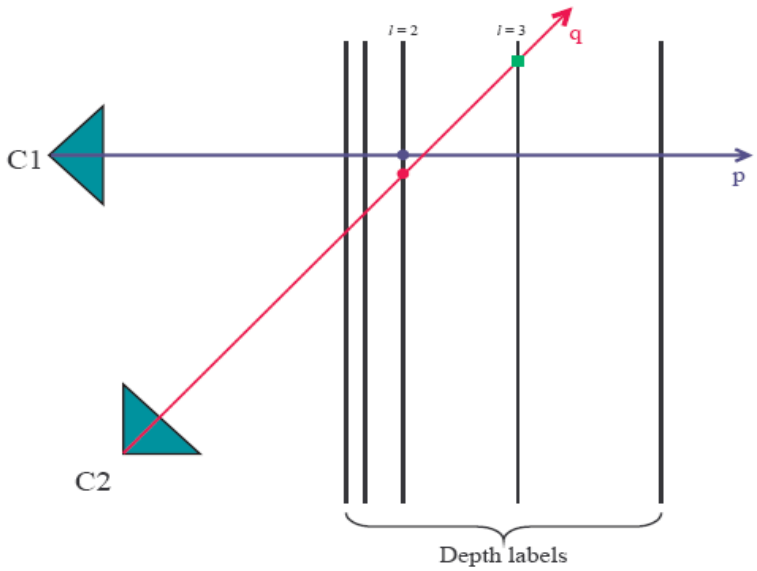


Figure 32 Occlusion and Visibility Constraint in stereo matching.

The visibility term is zero if visibility constraint is satisfied; otherwise it is infinity [14]. According to the visibility constraint, a 3D point $\langle p, f_p \rangle$ is present in a labeling f , it blocks the views of other cameras. The new set I_{vis} contains the 3D points $\langle p, f_p \rangle$ and $\langle q, f_q \rangle$ where $f_q \rangle f_p$.

$$E_{visibility}(f) = \sum_{\langle p, f_p \rangle, \langle q, f_q \rangle \in I_{vis}} \infty \tag{9.7}$$

Minimizing the energy function defined in Equation 9.1 is NP-hard problem [14]. For that reason, the goal is to find an approximation to the optimization algorithm based on graph cut to find a local minimum.

9.2.1 α Expansion Move Algorithm

If the smooth penalty function V is a metric on labels, the energy function defined in Equation 9.1 can be minimized by α expansion move algorithm [14].

Given a label α , a labeling f and another labeling f' is an α expansion move from f to f' for every pixel p .

$$f'(p) \neq f(p) \Rightarrow f'(p) = \alpha \tag{9.8}$$

A label, α , is chosen from the label set and the expansion move algorithm finds a unique labeling within a single α expansion move from the current labeling and updates the labeling if its energy is lower. Termination criteria is to reach such a labeling $\hat{f}$ that there is no α expansion move from $\hat{f}$ whose energy is lower than $E(\hat{f})$.

Finding the lowest energy α expansion move from f is the key problem. In expansion move algorithm each pixel has two options [14]:

- keep its old label f_p or,
- switch to the new label α .

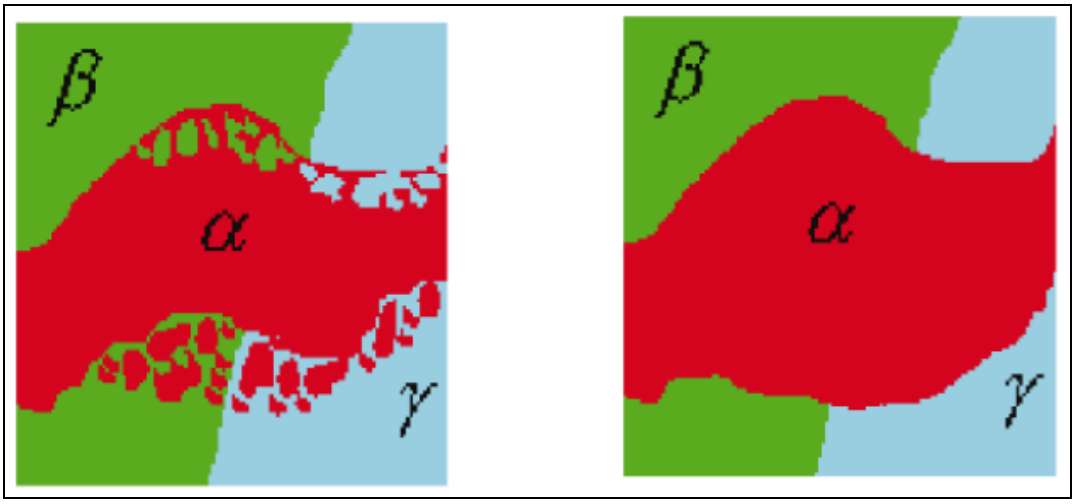


Figure 33 α Expansion Move Algorithm. Three different labeling is shown on the left part. After the expansion move algorithm, the pixels whose label is not α, is switched to label α.

Considering the options of the pixels, expansion move can be defined as a binary image that a single bit is defined for each pixel representing which option that pixel selects in the expansion move [14].

A binary image $x = \{x_p,\ p \in P\}$ is defined. Given an initial labeling f and a label α, if $x_p = 1$, $f'(p) = \alpha$ and if $x_p = 0$ then $f'(p) = f(p)$. This labeling is defined as $f^{\alpha}[x]$ [14]. The energy function E is rewritten as $\varepsilon(x) = E(f^{\alpha}[x])$. The energy function on the binary image is defined as [14]:

$$\varepsilon(x) = \sum_{p} \varepsilon_p(x_p) + \sum_{p,q} \varepsilon_{p,q}(x_p, x_q) \tag{9.9}$$

The data term is defined as:

$$\begin{aligned} &if\ \ x_p = 1\ \ \varepsilon_p(x_p) = D_p(\alpha) \\ &if\ \ x_p = 0,\ \varepsilon_p(x_p) = D_p(f_p) \end{aligned} \tag{9.10}$$

The smoothness term is defined as:

$$\begin{aligned} &if\ \ x_p = 1, x_q = 1,\ \varepsilon_{p,q}(x_p, x_q) = V(\alpha, \alpha) \\ &if\ \ x_p = 1, x_q = 0,\ \varepsilon_{p,q}(x_p, x_q) = V(\alpha, f_q) \\ &if\ \ x_p = 0, x_q = 1,\ \varepsilon_{p,q}(x_p, x_q) = V(f_p, \alpha) \\ &if\ \ x_p = 0, x_q = 0,\ \varepsilon_{p,q}(x_p, x_q) = V(f_p, f_q) \end{aligned} \tag{9.11}$$

The energy function $\varepsilon(x)$ can be minimized if $\varepsilon_{p,q}$ has *regularity* property [14]. The regularity property proposes that:

$$V(\alpha, \alpha) + V(l, l') \le V(l, \alpha) + V(\alpha, l') \tag{9.12}$$

for any labels l, l', α [14]. Since $V(\alpha, \alpha) = 0$ then the regularity property is just triangle inequality [14].

9.2.2 Graph Cuts

G is a weighted graph with vertices ***V*** and edges ***E***. Vertices represent the pixels and edges connect the vertices to each other with a weight defined as the data penalty or smooth penalty. There exist two terminal vertices {***s***, ***t***}, source and sink terminals. A cut ***C*** separates the graph from the edges into two parts; some vertices are in the source part $s \in V^s$ and the others are left in the sink part $t \in V^t$. The cost of the cut ***C*** equals to the sum of the edges between the vertex V^s and vertex V^t. The goal is to find the cut with the smallest cost. The minimum cut on graph ***G*** provides the labeling that minimizes the energy function within one α expansion move [17].

9.2.3 Graph Construction

The structure of the graph is determined by the current labeling and the label α .

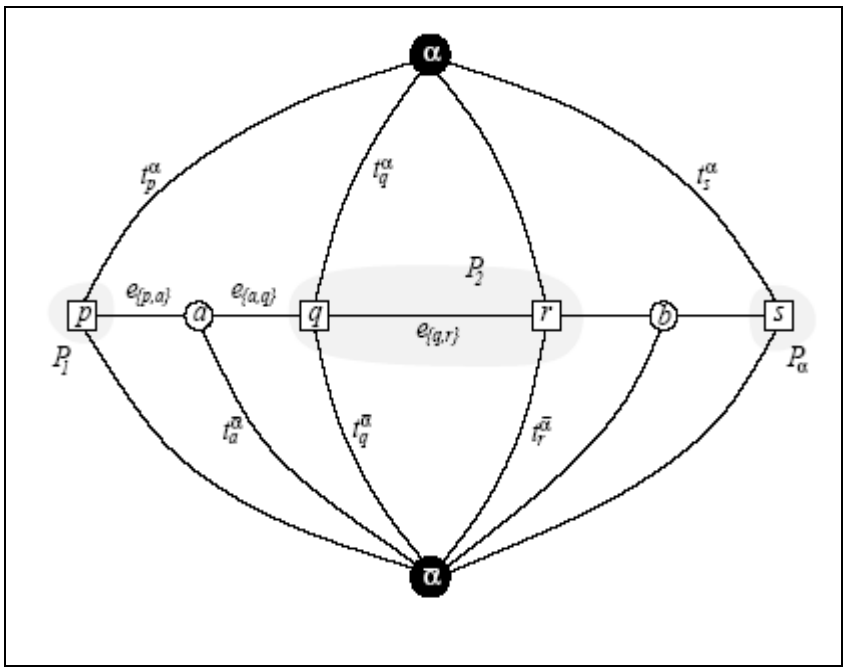


Figure 34 The configuration of the current labeling f and the new label α .

Figure 34 shows the configuration of the current labeling f and the new label α . The current labeling, the partition P , consists of the labels $\{P_1, P_2, P_\alpha\}$. The pixel p is set to label P_1 , the pixels q and r P_2 and the pixel s, P_α . Two nodes are added to the graph between the neighbor pixels with different labels, $a = a_{\{p,q\}}$ and $b = a_{\{r,s\}}$ .The vertices also includes the terminals α and $\overline{\alpha}$. Each pixel is connected to the terminals α and $\overline{\alpha}$ by t-links, t_p^α and $t_p^{\overline{\alpha}}$. Neighbor pixels which have the same labels are connected by n-links $e_{\{p,q\}}$ [17]. The weights of the edges are:

Table 10 The weights of the edges in the graph.

Edge	Weight	Reason
$t_p^{\overline{\alpha}}$	∞	$p \in P_\alpha$
$t_p^{\overline{\alpha}}$	$D_p(f_p)$	$p \notin P_\alpha$
t_p^{α}	$D_p(\alpha)$	$p \in P$
$e_{\{p,a\}}$	$V_{p,q}(f_p, \alpha)$	$\{p,q\} \in N, f_p \neq f_q$
$e_{\{a,q\}}$	$V_{p,q}(\alpha, f_q)$	
$t_a^{\overline{\alpha}}$	$V_{p,q}(f_p, f_q)$	

$e_{\{p,q\}}$	$V_{p,q}(f_p, \alpha)$	$\{p,q\} \in N,\ f_p = f_q$

Any cut ***C*** on graph ***G*** must contain one t-link for each pixel. This defines the new labeling f^c and for each pixel p :

$$\begin{aligned} f_p^c &= \alpha \quad if\ t_p^\alpha \in C \\ f_p^c &= f_p \quad if\ t_p^{\overline{\alpha}} \in C \end{aligned} \tag{9.13}$$

If the cut ***C*** separates the pixel p form the terminal α , the pixel p is set to label α , otherwise it is set to its old label f_p , if the cut separates p from the terminal $\overline{\alpha}$ [17].

9.3 Results and Conclusion

The graph cut stereo matching algorithm is briefly discussed above. Three stereo matching algorithms are compared in this section. First two algorithms are sub-types of graph cut algorithms and the last one is a traditional stereo matching algorithm. The first graph cut algorithm is called voxel labeling (KZ1) [17]. The second graph cut algorithm is called pixel labeling (KZ2) [25], and the last algorithm is a traditional stereo algorithm (BVZ) [26]. [26] focuses on solving the stereo matching problem by maximum a posteriori (MAP) estimate of a class of a Markov Random Fields which generalizes the Potts model and ignore the occlusions. The difference between KZ1 and KZ2 is the handling method of smoothing terms in the energy function. For more detail about the algorithms, reader may refer to the following references. [17] for KZ1, [25] for KZ2 and [26] for BVZ.

These algorithms are first applied to a very well known stereo pair, Tsukuba pair. Tsukuba pair, the ground truth disparity map and occlusions are shown in Figure 35.

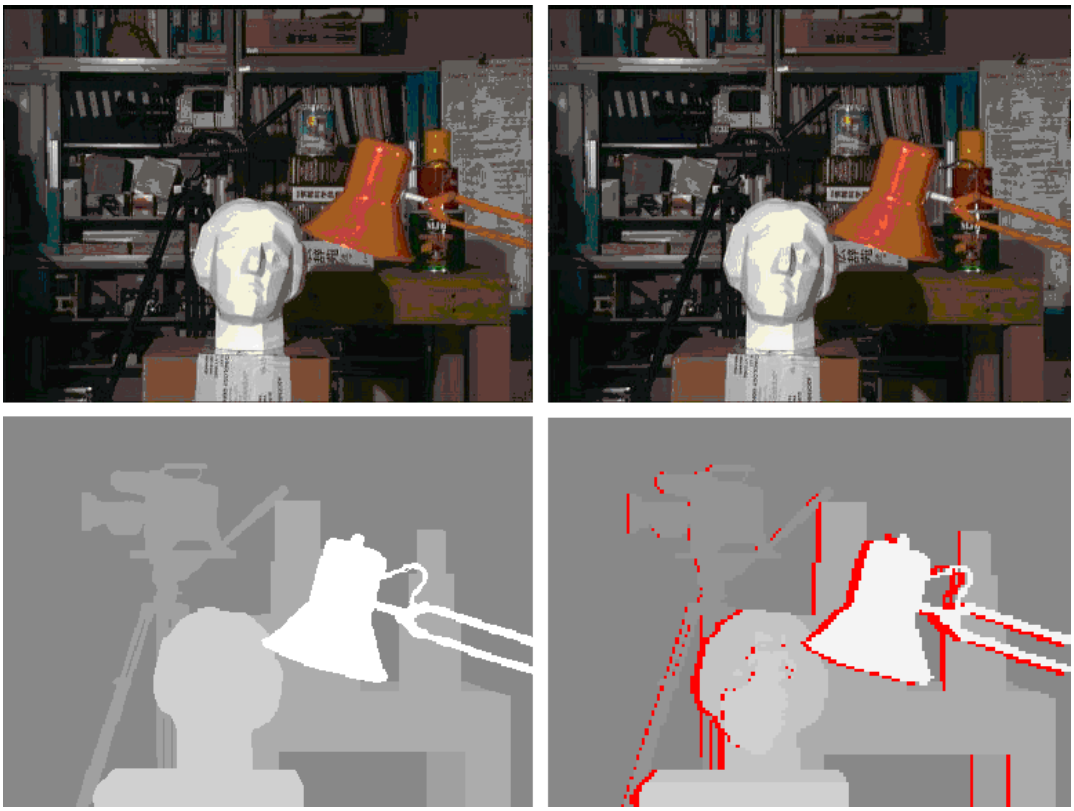


Figure 35 Tsukuba sequence. The lower left image is the ground truth for the disparity maps. The lower right image (red marked points) shows the occlusions between the two images.

