Structure from Motion Methods for 3D Modeling of the Ear Canal

Florin S. Telcean

Kongens Lyngby 2007

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Summary

Structure from Motion deals with 3D reconstruction from 2D images and it is one of the most widely researched problems within computer vision area. Recently, it was successfully integrated in many medical oriented applications.

Reconstruction of accurate models of the ear canal is a key step in the design and production of hearing aids. Actual methods are based on an invasive procedure (ear canal impression taking), require time, special trained skills and sophisticated and expensive hardware as 3D laser scanners. On the other side, the video otoscope became a standard tool in the hearing specialist office and it is able to provide images of the ear canal.

This thesis is about 3D reconstruction of the human ear canal from images using Structure from Motion methods. Two aspects are studied. First, the images of the ear canal are analyzed in order to see if they provide enough information for reconstruction algorithms. Second, the reconstruction accuracy of tube-like objects is analyzed in the context of a specific Structure from Motion algorithm.

Resumé

Structure from Motion handler om 3D rekonstruktion af 2D billeder og er én af de mest undersøgte problemerstillinger indenfor computervisionen. For nylig, blev den integreret med succes i mange medicinske metoder.

Rekonstruktion af præcise modeller af hørekanalen er nøglen i design og produktion af høreapparatter. Nuværende metoder er basserede på invasive procedurer (udformelse af hørekanalen) og de tager tid og har brug for special trænede egenskaber, sofistikerede og dyre hardware såsom 3d laser scannere. På den anden side, video otoskopien blev et standard redskab hos hørelægerne og er i stand til at gendanne billeder af hørekanalen.

Dette projekt handler om 3D rekonstruktion af den humane hørekanal ved hjælp af billeder ved brug af Structure from Motion metoder. Man undersøger to aspekter. Til at starte med, analyserer man billederne af hørekanalen for at samle nok informationer for rekonstruktions algorytmerne. Derefter rekonstruktions nøjagtigheden af cylinderagtige objekter er analyseret i sammenhæng med et specifik Structure from Motion algorytm.

Preface

This thesis has been prepared at Informatics and Mathematical Modeling (IMM) department of the Technical University of Denmark (DTU), and it is a requirement for acquiring the Master of Science degree in engineering.

The extent of the project work is equivalent to 30 ECTS credits, and was carried out between 21st of August 2006 and 31st of January 2007. The work leading to this thesis was supervised by Bjarne Kjær Ersbøll and co-supervised by Henrik Aanæs.

The purpose of this thesis is to study the feasibility of Features Based Structure from Motion methods for 3D reconstruction of human ear canal.

Kgs. Lyngby, January 31, 2007

Florin S. Telcean – s041386

Acknowledgments

I would like to thank my supervisors Bjarne Kjær Ersbøll and Henrik Aanæs for their good advices and great support in structuring my work. A special thank to Regin Kopp Petersen for technical support. Finally, I would like to thank my friends for their understanding in this stressful period, and also for their encouragements to carry this work to the end.

Contents

CHAPTER 1

Introduction

Recovering the 3D structure of a scene together with the camera motion from a sequence of images is known as the Structure from Motion (SFM) problem and challenged researchers over the last two decades. If in the past the most important applications were in visual inspection and robot guidance, in recent years an increasingly interest has shown for visualization. Creating accurate models of existing scenes has now applications in virtual reality, video conferencing, manufacturing and medical visualization, to mention only a few.

Some of the current solutions designed to extract 3D information of the objects or scenes are often based on expensive specialized hardware like laser scanners. The recent developments in computer hardware and digital imaging devices, as well as the requirement of robust and low cost systems encouraged the development of image-based approaches. Many of new developed methods can produce 3D models of real scenes with just a simple consumer camera and a computer processing images acquired with the camera (e.g. [1, 2]).

Structure from motion it is not a single, well defined problem. It covers a range of problems related to different imaging scenarios, camera motion, and models of the scene [4]. The complexity of SFM is also reflected in the extensive research in the area over such a long period of time. Even if many aspects related to SFM reached a kind of maturity, SFM is still subject of further research. There is not a generally applicable SFM algorithm capable to recover

the 3D information from any kind of real scenes and under any conditions. Current SFM algorithms may perform well on a certain type of scenes and under very well defined conditions, but when these conditions change they fail. This is a very strong argument to design specific SFM algorithms for specific problems [5].