Figure 36 shows the result of the stereo algorithms on Tsukuba pair. Here, the estimated disparity map and occluded regions are given for each algorithm.

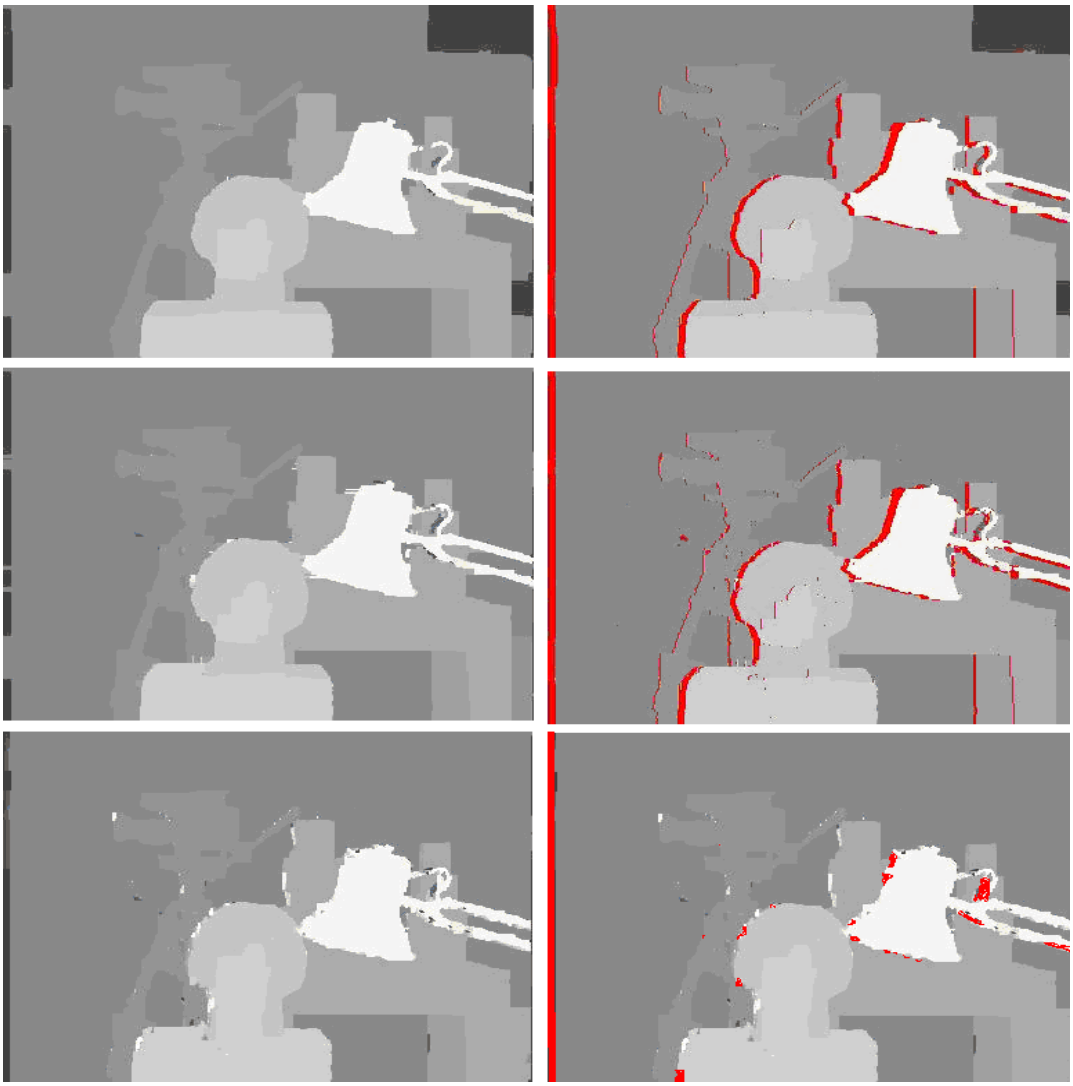


Figure 36 The performance of stereo matching algorithms on Tsukuba sequence. The red marks on images in the right part shows the occlusions. The first pair is KZ1, the second pair KZ2 and the last pair is BVZ.

In the disparity maps, the objects close to the camera are brighter than the objects far from the camera. With this scaling the disparity maps behave as a depth map of the scene.

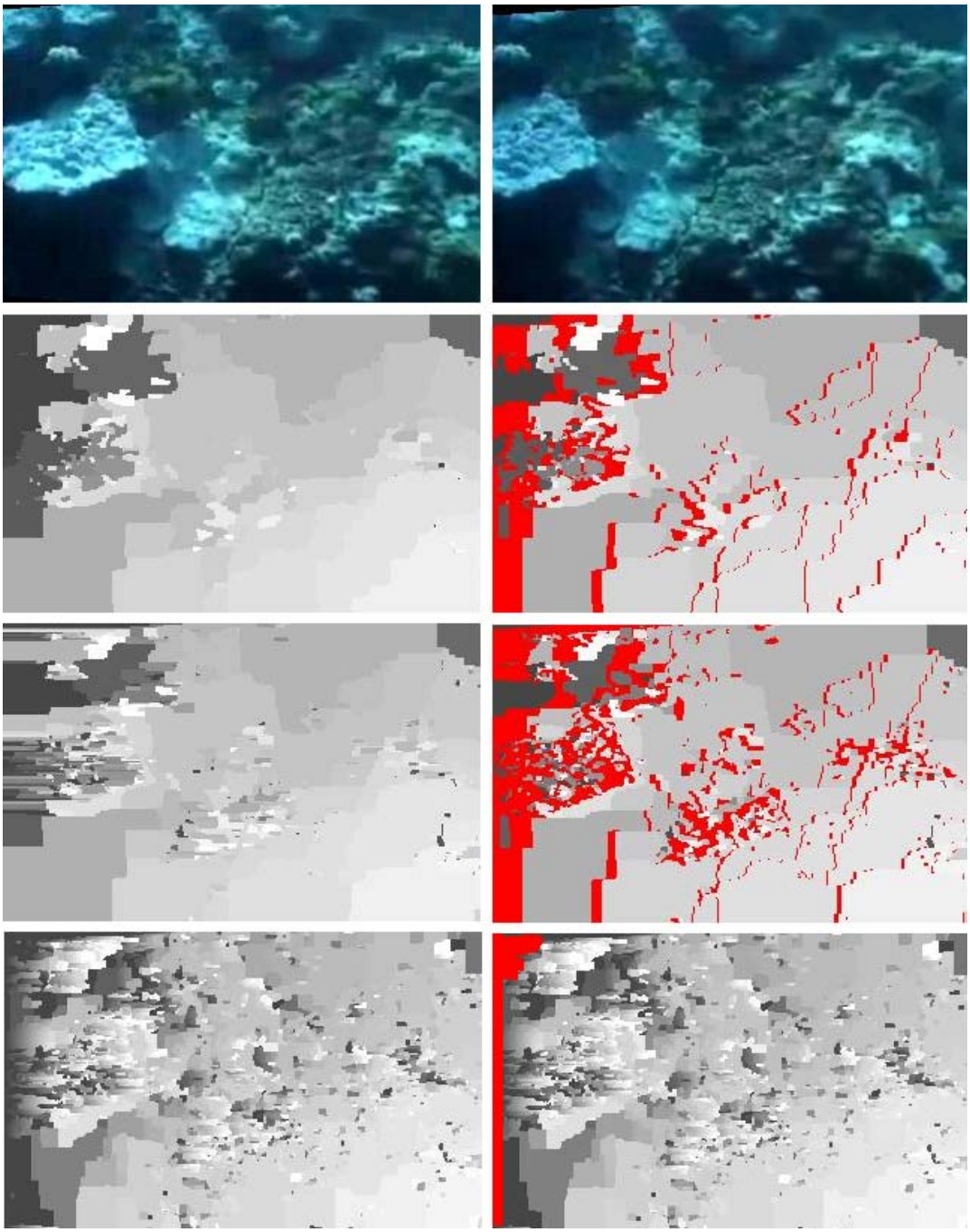


Figure 37 The performance of stereo matching algorithms on coral sequence. The red marks on images in the right part shows the occlusions. The first pair is KZ1, the second pair KZ2 and the last pair is BVZ.

These algorithms are also applied to underwater stereo pair. The resultant disparity maps and occluded regions are shown in Figure 37 and the same type of performance are observed.

From Figure 37 it is seen that graph cut algorithms performs better than traditional stereo algorithm and among the graph cut algorithms KZ1 performs better than KZ2. The smoothness in the object borders is the advantage of KZ1 and there are also more less spikes in the sloping sides in KZ1 when compared to KZ2.

CHAPTER 10

TRIANGULATION

10.1 Introduction

The final step of the 3D reconstruction process is the triangulation step where the coordinates of 3D points are computed from the corresponding pairs and the projection matrix of the cameras. In theory, since the 3D point is visualized by both of the cameras, left and right, the back-projected rays from the cameras should meet at the location of 3D point. Due to the noise in determining the corresponding pairs and estimating the fundamental matrix, the back-projected rays do not meet. In this case, it is necessary to find the best 3D point of intersection [23].

Assume a 3D point M in 3D space viewed by two cameras whose projection matrices are P and P' respectively and also m and m' are the projected points in the two images satisfying the epipolar constraint $m'^{T} F m = 0$. The two rays back-projected from the points m and m' lie on an epipolar plane that passes from the camera centers. Since the rays lie on plane they must intersect in a point, which is the 3D point projects via the left and right cameras to the points m and m'. But these rays do not intersect because of the presence of noise. The goal of triangulation is to back-project these rays and to intersect them in 3D space.

Several algorithms have been developed for triangulation. Two of most common methods, linear triangulation and polynomial triangulation are examined in this thesis.

Assume a triangulation method τ to compute the 3D point M from corresponding points m and m' with the cameras with projection matrices P and P'.

$$M = \tau(m, m', P, P') \tag{10.1}$$

10.2 Linear Triangulation

Linear triangulation method is the most common and simple triangulation method to overcome the problem defined above. The 3D point M is projected via the two cameras to the points $\lambda m \approx PM$ and $\lambda' m' \approx P'M$, where λ and λ' are scalar factors. Cross multiplication method is applied in order to eliminate the scale factors $m \times PM = 0$. If the relation is written in the open form, it gives three equations two of which are independent and linearly dependent to M.

$$\begin{aligned} u(p_3^T M) - (p_1^T M) &= 0 \\ v(p_3^T M) - (p_2^T M) &= 0 \\ u(p_2^T M) - v(p_1^T M) &= 0 \end{aligned} \tag{10.2}$$

where p_i^T is the i-th row of P. Equation 10.2 can also be written for the right image and an equation of the form $AM = 0$ can be composed:

$$A = \begin{bmatrix} up_3^T - p_1^T \\ vp_2^T - p_1^T \\ u'p_3'^T - p_1'^T \\ v'p_2^T - p_1'^T \end{bmatrix} \tag{10.3}$$

The solution of $AM = 0$ is the unit eigenvector corresponding to the smallest eigenvalue of A.

10.3 Polynomial Triangulation

The noisy point correspondences do not meet the epipolar constraint. If $\hat{m}$ and $\hat{m}'$ are the point correspondences close to m and m' but satisfying the epipolar constraint, a geometric error cost function is defined:

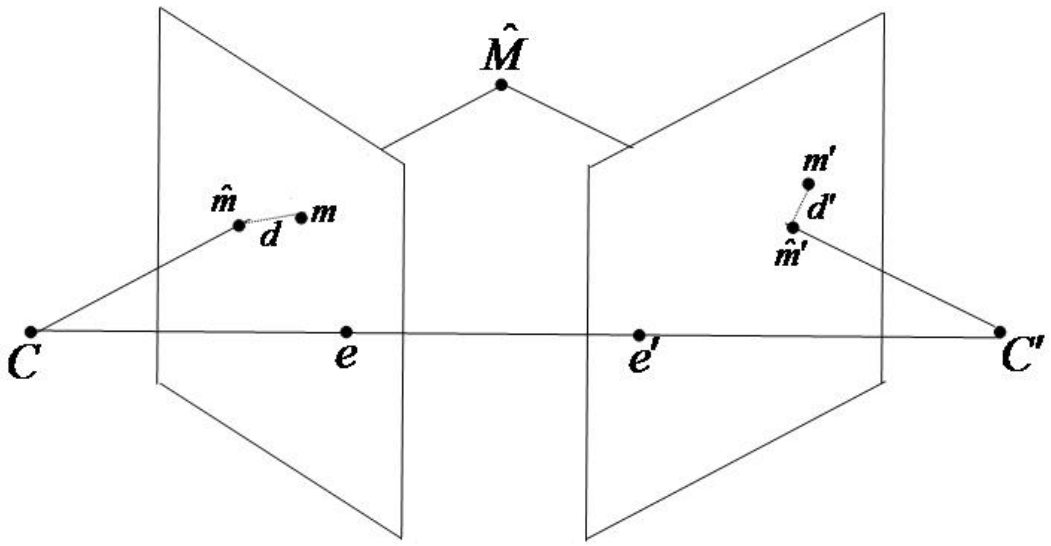


Figure 38 Polynomial triangulation computes the closest corresponding point which minimized the error function defined in Equation 10.4.

$$C(m, m') = d(m, \hat{m})^2 + d(m', \hat{m}')^2 \qquad (10.4)$$

where $d(*,*)$ is the Euclidean distance between the points. Once the point correspondences $\hat{m}$ and $\hat{m}'$ are found which minimizes the cost function, the 3D point $\hat{M}$ can be computed by linear triangulation and the back-projected rays will precisely intersect [6]. This cost function can be minimized using optimization algorithm like Levenberg-Marquardt, but polynomial triangulation provides a non-iterative method, the solution of six-degree polynomial.

Any corresponding pair that satisfies the epipolar constraint must lie on the corresponding epipolar lines in the two images. $\hat{m}$ is the optimum point lies on the epipolar line l and $\hat{m}'$ is the optimum point lies on the epipolar line l'. Since the cost function is the Euclidean distance between the measured points and the

optimal points, the closest distance between these points are the perpendicular distance between the measured point and the corresponding epipolar line. So that the cost function can be rewritten as:

$$C(m, m') = d(m, l)^2 + d(m', l')^2 \tag{10.5}$$

Therefore $\hat{m}$ and $\hat{m}'$ are the closest points on lines l and l' to the points m and m'. The strategy of minimization is as follows:

1. Parameterize the epipolar lines in the left image with t. An epipolar line in the left image is written as $l(t)$.

2. By using the fundamental matrix F, define the epipolar lines in the right image as $l'(t)$.

3. The cost function is defined as

$$C(m, m') = d(m, l(t))^2 + d(m', l'(t))^2 \tag{10.6}$$

4. Find the value of t which minimizes the cost function.

By a suitable parameterization of the epipolar lines, the cost function turns to a polynomial function of t [6].

First step of the minimization algorithm is to apply a transformation to the corresponding points to place them at the origin, $[0\ 0\ 1]^T$. With this transformation the epipoles is on the X-axis with coordinates, $[1\ 0\ \ f\ ]^T$ and $[1\ 0\ \ f'\,]^T$ and the fundamental matrix has a special form:

$$F = \begin{bmatrix} ff'd & -f'c & -f'd \\ -fb & a & b \\ -fd & c & d \end{bmatrix} \tag{10.7}$$

Suppose a point with the coordinate $[1 \;\; t \;\; f]^T$. Since the epipole is $[1 \;\; 0 \;\; f]^T$, the equation of the epipolar line of this point is $[1 \;\; t \;\; f] \times [1 \;\; 0 \;\; f] = [tf \;\; 1 - t]$ which is of the form $l(t)$. The squared distance between the epipolar line and the origin is:

$$d(m, l(t))^2 = \frac{t^2}{1 + (tf)^2} \tag{10.8}$$

The corresponding epipolar line in the right image is:

$$l'(t) = F[0 \;\; t \;\; 1]^T = [-f'(ct + d) \;\; at + b \;\; ct + d]^T \tag{10.9}$$

The squared distance of this line to the origin is:

$$d(m', l'(t))^2 = \frac{(ct+d)^2}{(at+b)^2 + f'^2(ct+d)^2} \tag{10.10}$$

The total squared distance is:

$$s(t) = d(m, l(t))^2 + d(m', l'(t))^2 = \frac{t^2}{1+(tf)^2} + \frac{(ct+d)^2}{(at+b)^2 + f'^2(ct+d)^2} \tag{10.11}$$

The derivative of the function is:

$$s'(t) = \frac{2t}{(1+f^2t^2)^2} - \frac{2(ad-bc)(at+b)(ct+d)}{((at+b)^2 + f'^2(ct+d)^2)^2} \tag{10.12}$$

The maxima and minima of $s(t)$ will occur when $s'(t) = 0$. Equating the Equation 10.12 to zero will give:

$$
\begin{aligned}
g(t) &= t((at+b)^2 + f'^2(ct+d))^2 \\
&\quad -(ad-bc)(1+f^2t^2)^2(at+b)(ct+d) \\
&= 0
\end{aligned}
\tag{10.13}
$$

The maxima and minima of $s(t)$ occur at the roots of the polynomial defined in Equation 10.13. The polynomial $g(t)$ is a six-degree polynomial having six roots. The real root giving the minimum value of $s(t)$ is the value of t that is looked for. The compete polynomial triangulation is summarized as [6]:

1. Define the transformation matrices that take $m = [u \;\; v \;\; 1]^T$ and $m' = [u' \;\; v' \;\; 1]^T$ to the origin:

$$
T = \begin{bmatrix} 1 & & u \\ & 1 & v \\ & & 1 \end{bmatrix} \quad T' = \begin{bmatrix} 1 & & u' \\ & 1 & v' \\ & & 1 \end{bmatrix}
\tag{10.14}
$$

2. Replace F with $T'^{-T} F T^{-1}$. The new F corresponds to translated coordinates.

3. Compute the left and right epipole $e = [e_1 \;\; e_2 \;\; e_3]^T$ and $e' = [e'_1 \;\; e'_2 \;\; e'_3]^T$ via $Fe = 0$ and $F^T e' = 0$. Normalize the epipoles such that $e_1^2 + e_2^2 = 1$.

4. Form the rotation matrices

$$
R = \begin{bmatrix} e_1 & e_2 & \\ -e_2 & e_1 & \\ & & 1 \end{bmatrix} \quad R' = \begin{bmatrix} e'_1 & e'_2 & \\ -e'_2 & e'_1 & \\ & & 1 \end{bmatrix}
\tag{10.15}
$$

5. Replace F with $R'FR^T$. The resultant F has the form in Equation 10.7

6. Set $f = e_3$, $f' = e'_3$, $a = F_{22}$, $b = F_{23}$, $c = F_{32}$, $d = F_{33}$.

7. Form the polynomial $g(t)$ and solve for t to get the six roots.

8. Evaluate the cost function defined in Equation 10.10 for the real-part of the roots of $g(t)$. Select the value of t as $t_{\min}$ that gives the minimum value of the cost function.

9. Evaluate the two lines $l = [tf \quad -1 \quad t]^T$ and l' defined in Equation 10.9 with $t_{\min}$ and find $\hat{m}$ and $\hat{m}'$ that are the closest points on the epipolar lines to the origin. For a general line $[\lambda \quad \mu \quad \nu]^T$ the closest point to the origin is $[-\lambda\nu \quad -\mu\nu \quad (\lambda^2 + \mu^2)]^T$.

10. Transfer back to the original coordinate system by replacing $\hat{m}$ with $T^{-1}R^T\hat{m}$ and $\hat{m}'$ with $T'^{-1}R'^T\hat{m}'$.

11. Compute the 3D point $\hat{M}$ via linear triangulation.

10.4 Results and Conclusion

Triangulation can be defined as a sparse 3D reconstruction. Only the corresponding points are reconstructed. For that reason, in this section only the images that have over 500 corresponding points are triangulated. For an underwater image, it is hard to find over 500 corresponding points, because the underwater scene structure has not been textured very much.

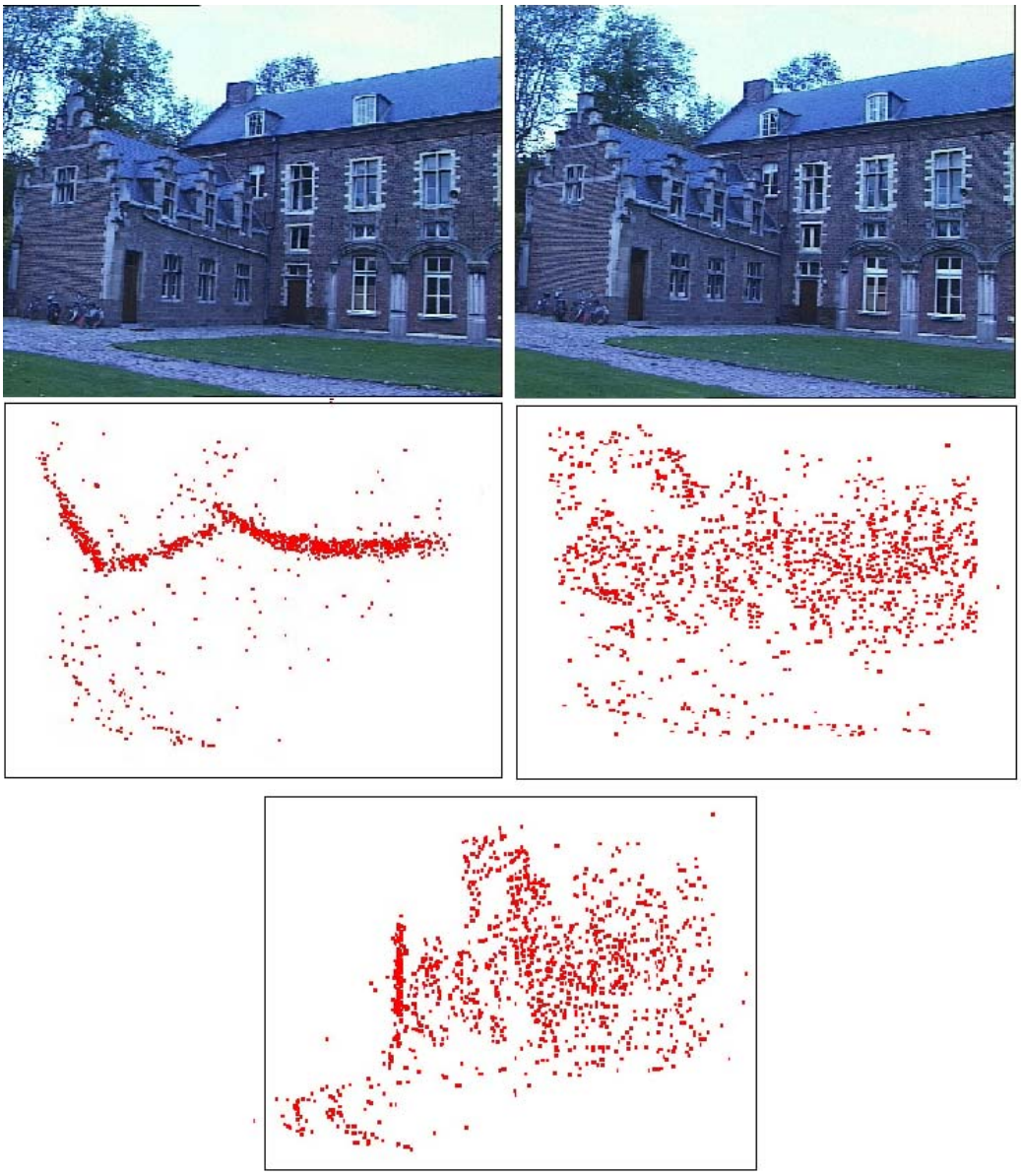


Figure 39 Castle sequence. The top pair are the images from the left and the right camera. The middle left shows the 3D reconstructed model from top view, the middle right from front view. The last image shows the 3D model from side view.

10.5 Dense 3D Reconstruction

Since in the triangulation method, only the corresponding points are triangulated, only %2~3 of the points are triangulated, which is called sparse 3D reconstruction. The size of the images in castle sequence is 768x576 which means 442368 points. When only the 3D coordinates of the corresponding point are calculated, this makes 1305 points which provides a poor visualization.

Since each pixel's correspondence is computed in stereo matching, it is possible to find the 3D points projected to each pixel in the image. By this way, 3D model of the whole scene can be reconstructed, which is called dense 3D reconstruction.

The first step of dense reconstruction is to compute the disparity map, which is discussed in the stereo matching. Occlusions cause holes in the disparity map. These are filled according to the disparity values of neighbor pixels.

The second step is the spike removal. The pixels which have a large disparity difference according to the neighbor pixels are called spikes. If there is a large difference in disparity value between a pixel and its neighbors, the disparity of that pixel is replaced by the average disparity value of its neighbors. The spikes are removed with this process.

The last step is smoothing. The disparity map is filtered by a Gaussian filter in order to smooth the disparity map and to provide a continuous surface.

These post-processing steps are applied to the disparity of coral pair and the result is shown in Figure 40.

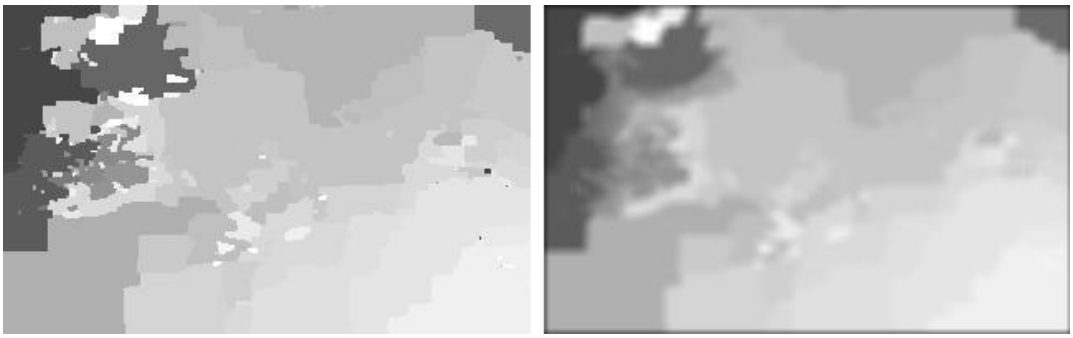


Figure 40 The disparity map of the coral sequence. The left part is the disparity map before spike removal and smoothing and the right part is the smoothed disparity map with removed spikes.

The disparity values of the pixels are not their actual z-coordinate. Disparity represents a relative depth value. The disparity value is be transformed to the actual depth value as follows:

$$Z = \frac{1}{\left(\frac{d}{255} \left(\frac{1}{Min_{Z}} - \frac{1}{Max_{z}} \right) + \frac{1}{Max_{z}} \right)} \tag{10.16}$$

where d is the disparity value, Min_{Z} and Max_{Z} are the minimum and maximum depth values in the images. These values are estimated empirically and set to (80,200) for the coral sequence.

Once the depth value of the 3d point is computed, the computation of X and Y are left. The 2D projection of the 3D point and the camera projection matrix is known. The X and Y coordinates of 3D point can be calculated by solving the equation:

$$m = PM \tag{10.17}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{10.18}$$

The steps of the solution are:

- Compute the values c_0 , c_1 and c_2 :

$$\begin{aligned} c_0 &= ZP_{02} + P_{03} \\ c_1 &= ZP_{12} + P_{13} \\ c_2 &= ZP_{22} + P_{23} \end{aligned} \tag{10.19}$$

where P_{ij} is the i-th row and j-column of the projection matrix P .