In the last years Structure from Motion methods have proved their applicability in medical area. The endoscopic camera became a popular and powerful tool in minimum invasive surgery, providing the possibility to visualize the internal organs and structures for diagnosis. Structure and motion estimation techniques were successfully applied on images provided by video-endoscope systems [6-16]. 3D reconstruction from CT or MRI data is well known in virtual endoscopy systems, but it provides only 3D shape visualization without real textures [3]. Structure from motion was successfully used in mapping 2D images provided by the endoscope to volume data (e.g. [13, 14]), thus contributing to the construction of very accurate textured 3D models of the inner structures of human body. Some success was also achieved in recovering the 3D structure of different organs or parts of them only from endoscopic images [3, 6, 8, 9]. In [8] the 3D model of the operating field is obtained using a stereoscopic endoscope. Other applications in the minimal invasive surgery are endoscope tracking [7, 12], or the 3D modeling of deformable surfaces [9, 10]. These results are encouraging and probably in the near feature SFM will be used in many other medical applications.

At the time of writing of this thesis, to the best of my knowledge, there was not any known research related to the 3D modeling of the ear canal from endoscopic image sequences.

Building accurate models of the ear canal has a direct application in the hearing aid industry. The miniaturization of hearing aids allows them to be placed directly in the ear canal. Thus they are able to provide better acoustic performance and in the same time they are cosmetically appealing. If until recently the manufacturing of hearing aids was a completely manual task, the actual trend is to automatize the production process. Of course, this requirement implies the construction of a digital model of the ear canal. Currently, this model is obtained by scanning an impression of the ear canal with laser scanners. Taking an impression of the ear canal is a task done completely manually, requires time and special trained skills of the operator. The 3D modeling step is based on expensive and specialized scanning equipment. The invasive nature of the impression taking process is probably one of the most negative aspects of this procedure, and can be a very unpleasant experience for the patient. There is also a risk of producing injuries of the ear canal, or worse of the ear drum, if the procedure is not performed properly. All these aspects

suggest that, if possible, a better solution for modeling the ear canal should be found. Recovering the 3D model using the endoscopic image sequences may be a good candidate to possible solutions as it is less invasive, faster, cheaper, and does not require special trained skills.

The purpose of this thesis is not to provide a full SFM solution for the given problem, but rather to study the applicability of SFM methods to the 3D reconstruction of the ear canal.

1.1 Thesis overview

Chapter 2 is a short introduction to the techniques used in present for 3D modeling of the ear canal, emphasizing the reasons of writing this thesis.

Chapter 3 gives the theoretical fundaments for feature based structure and motion estimation from images.

Chapter 4 is an overview of different feature detection and tracking methods, and also presents the results of experiments performed with otoscopic images.

Chapter 5 deals with the reconstruction accuracy of tube-like objects. Several experiments with synthetic and real data are performed and analyzed.

Chapter 6 presents the conclusions of this work.

1.2 Nomenclature

BTE	Behind The Ear hearing aid
CIC	Completely In the Ear hearing aid
CS	Coordinate System
EBR	Edge Based Region
IBR	Intensity Based Region
ICP	Iterative Closest Point
ITC	In The Canal hearing aid
ITE	In The Ear hearing aid
KLT	Kanade-Lucas-Tomasi tracking method
MSER	Maximally Stable Extremal Region detector
NCC	Normalized Cross-Correlation
PC	Principal Components

PCA	Principal Components Analysis
RANSAC	Random Sampling Consensus
SFM	Structure from Motion
SIFT	Scale Invariant Feature Transform
SSD	Sum of Square Differences
SURF	Speeded Up Robust Features
SVD	Singular Value Decomposition
TPS	Thin Plate Spline
VO	Video Otoscope / Video Otoscopy