- Define the equations:

$$\begin{aligned} u &= XP_{00} + YP_{01} + c_0 \\ v &= XP_{10} + YP_{11} + c_1 \\ 1 &= XP_{20} + YP_{21} + c_2 \end{aligned} \tag{10.20}$$

By solving the three equations defined in Equation (10.19), the X and Y-coordinates is calculated as:

$$X = \frac{Y(P_{01} - uP_{21}) + c_0 - uc_2}{uP_{20} - P_{00}} \tag{10.21}$$

$$Y = \frac{u(c_1P_{20} - c_2P_{10}) + v(c_2P_{00} - c_0P_{20}) + c_0P_{10} - c_1P_{00}}{v(P_{20}P_{01} - P_{21}P_{00}) + u(P_{10}P_{21} - P_{11}P_{20}) + (P_{00}P_{11} - P_{10}P_{21})} \tag{10.22}$$

For each pixel, the 3D projected point is computed. By this way the 3d model of the whole image is reconstructed.

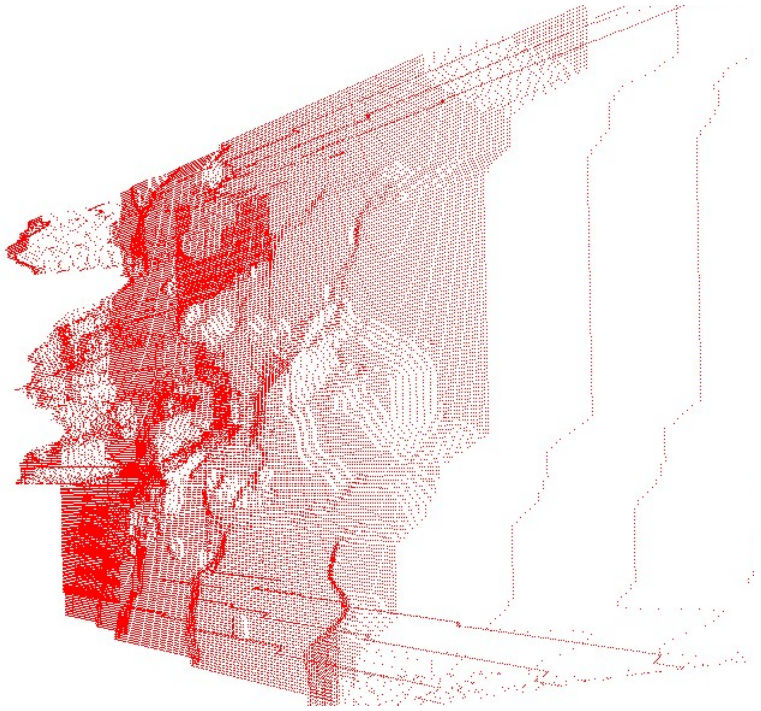


Figure 41 The 3D point cloud of coral sequence. The reconstructed point number is 76800.

The 3D point cloud is triangulated in order to form a mesh and the result mesh is covered with a default surface.

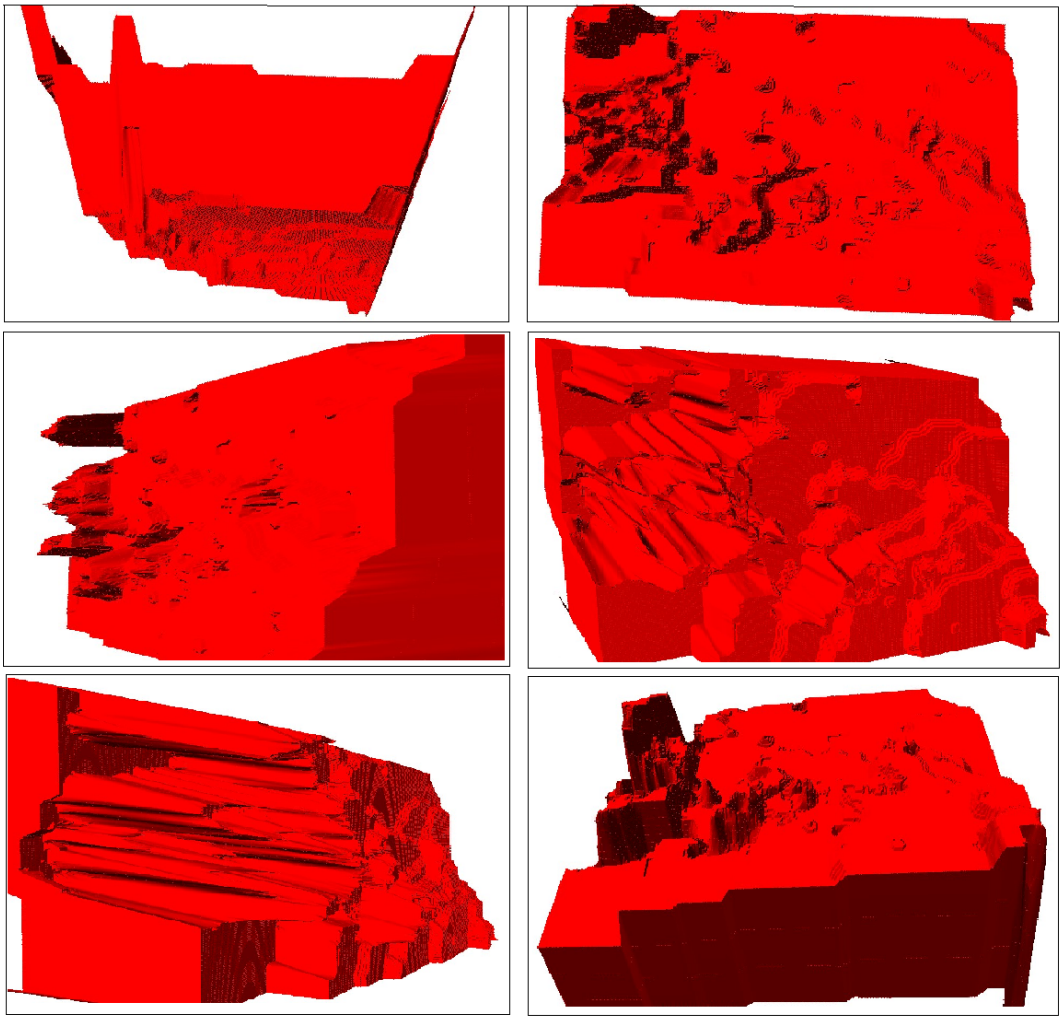


Figure 42 The dense 3D reconstruction of coral sequence.

The 3D representation will be more realistic by texture mapping the mesh model. The image taken from the left camera is chosen as the texture. A polygon is formed by using the reference pixel and its right and down neighbors. Since mesh is formed by polygons, each texture polygon is mapped to the surface polygon.

CHAPTER 11

EXPERIMENTS

11.1 Introduction

This chapter is devoted to the 3D reconstruction of different types of underwater scenes. Three sequences are examined. First one is the coral sequence, which is used throughout the thesis to demonstrate the sub-blocks of the reconstruction process and to compare the algorithms. The second one is boat sequence and the last is another coral sequence but this sequence is different from the first two by the motion of the camera. In the first two sequence the camera makes a translational motion with a relatively small rotation around the object, but in the last sequence the camera makes translational motion along the principal axis with again a relatively small rotation and the effect of this different motion is examined.

11.2 Coral 1 Sequence

Figure 43 shows two frames from a video sequence captured from a camera following a quasi-circular translational path around a coral reef in clear water.

Figure 43 Chosen two frames from the coral sequence. The left image represents the left camera and the right image represents the right camera in the stereo pair.

The two frames are preprocessed with the method examined in Chapter 3. The resultant images are shown in Figure 44. It seems to decrease the visibility quality but it increases the number of detected features.

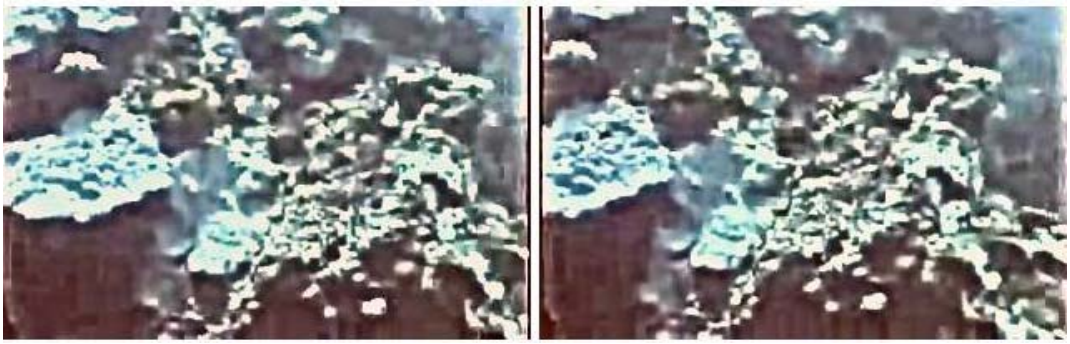


Figure 44 The result of preprocessing of coral sequence.

Once the images are preprocessed, the next step is to detect features, find corresponding pairs and compute the fundamental matrix which represents the epipolar geometry between the two cameras. The feature points are detected with SIFT method and matched with SIFT descriptors. The fundamental matrix is estimated via normalized 8-point algorithm and RANSAC. In Figure 45, the left image shows the matched corresponding points and their motion between the two images on the left image. The red cross shows their position in the left image and the green cross shows their position in the right image. The blue line represents their path between the two frames. In Figure 45, the right image shows the epipolar lines wit green lines and the feature points with red cross.

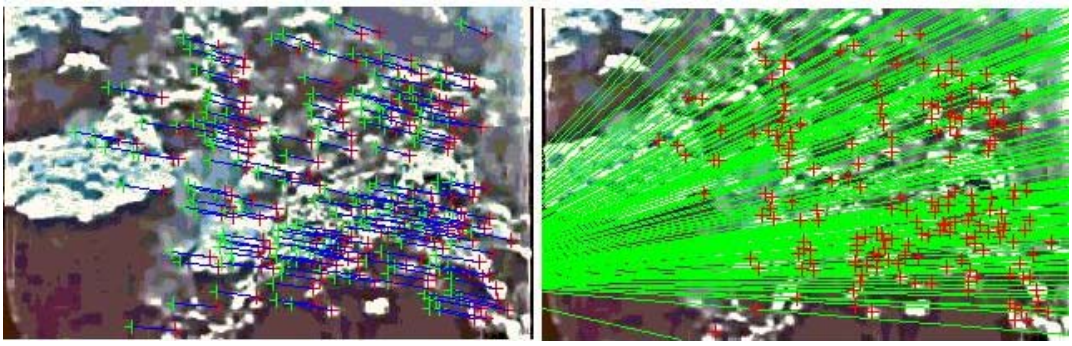


Figure 45 The path of corresponding points between the two frames is shown in the left and the epipolar lines and feature points is shown in the right image.

With the estimated fundamental matrix, the intrinsic parameters are computed via simplified Kruppa equations and the extrinsic parameters are computed via robust method of rotation and translation matrices. Once the intrinsic and extrinsic parameters are computed, the next step is rectification. The two frames are rectified via uncalibrated rectification algorithm, since it performs better than the calibrated rectification and give less distorted images. Figure 46 shows the rectified image pair.

Figure 46 The coral sequence rectified via uncalibrated rectification algorithm.

To perform dense reconstruction, the correspondence pixel of each pixel in the image should be computed. The correspondence of each pixel and disparity map is computed with graph cut based stereo matching. In Figure 47, the left part shows the disparity map of the coral sequence and the right part shows the occluded parts in the disparity map. The occluded regions are filled according to their neighbor disparity values.

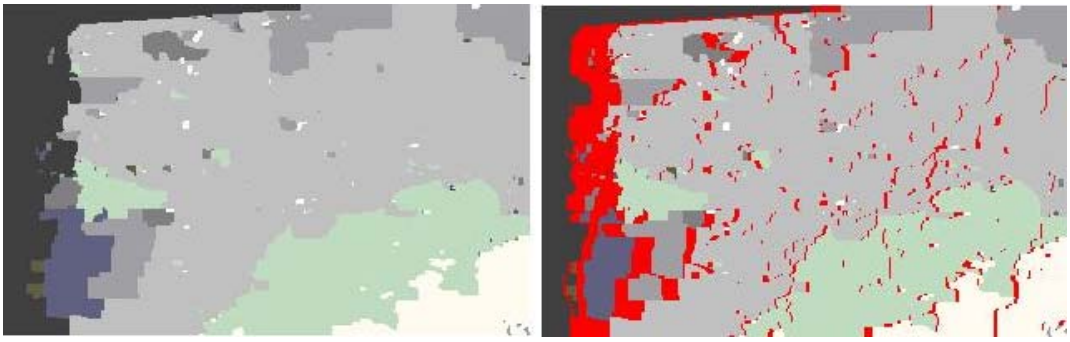


Figure 47 The disparity map of the coral sequence. The occluded regions are shown with red in the right image.

Figure 48 shows the 3D point cloud computed via dense reconstruction from different viewpoints.

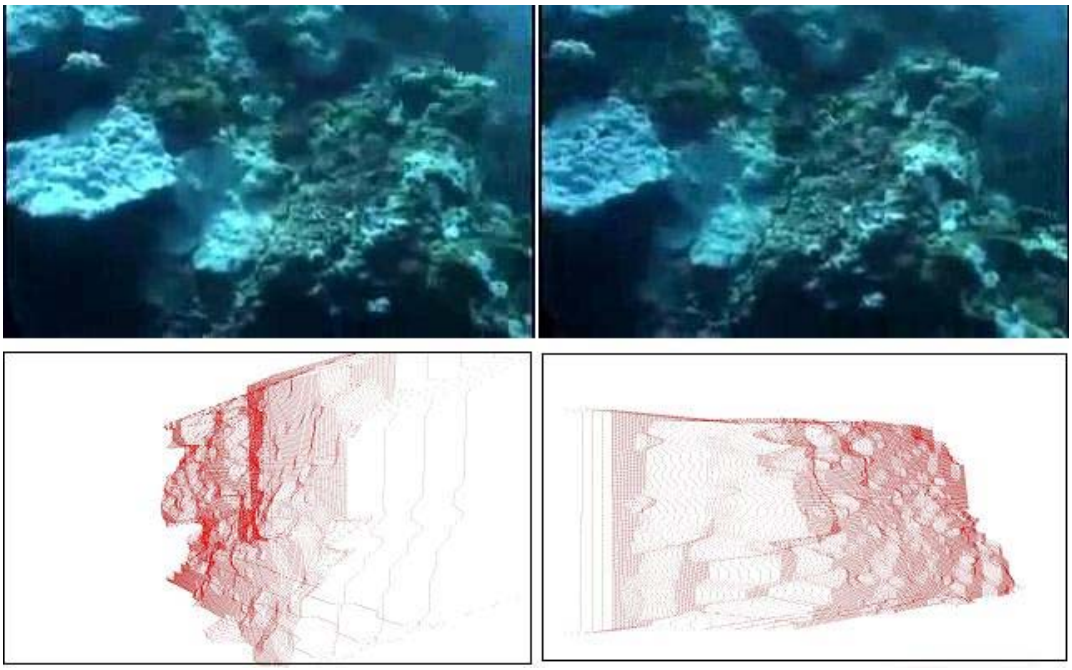


Figure 48 The two frames of coral sequence and reconstructed 3D point cloud.

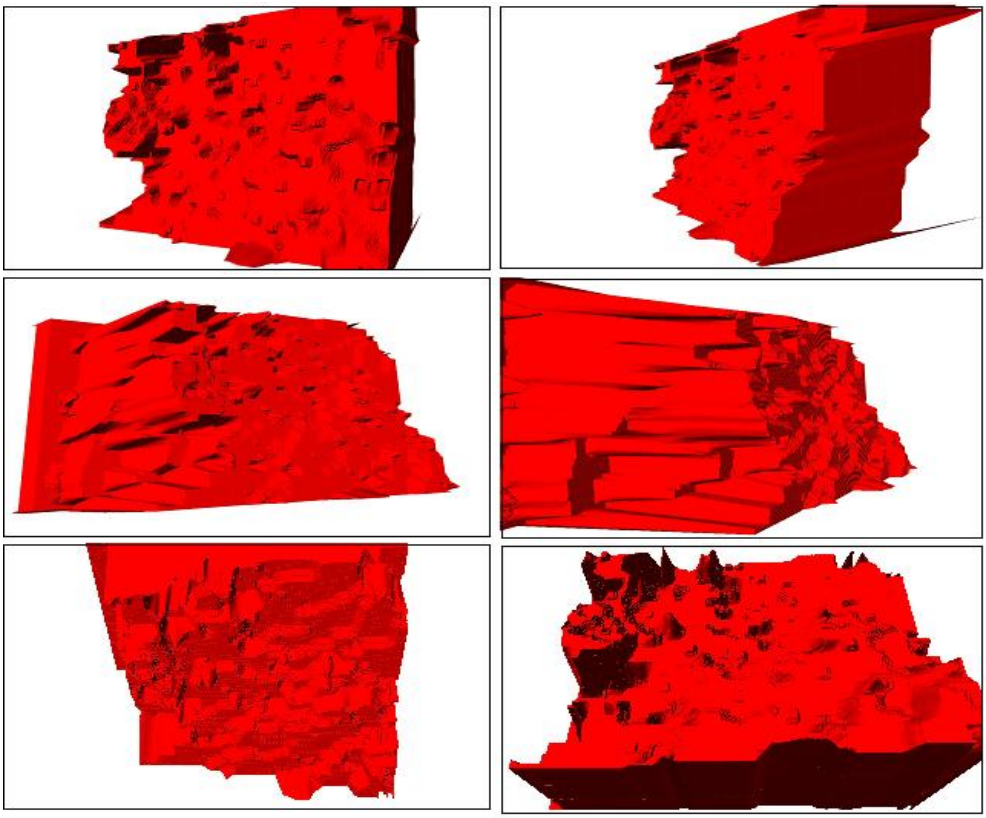


Figure 49 The generated mesh model of coral sequence.

A mesh surface is generated by 3D point cloud for a better visualization. Figure 49 shows the generated mesh and its view from different locations.

Coral 1 Sequence is the data set with the properties of underwater images captured in clear water with enough texture for the reconstruction. Figure 49 shows that the 3D reconstruction algorithm performs well in textured objects in clear water where the effect of water is little to distort the image.

11.3 Boat Sequence

Figure 50 shows the two chosen frames from a camera following a quasi-circular translational path around a submarine in blurred water.

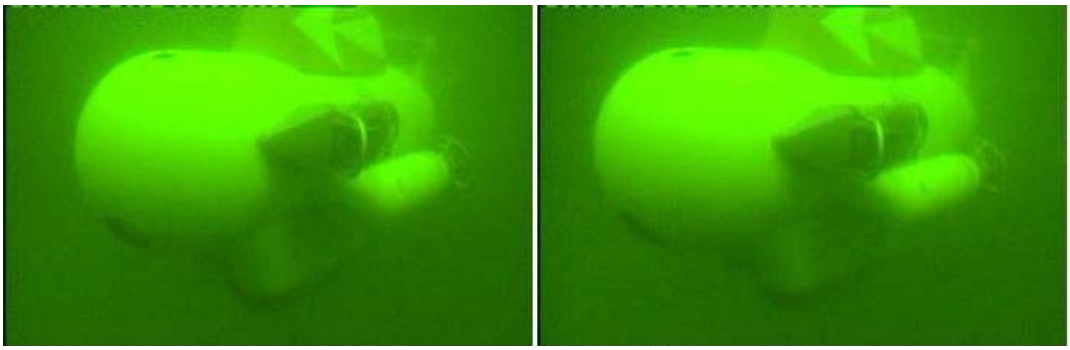


Figure 50 Chosen two frames from the boat sequence. The left image represents the left camera and the right image represents the right camera in the stereo pair.

Figure 51 shows the resultant images of preprocessing step. The blurring effect of the water is removed and it is easier to detect features.

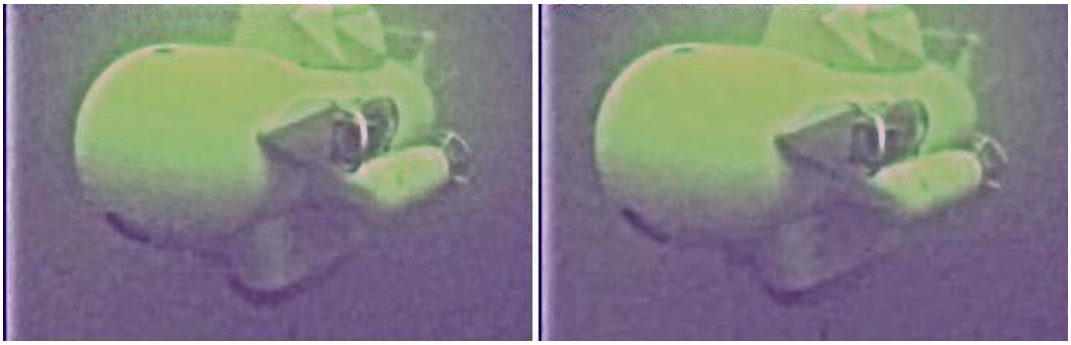


Figure 51 The two frames of boat sequence after preprocessing.

Detected and matched feature points and the epipolar lines are shown in Figure 52.

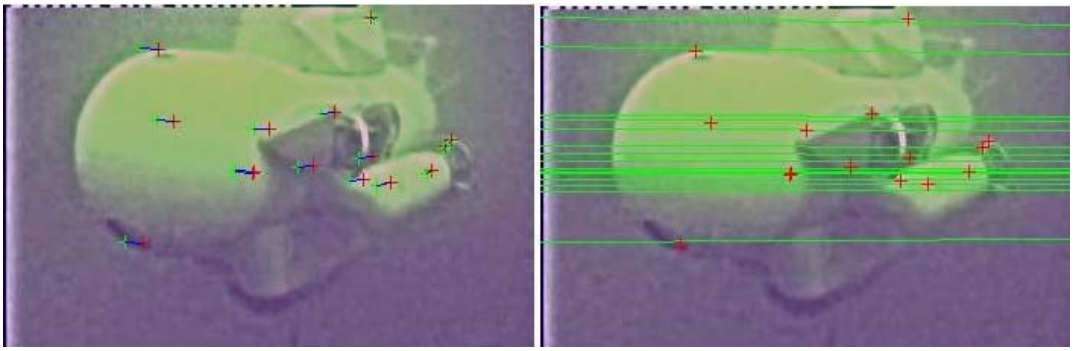


Figure 52 The corresponding points is shown in the left and the epipolar lines and feature points is shown in the right image.

The images rectified via uncalibrated rectification are shown in Figure 53.

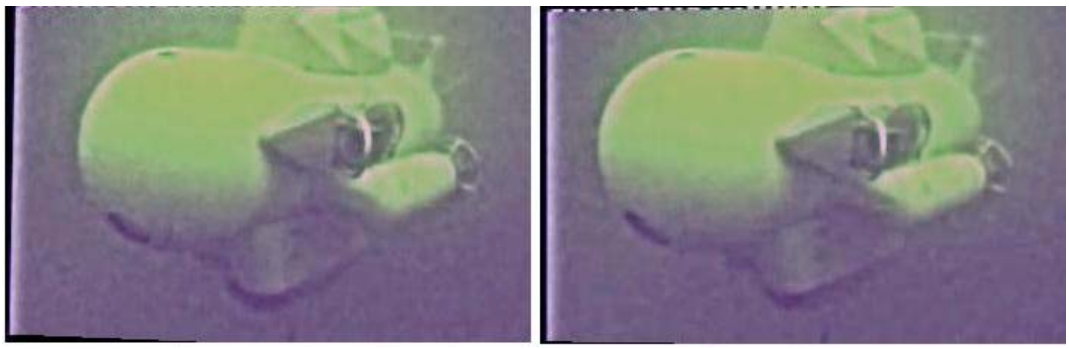


Figure 53 The boat sequence rectified via uncalibrated rectification algorithm.

Graph Cut based stereo matching algorithm is applied to the rectified images. the resultant disparity map and the occluded regions are shown in Figure 54. As it is seen from the Figure 54, the blurred water decreases the performance of stereo matching algorithm. The disparity map in textureless regions shows different depth levels, although the textureless region which represents the water at the back ground does not have any depth variation. But the blurred water makes it difficult to find the corresponding of each pixel in the image and causes false matches. Despite the disparity errors on the background, the shape of the submarine is preserved. But the little disparities differences are not detected. Since the submarine has a curved surface, but at the disparity map the surface of the submarine is presented as flat. The reason is that the errors in detecting the feature points, estimating the fundamental matrix and auto-calibration process, (lack of calibration matrix), cause a decrease in the accuracy of the dense reconstructed model.

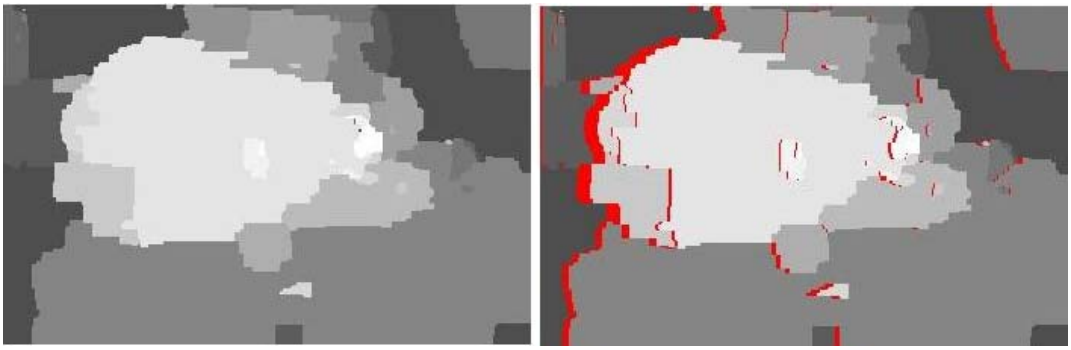


Figure 54 The disparity map of the boat sequence. The occluded regions are shown with red in the right image.

With the computed disparity map, the reconstructed 3D point cloud of the boat sequence is shown in Figure 55.

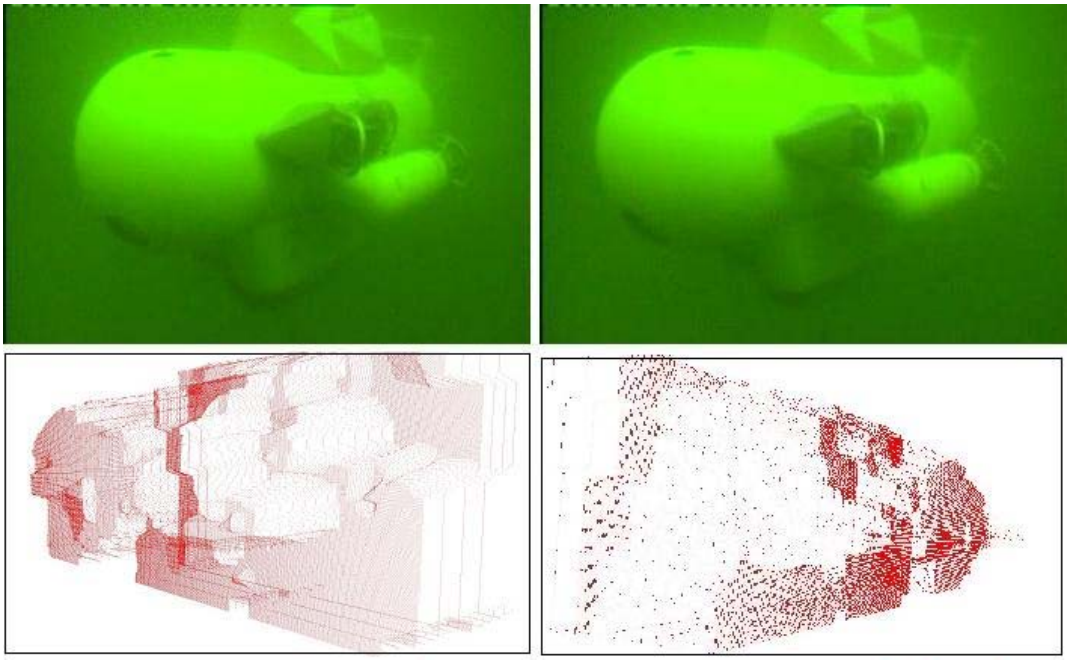


Figure 55 The two frames of coral sequence and the resultant reconstructed 3D point cloud.

The reconstructed point cloud is used to generate a mesh and covered with a surface for a better visualization.