CHAPTER 2

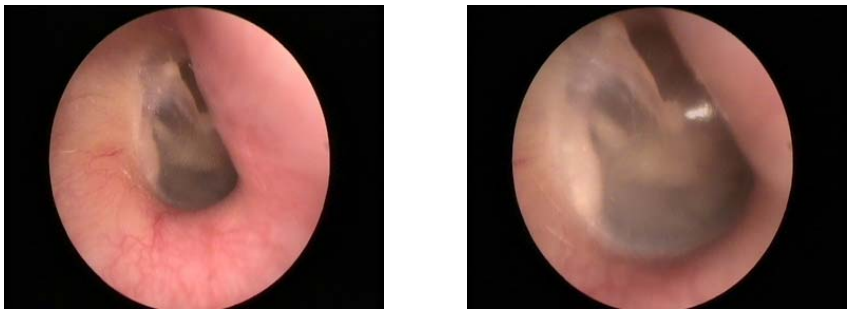
Background

2.1 Ear Canal Anatomy

The external ear consists of the auricle or pinna, ear canal (also called external auditory canal) and the outer surface of the eardrum (or tympanic membrane). The pinna is the outside portion of the ear and it is normally referred as ear. Pinna is made of skin-covered cartilage.

Figure 2.1 The anatomy of the external ear

The ear canal extends from the pinna to the ear drum and it has an oblong S-shape. It is a small, tunnel like tube, about 26mm long and 7mm in diameter. Size and shape of the canal vary among individuals.



a) *The inner ear canal* b) *The ear drum*

Figure 2.2 Otoscopic images of the ear canal

The eardrum (outer layer of the tympanic membrane) is located at the inside end of the ear canal where it separates the external canal from the middle ear. The eardrum has a slightly circular shape.

The outer 2/3rds of the ear canal is surrounded by cartilage, has thick skin, numerous hairs and contains glands that produce cerumen (ear wax).

The inner portion of the ear canal (aprox. 1/3rd) is narrower and surrounded by bone. This part is covered by very thin and hairless skin. The skin in this section is very sensitive to touch, and it can be easily injured. Due to obliquity of tympanic membrane inferior wall of the inner canal is about 5 mm longer than superior wall.

The size and shape of the ear canal (subject of change for example when a person is speaking or chewing) are important factors to consider in the hearing aids manufacturing.

2.2 Hearing aids

The hearing aid is an instrument that amplifies the sounds for people with hearing problems. As technology evolves the hearing aids become more advanced and highly sophisticated devices. If in the past the hearing aids were analogical devices, today digital aids are programmable to fit the specific acoustic needs of each user. The miniaturization of hearing aid components it still an area of research and experiments but it already makes possible the construction of hearing aids small enough to be placed completely in the ear

canal. This type of hearing aids offer many advantages for the user comparing with the more traditional ones we normally can see behind the ear of wearers.

Even if the hearing aids come in different forms, basically all of them contain the same main elements:

- a microphone to capture the sounds,
- an electronic amplifier to amplify the signal provided by microphone,
- an earphone or receiver (speaker),
- an ear mold or plastic shell that transfers the amplified sound from the earphone to the eardrum (directly or through plastic tubes),
- a power source / battery.

There are four types of hearing aids:

- Behind the ear (BTE) hearing aid: the case housing the electronics is fixed behind the ear. An earmold is fixed in the canal and the sound is directed through a tube. They are the largest hearing aids available, can provide higher amplification of the sound, and can house larger batteries.
- In the ear (ITE) hearing aids fill the outer ear.
- Completely in the canal (CIC) are the smallest hearing aids available and are customized for the wearer's ear. They are placed deep inside the ear canal, in this way resembling a natural reception of the sound, since the microphone and the speaker are both in the canal. Being barely visible from exterior, this type of hearing aids is cosmetically appealing for the wearer.
- In the canal (ITC) hearing aid are just a little bit larger than the CIC ones, but can house a larger battery.

2.3 Hearing aids production

Until recently, the production of a CIC for a given ear was completely a manual and difficult task and the quality of the finished instrument was dependent on the skill of the operator. As the hearing aids are made individually for each patient, it is very important to have the possibility to build hearing aid shells and earmolds that fit properly in the ear.