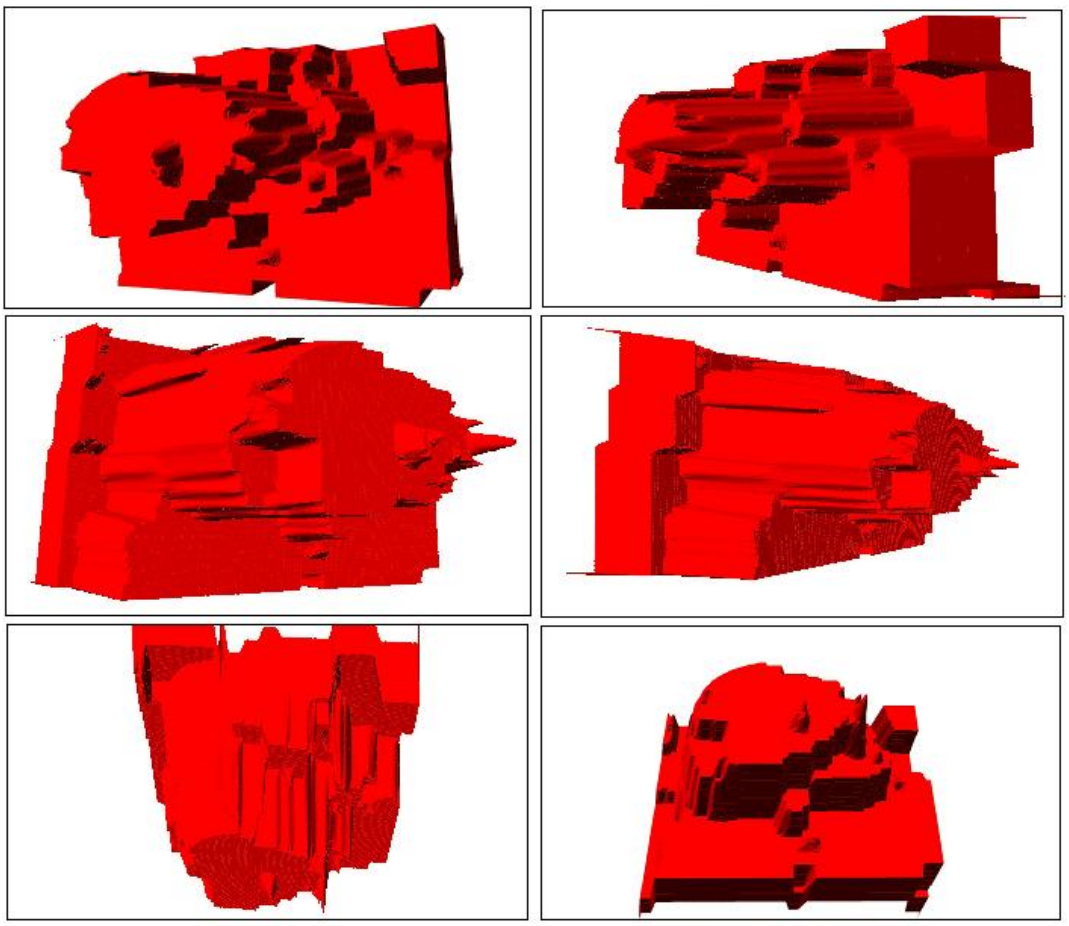


Figure 56 The generated mesh model of the boat sequence.

11.4 Coral 2 Sequence

Coral 2 Sequence is different from the previous two data sets in the way of camera motion. In the first two data sets the camera makes a translational motion with a little rotation around the observed object; but in Coral 2 sequence the camera makes a translational motion along the principal axis.

In the motion along principal axis, the features do not move along the horizontal axis, but along the vertical axis. The camera gets closer to the object between the successive frames. Since the object comes closer, the features' scale changes between frames which make them difficult to match. Since SIFT is a rotation and scale invariant feature detector, the scale difference between the features do not affect the performance of SIFT descriptor.

Figure 57 shows the two frames from Coral 2 sequence. The translational motion along the principal axis can be extracted from the positions of the coral reefs in the images. During the motion, the coral reef comes closer to the camera.

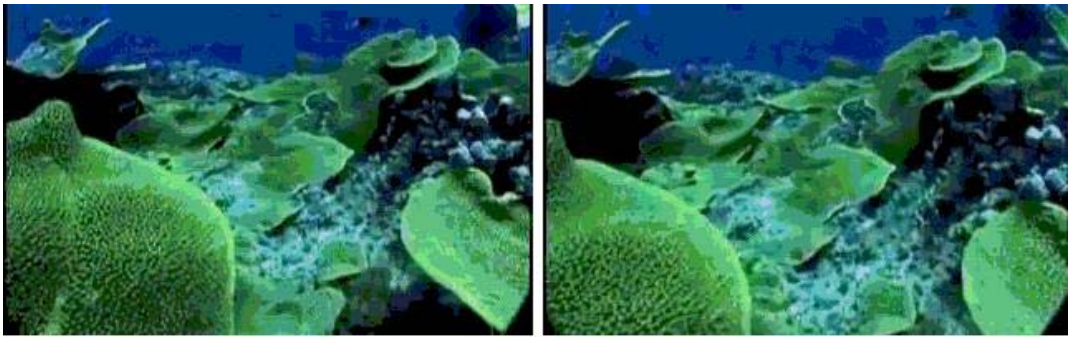


Figure 57 The two frames of Coral 2 sequence. The camera makes a translational motion along the principal axis.

The preprocessed images are shown in Figure 58.

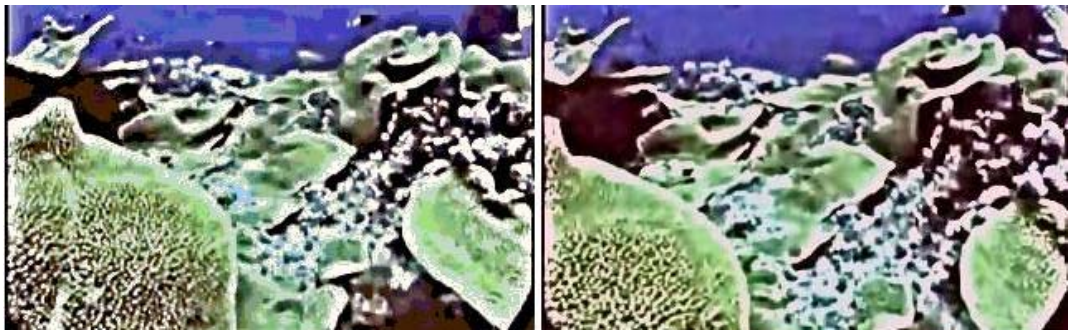


Figure 58 The two frames of Coral 2 sequence after preprocessing.

The motion along the principal axis causes the epipoles located in the images. the path of the feature points and the epipolar lines and the epipole of the right camera in the left image is shown in Figure 59.

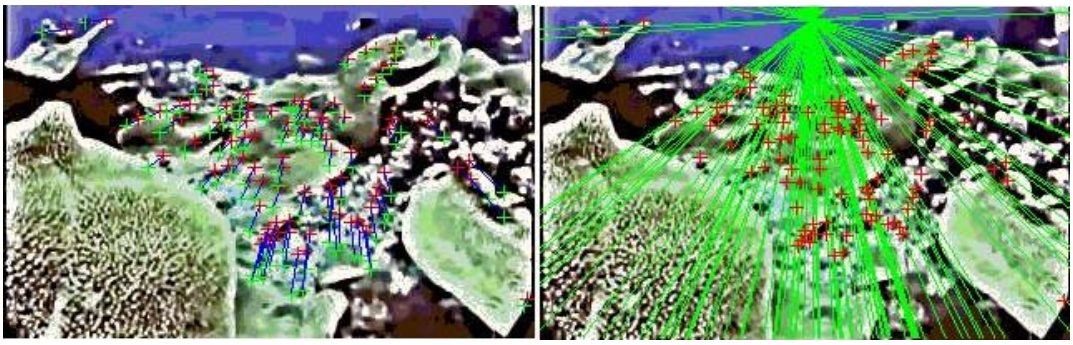


Figure 59 The corresponding points is shown in the left and the epipolar lines and feature points is shown in the right image.

The rectified images are shown in Figure 60. As it is seen, the uncalibrated rectification algorithm gives a more distorted image than the motion type in the previous data sets.

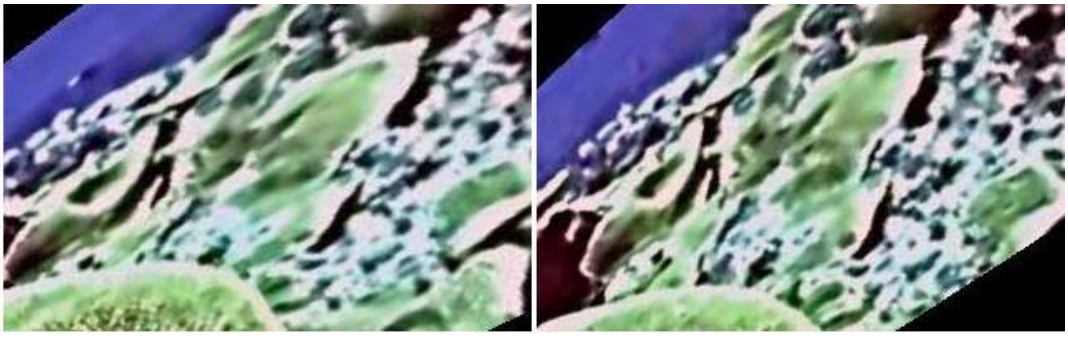


Figure 60 The rectified images via uncalibrated rectification algorithm.

The distorted rectified images also affect the performance of the stereo matching algorithm. The resultant disparity map is shown in Figure 61. The white spots in the disparity map show the false matches during the stereo matching. Although the region around the white spots is smooth and does not have any depth discontinuity, the stereo matching algorithm fails to find the correct correspondences and disparity values.

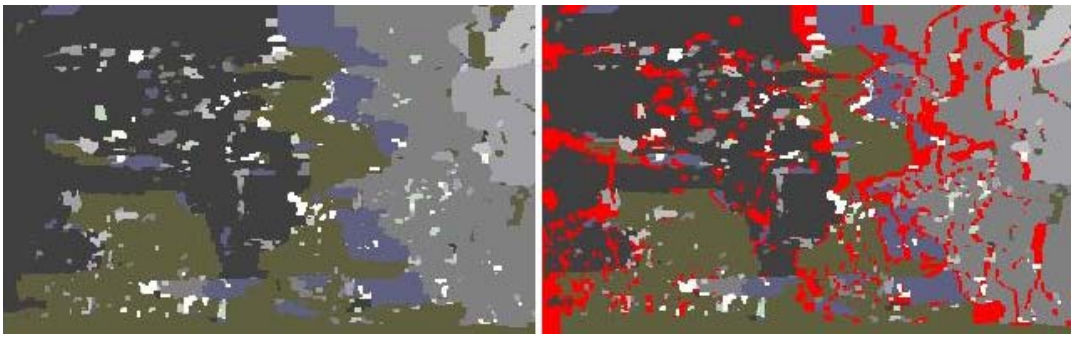


Figure 61 The disparity map of Coral 2 sequence.

Figure 62 shows the 3D point cloud reconstructed from the Coral 2 sequence.

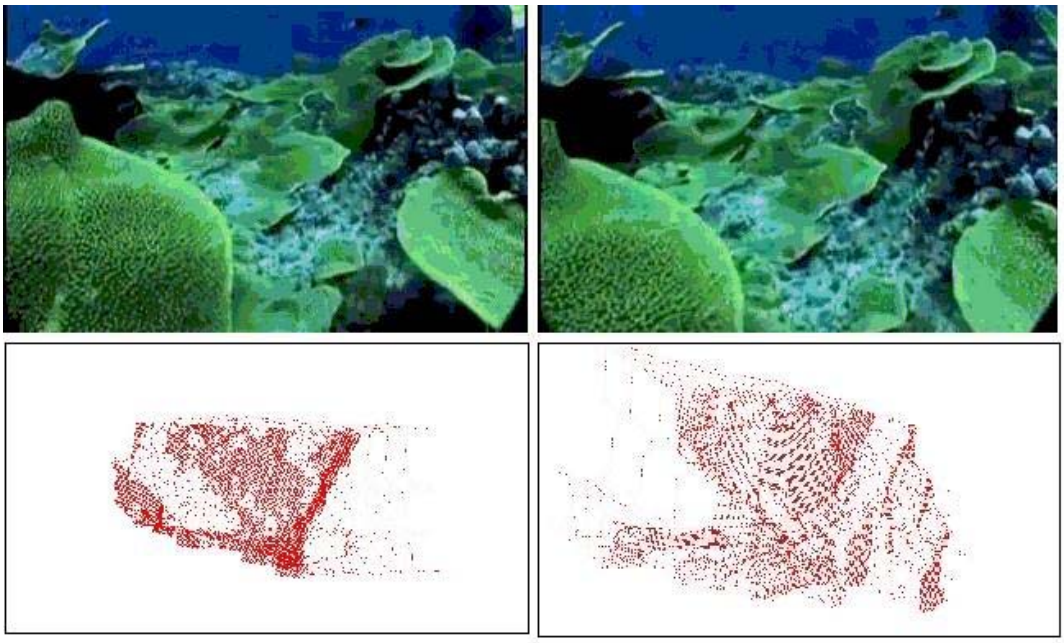


Figure 62 The two frames of Coral 2 sequence and the resultant reconstructed 3D point cloud.

Figure 63 shows the mesh model generated from 3D point cloud.

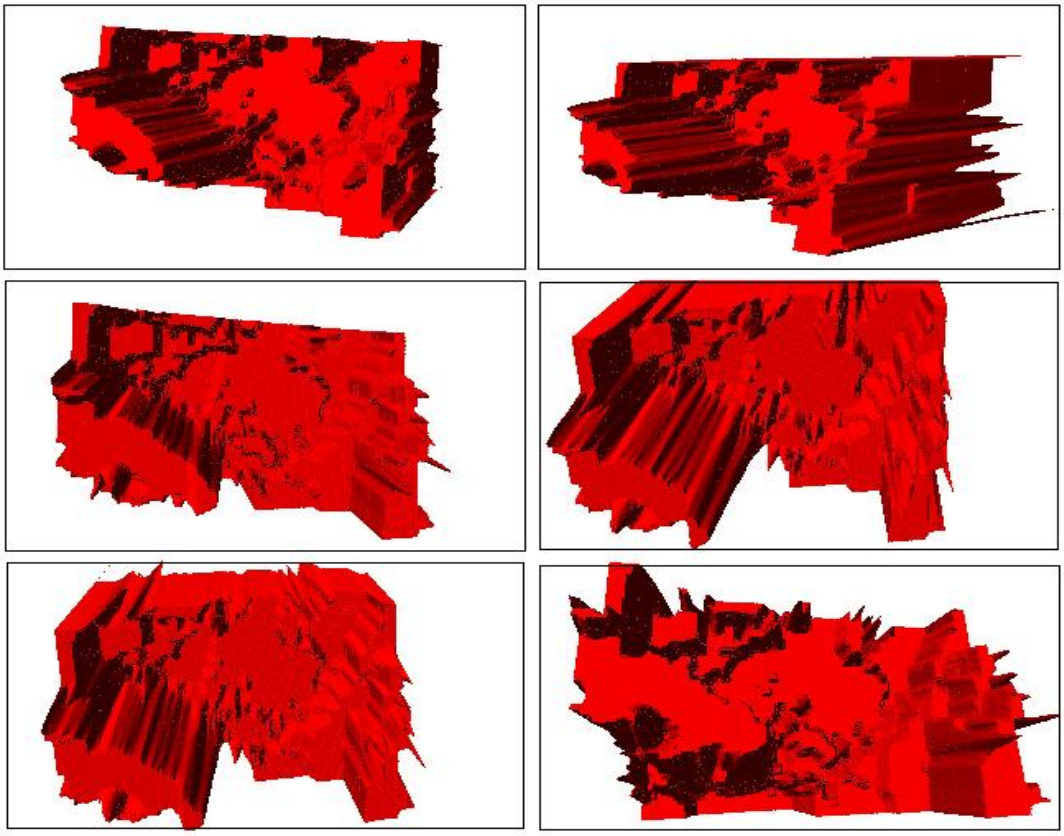


Figure 63 The mesh model of Coral 2 sequence.

Figure 63 shows that the 3D model suffers from accuracy. The positions of the corals are relatively true, but the shape of the corals is not preserved as well as in the previous data sets and the depth difference between the foreground and background is relatively small. The reason is that rectification algorithms fail when the camera moves along the principal axis. In the previous two data sets, the camera moves not along the principal axis but perpendicular to it and 3D reconstruction process give good results.

It is understood from the examined three data sets that 3D reconstruction process performs well when the camera makes a motion perpendicular to the principal axis and its performance falls when the camera moves along the principal axis.

11.5 METU Pool

The 3D reconstruction algorithm is also tested on PC-104 real time MPEG-4 video compressor, encoder and frame grabber module shown in Figure 64 with Helmet mountable underwater black & white video camera. The experiment is conducted in METU swimming pool.

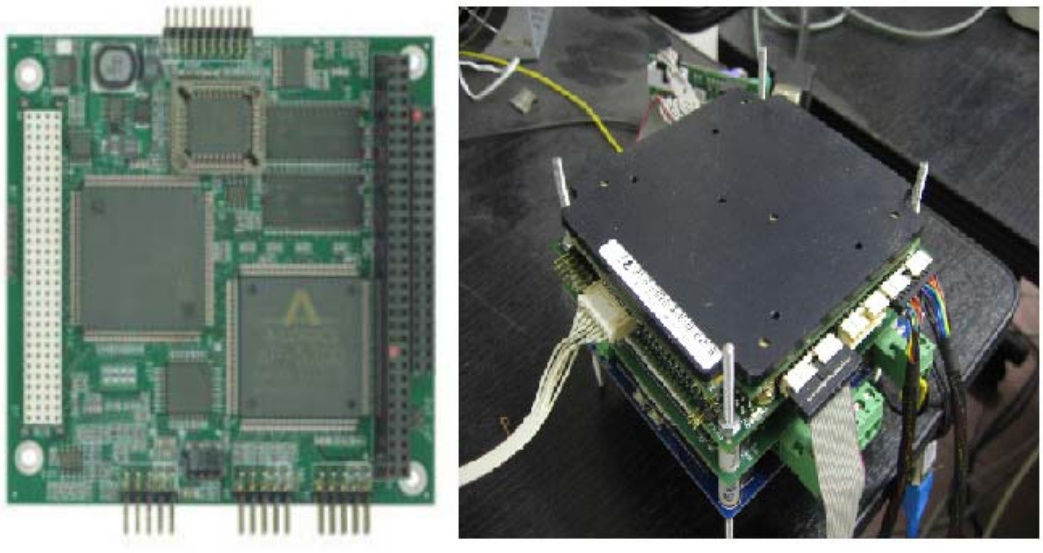


Figure 64 PC-104 is shown in the left part. The right image shows the PC-104 card used in this thesis.

Figure 65 shows the Helmet mountable underwater black & white video camera used in this thesis.

Figure 65 The underwater black & white video camera.

The underwater camera makes a translational motion around the submerged object, a statuette on a box. Figure 66 shows the two frames from the pool sequence captured in the METU Pool.

Figure 66 Two frames from pool sequence.

Since the water in the pool is clearer than the water in the sea, the images are not blurred and there is no need to preprocess the images. But once the images are preprocessed, a characteristic of the underwater camera used in this thesis is observed, moiré effect. Moire effect is a wavy repetitive pattern on the image and the importance of removing the moiré effect is that the following steps of preprocessing increase the contrast, also the moiré effect, and this increases the chance of degraded results. Figure 67 shows the images after preprocessing.

Figure 67 The two frames of pool sequence after preprocessing.

As it is seen from Figure 67, the preprocessing steps increase the moiré effect, although the moiré effect is removed from the raw images in the first step. The increased moiré effect decreases the image quality for feature detection algorithms. For that reason the preprocessing is not used in this sequence.

Figure 68 shows the detected feature and epipolar lines.

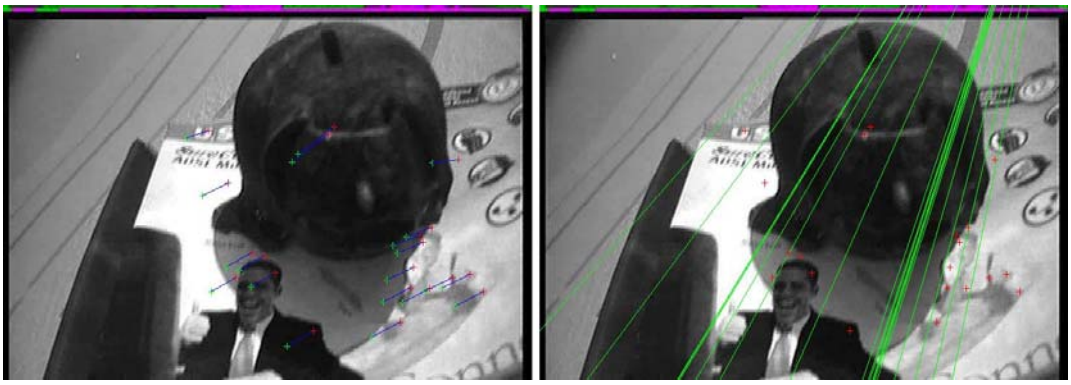


Figure 68 The detected feature and the epipolar lines.

The rectified images are shown in Figure 69.

Figure 69 The rectified images by uncalibrated rectification algorithm.

The next step is to compute the disparity map from rectified image pair. The disparity map is shown in Figure 70. The left part is the disparity map and the right part is the disparity map after the smoothing process.

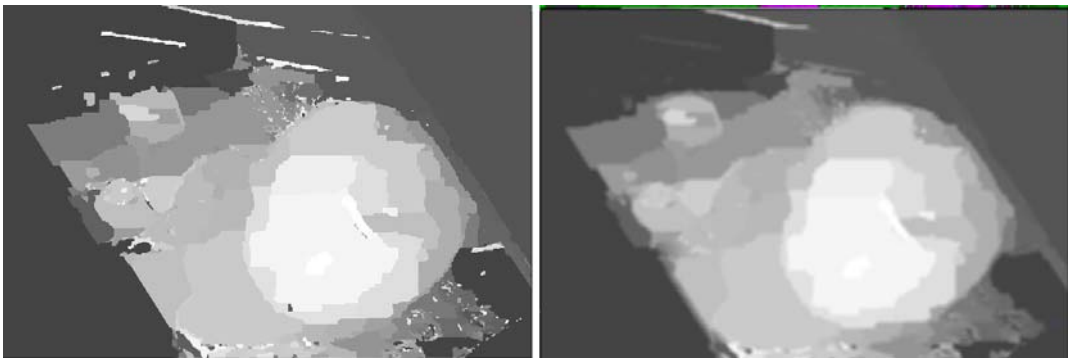


Figure 70 The left part shows the resultant disparity map from stereo matching algorithm and the right part is the result of smoothing process.

Figure 71 shows the computed 3D point cloud. The computed 3D point clod is covered with a surface for a better visualization. Figure 72 shows the 3D model covered with a surface from different view points.

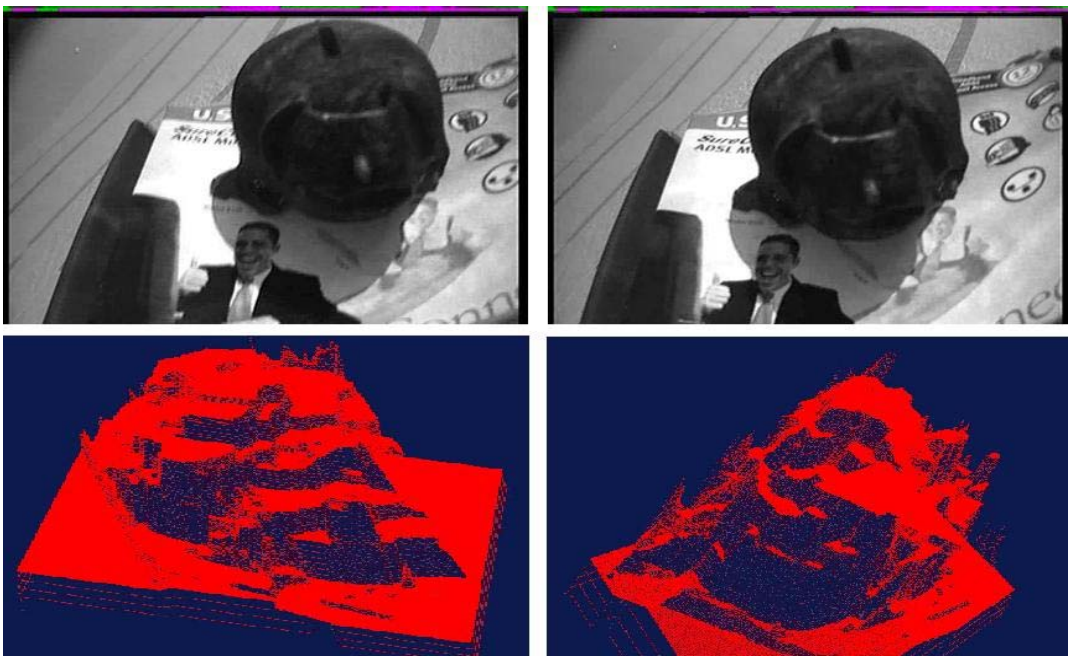


Figure 71 The two frames of pool sequence and the computed 3D point cloud.

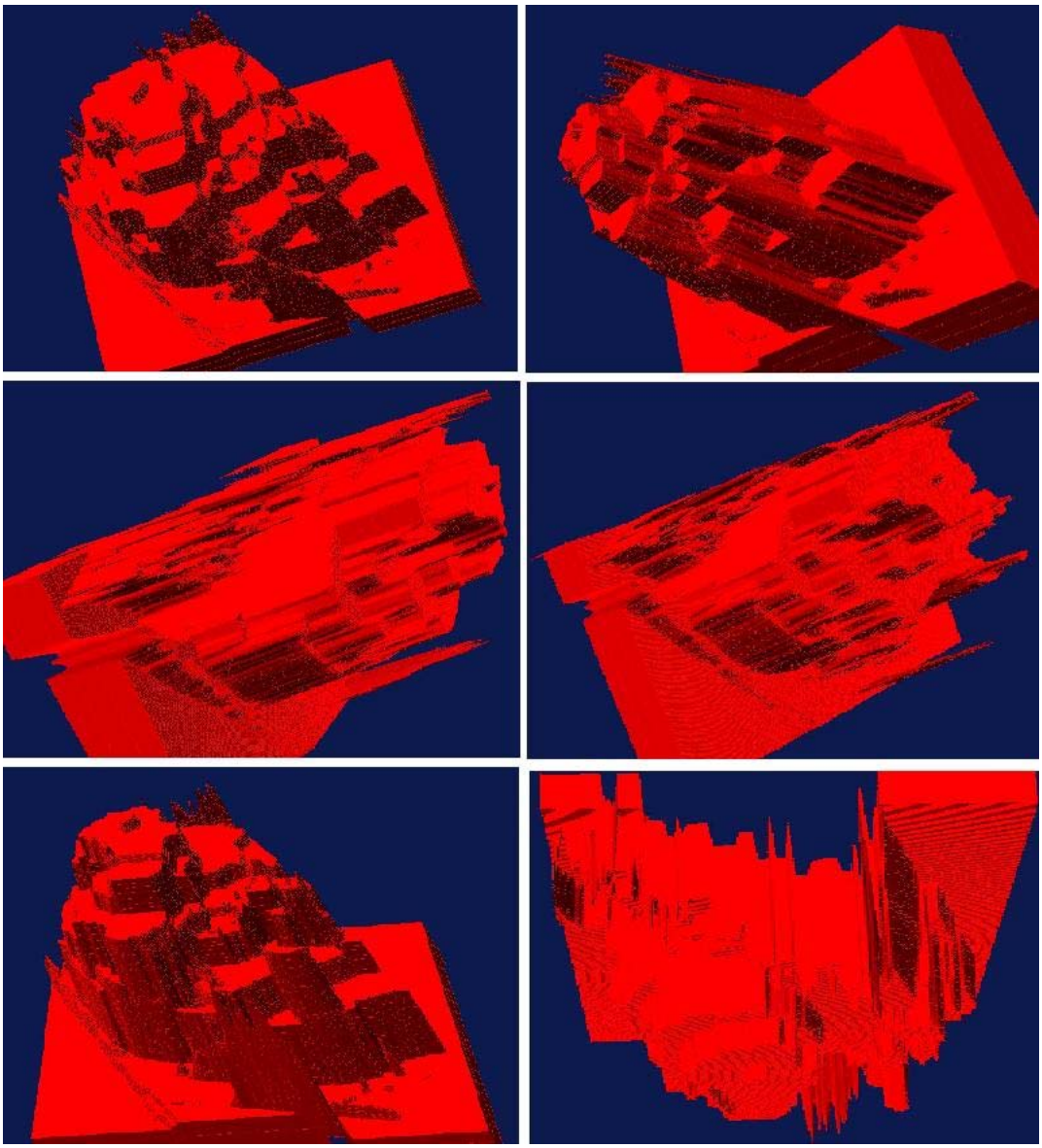


Figure 72 The 3D model of pool sequence form different viewpoints.

CHAPTER 12

CONCLUSION

12.1 Summary of the Thesis

The 3D reconstruction of underwater scenes is composed of successive sub-blocks.

The first step is the image enhancement process. Since the underwater is a complex structured environment and suffers from low contrast, non-uniform lighting, back-scattering, blurring etc…, the images from underwater has to be preprocessed before applying the image processing algorithms. A preprocessing filter is applied to the images which removes the effect of the water and enhance the images.

The second step is the feature detection and matching process. In order to find the corresponding points between the images, the feature points have to be found in each image. The most famous feature detection algorithm is Harris corner detector. A modified version of Harris corner detector and another feature detection algorithm, SIFT, are examined and compared.

After the corresponding points are determined, the next step is the computation of fundamental matrix which defines the epipolar geometry between the two images. The combination of normalized 8-point algorithm and RANSAC is used in estimating the fundamental matrix which is robust to noise and also provides outlier removal during the estimation process.

The forth step is the auto-calibration. Calibration matrix is one of the most important parameters in 3D reconstruction process. Since the underwater camera is uncalibrated, the calibration matrix is not known. Camera can be calibrated

with a known structured calibration object and also it can be calibrated without using any calibration object, using calibration patterns are impractical for underwater applications. Two different approaches to the auto-calibration problem are examined, auto-calibration using the virtual conic and simplified Kruppa equations.

The estimation of the rotation and translation matrix is performed with two different algorithms, linear and robust one. These two algorithms are examined and compared with ground truth data.

Rectification is the pre-process of stereo matching which is crucial for dense 3D reconstruction. Rectification provides new camera matrices with parallel and horizontal epipolar lines. This decreases the search area and improves the computation time in stereo matching. Two algorithms are examined, calibrated rectification and uncalibrated rectification.

Stereo matching is the process of finding the corresponding pixel of each pixel in the image. Graph based stereo matching algorithm is examined and compared with traditional stereo matching algorithm.

The last step is the triangulation. Because of the noise in the image, the back-projected rays from corresponding points do not intersect in 3D space. An optimal intersection point is computed by triangulation. Triangulation provides sparse 3D reconstruction. Two algorithms are examined, linear and polynomial triangulation. Finally a method for the dense reconstruction and the processes for a better 3D model are examined.

12.2 Discussion

In image enhancement step, the preprocessing filter removed the effects of the water and enhanced the images. The performance of the filter is tested with the detected features before and after the filtering and it is seen that the preprocessing filter significantly increases the number of detected feature. The more detected features, the more robust reconstruction are performed.

In feature detection and matching step, Harris corner detector and SIFT method are compared. SIFT method gives a better performance than Harris corner

detector. Despite the preprocessing filter, the blur in the image decreases the accuracy in corner detector. The invariance of SIFT to the rotation and scaling is the major reason why SIFT performs better in underwater. In feature matching the results show that SIFT descriptor performs better than normalized cross correlation (NCC).

The normalized 8-point algorithm and RANSAC are used in the estimation of fundamental matrix and outlier detection. This combination gives satisfactory results with relatively low Sampson errors.

Auto-calibration is one of the key steps in the reconstruction process. Its performance determines the performance of the reconstruction significantly. Simplified Kruppa equations and auto-calibration by virtual conic algorithms are compared. At least three images are required for the simplified Kruppa equations to determine three fundamental matrixes for the estimation of calibration matrix with five unknown parameters. Kruppa equations can not be solved in a straightforward manner. Instead, minimization algorithms are used to solve. Levenberg-Marquardt minimization algorithm, which is the most famous minimization algorithm in computer vision, is used in this thesis. An initial estimate is required to solve the minimization problem. The initial estimate directly affects the performance of the algorithm, as it can be observed form the results in Table 6. The more accurate initial estimate, the more good results the algorithm gives. The focal length in x-axis and y-axis are assumed to be equal and the principal point is assumed to be in the middle of the image with zero skew. The results in Table 6 show that this is a reasonable assumption. The latter algorithm, auto-calibration with virtual conic, fails to estimate the calibration matrix. The result of the algorithm is exactly equal to the calibration matrix used to normalize the projection matrix at the beginning of the algorithm. The reason is that the parameter ν which must converge during iteration, do not converge, so a reasonable dual absolute quadric can not be computed. The results show that simplified Kruppa equations performs better and gives reasonable calibration matrices.

Since rectification is the pre-step of stereo matching, its performance affects the performance of stereo matching. The goal of rectification is to form new extrinsic

parameters which provide parallel and horizontal epipolar lines with minimal distortion in the images. Two rectification algorithms are examined, calibrated rectification and uncalibrated rectification. The first mentioned algorithm requires calibration matrices with projection matrices. It computes a transformation which transfers the epipoles to the infinity. The latter algorithm approaches the rectification problem as a nonlinear least square problem with six unknowns and solves by minimizing the energy function given in Equation 8.15. The two algorithms are tested with coral sequence. It is seen that for a good result in calibrated rectification an accurate calibration matrix is required. The calibration matrix in the coral sequence is estimated by simplified Kruppa equations. As it is seen from Figure 29, the rectified images satisfy the rectification constraints with parallel and horizontal epipolar lines. But there is too much distortion in the images during the warping process. The uncalibrated rectification performs better than calibrated one and results with less distorted images. Since there is no common measurement to compare the performances of the rectification algorithm, the examined algorithms are compared with the mean differences between the y-axis coordinates of the corresponding points. Table 8 shows the results for calibrated and uncalibrated rectification algorithms. The results show that uncalibrated rectification gives a better performance than the calibrated one, which is expected, since the uncalibrated rectification minimizes this difference while computing the new camera matrices.

After the images are calibrated, the next step is stereo matching. Stereo matching is the process of finding the correspondence pixel of each pixel in the image and compute the disparity map of the scene, which can be assumed as the depth map of the scene. Several methods have been developed for stereo matching. Graph-cut algorithm creates more smooth disparity maps and performs better in textureless areas. For that reason graph cut is chosen in this thesis. It is compared with a traditional stereo matching algorithm using the Tsukuba sequence, which provides ground truth data for stereo matching algorithms, and coral sequence. Beside this comparison, the versions of graph cut algorithm, voxel labeling and pixel labeling, are compared. As it is seen from Figure 35, voxel labeling gives the best performance among pixel labeling and traditional stereo matching algorithm. The performance of stereo matching depends on the performance of the rectification and the texture characteristic of the scene. Stereo

matching algorithms do not perform well in homogeneous areas because of the difficulty of finding correspondence in this type of areas.

The final step is the computation of the 3D coordinates of the points whose projections are the corresponding points by triangulating the corresponding points. Two triangulation algorithms, linear and polynomial triangulation, are examined and compared. In Euclidean geometry, two algorithms perform similar. But in projective geometry polynomial triangulation gives better performance. When the 3D points computed with triangulation are projected on the images with the projection matrices of the camera for verification, it is seen that the re-projected points differs from the original ones with a difference of 3-7 pixels in y-axis. The reason of this difference is the error in the estimation of fundamental matrix.

Finding the 3D coordinates of the points with triangulation can be called sparse 3D reconstruction. Sparse means that not all of the points but only the corresponding points are triangulated. This means that only 20% of the points are triangulated. This provides a good 3D model, since the corresponding points are the most recognizable features in the image. But for a better and more detailed model, dense reconstruction must be performed. For dense reconstruction only the disparity map and the projection matrix of the camera is enough. The model is reconstructed form the view of the camera whose projection matrix is used. If the projection matrix of the left camera is used in the reconstruction process, the model is reconstructed according to the view of the left camera. It is same for the right camera. The reconstruction is performed via Equation 10.16. The computed 3D model represents the scene viewed by the two cameras. The model contains some errors, especially in the textureless regions, where stereo matching algorithm fails to find correspondence and also the errors during the estimation of projection matrix of the cameras.

12.3 Future Work

The 3D model is reconstructed from only 2 frames and the baseline distance between the frames are relatively small. All the frames in the video sequence may be used for the reconstruction and the 3d model of the scene from all viewpoints is achieved. Since underwater provides a limited visibility range, 25

meters in clear water and 3-5 meters in blurred water. It is impossible to cover the research site, for example an archeological site, with this visibility range. For that reason mosaicing algorithms are developed to combine the captured images in an appropriate order to get the image of the whole site. The 3D reconstruction process can be combined with mosaicing algorithm to reconstruct the 3D model of the whole site.

REFERENCES

[1] M.Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, “Visual modeling with a hand-held camera”, International Journal of Computer Vision 59(3), 207-232, 2004.

[2] C. Harris, M. Stephens, “A Combined Corner and Edge Detector”, In 4th Alvey Vision Conference, S. 147-151, 1988.

[3] D. G. Lowe, “Distinctive Image Feature from Scale-Invariant Keypoints”, International Journal of Computer Vision, 2004.

[4] Z. Zhang, R. Deriche, O. Faugeras, Q. T. Luong, “A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry”, INRIA, Report No. 2273, 1994.

[5] A. Noble, "Descriptions of Image Surfaces", PhD thesis, Department of Engineering Science, Oxford University 1989, p45.

[6] R. Hartley, A. Zisserman, “Multiple View Geometry in Computer Vision”, Cambridge University, 2004

[7] R. Hartley, “In Defense of the 8-Point Algorithm”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(2), 133-1337, 1997

[8] H. C. Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections”, Nature. 293:133-135, September 1981

[9] A. Fusiello, E. Trucco, A. Verri, “A compact algorithm for rectification of stereo pairs”, Machine Vision and Applications, 2000

[10] L. Irsana, A. Fusiello, “Quasi-Euclidean Uncalibrated Epipolar Rectification”, Research Report RR 43/2006, Dipartimento di Informatica – Universita di Verona, 2006

[11] D. Scharstein, R. Szeliski, “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms”, International Journal of Computer Vision, 47(1/2/3):7-42, 2002

[12] M. F. Tappen and W. T. Freeman, “Comparison of Graph Cuts with Belief Propagation for Stereo, using Identical MRF Parameters”, In Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV), Pages 900 - 907, 2003

[13] V. Brandou, A. G. Allais, M. Perrier, E. Malis, P. Rives, J. Sarrazin, P. M. Sarradin, “3D Reconstruction of Natural Underwater Scenes Using the Stereovision System IRIS”. Oceans 2007 Europe, 2007

[14] V. Kolmogorov and R. Zabih, “Graph Cut Algorithms for Binocular Stereo with Occlusions”, In Mathematical Models in Computer Vision: The Handbook, Springer-Verlag, 2005

[15] J. Sun, H. Y. Shum, N.-N. Yeng, “Stereo matching using belief propagation”, IEEE Trans. Pattern Anal. Mach. Intell. 25 (7) (July 2003).

[16] V. Kolmogorov, “Graph Based Algorithms for Scene Reconstruction from Two or More Views”, Ph.D. Thesis, Cornell University, 2003

[17] V. Kolmogorov and R. Zabih, “Multi-camera Scene Reconstruction via Graph Cuts”, In European Conference of Computer Vision (ECCV), 2002

[18] Y. Boykov, O. Veksler, R. Zabih, “Fast Approximate Energy Minimization via Graph Cuts”, In Internation Conference of Computer Vision, pages 377-384, 1999

[19] U. Topay, “3D Scene Reconstruction from Uncalibrated Images”, Ms. Thesis, METU, 2002

[20] S. Bazielle, I. Quidu, L. Jaulin, J. P. Malkasse, “Automatic Underwater Image Pre-Processing”, CMM, 2006

[21] M. I. A. Lourakis and R. Deriche, “Camera self-calibration using the Kruppa equations and the SVD of the fundamental matrix: The case of varying intrinsic parameters”, Research Report, INRIA, 2000

[22] Hartley Richard, “Kruppa’s Equations Derived from the Fundamental Matrix”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19, 2, 1997

[23] R. Hartley, P. Strum, “Triangulation”, Computer Vision and Image Understanding, Vol. 68, No. 2, pp. 146-157, 1994

[24] M. Pollefeys, “Visual 3D Modeling from Images”, Tutorial Notes, University of North Caroline, 2004

[25] V. Kolmogorov and R. Zabih,"Computing Visual Correspondence with Occlusions using Graph Cuts", International Conference on Computer Vision, 2001.

[26] Y. Boykov, O. Veksler, R. Zabih, "Markov Random Fields with Efficient Approximations", IEEE Computer Vision and Pattern Recognition Conference, 1998.

[27] A. Fusiello, V. Murino, “Augmented Scene Modeling and Visualization by Optical and Acoustic Sensor Integration”, IEEE Transactions on Visualization and Computer Graphics, 2004

[28] H. Singh, C. Roman, L. Whitcomb, D. Yoerger, “Advances in Fusion of High Resolution Underwater Optical and Acoustic Data”, IEEE, 2000

[29] Y. Y. Schechner, N. Karpel, “Recovery of Underwater Visibility and Structure by Polarization Analysis”, IEEE Journal of Oceanic Engineering, 2005

[30] O. Faugeras, Q. T. Luong, S. J. Maybank, “Camera Self-Calibration: Theory and Experiments”, Proceedings of the 2nd European Conference on Computer Vision, 321-334, 1992

[31] R. Hartley, “Kruppa’s Equations Derived from the Fundamental Matrix”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997

[32] M. A. Fischler, R.C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, Communications of th ACM, Volume 24 Number 6, 1981

[33] O. Pizarro, R. Eustice, H. Singh, “Large area 3D reconstruction from underwater surveys”, Oceans’04 Vol. 2, pages 678-687, 2004

[34] K. Plakas, E. Trucco, A. Fusiello, “Uncalibrated Vision for 3D Underwater Application”, Oceans'98 IEEE/OES Conference, 1998