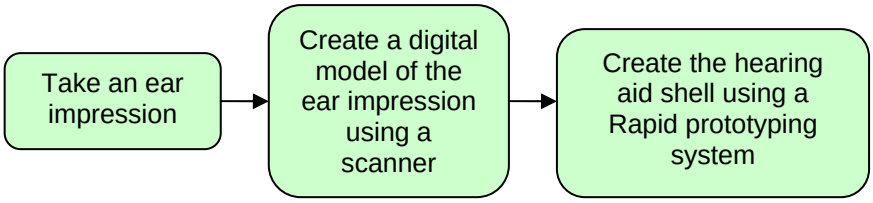


Figure 2.3 Main steps of hearing aids shells manufacturing

A hearing aid that is not properly fitted in the canal cannot ensure a good functionality of the device, and it is also uncomfortable for the wearer [18].

The traditional manual processing technique can not offer high accuracy and it is a long time process. On a production basis, accuracy and timing are very important factors. These are good reasons to eliminate as much as possible the human intervention from the production line. Thus, the production of hearing aids shells is today much more automatized, even if it's still dependent on human actions. As showed in *Figure 2.3*, three main steps are required to be completed in order to build a custom hearing aid shell or earmold:

1. Take an impression of the ear;
2. Create a digital model of the ear impression using a 3D scanner;
3. Create the physical shell or earmold reproducing the digital model using a rapid prototyping system (a kind of 3D printer).

Only one out of the three steps requires extensive human intervention, namely the ear impression taking process. In the followings these three steps are discussed and detailed.

2.4 Ear impressions

In order to create a custom hearing aid or earmold, a replica of the ear called ear impression has to be created. Techniques available today allow hearing professionals to make the ear impressions in the office. An ear impression is made by injecting a soft silicone material into the ear canal and outer portion of the ear. In order to protect the ear drum, a dam made from special cotton or foam material is placed in the ear canal. The impression material is inserted using a syringe or a silicon gun. The "gun" has two separated containers, one for the silicon material and one for a stabilizer, and these two are mixed on injection.

Figure 2.4 Example of ear impression

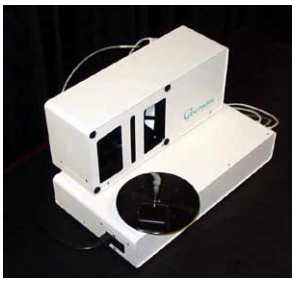
Depending on the type of material used, after 5-15 minutes the mix hardens and thus it provides a detailed replica of the ear. This is then removed by the specialist along with the protection dam. The ear impressions obtained in this way, individually from each patient's ear, are used to build very precisely the shells of the hearing aids or earmolds. The execution precision of the hearing aid shell or earmold is very important since the comfort of the patient depends on it.

Considerable professional skill and care must be exercised in selecting the size, material and placement of the protection dam within the external ear canal [68]. The material compressibility of the dam should be also related to the density of the silicone material used to take the impression.

Impression taking is an invasive procedure for the patient since a foreign object is introduced in the ear canal and then extracted. There is always the risk of producing some medical problems when taking an ear impression, varying from minor patient discomfort to some slight trauma of the ear. The incidence of significant trauma to the external or middle ear seems to be low anyway [17]. It is also showed in [68] that the material mix consistency and injection force have a profound otic impact in the case of improper ear impression-taking technique. Particular risks present patients with a damaged ear drum or with a previous surgery.

2.5 Ear Impression 3D Scanners

3D scanners are used to create a 3D digital model of an ear impression. Of course they are not dedicated to scan only ear impressions; they can also be used to obtain 3D models of other small objects.



Cyberware's Model 7G 3D scanner

3Shape S-200 3D scanner

Figure 2.5 Ear Impression 3D Scanners

There are many producers offering 3d scanners and most of them are based on laser technology. The laser beams are used to determine the depth of points on the surface of the scanned object. Two models of 3D scanners based on laser technology are presented in the *Figure 2.5*. Other 3D scanners use a structured light pattern projected onto the surface of the object in order to recreate its 3D model.

The object is placed on a rotating support and multiple scans are performed from different viewing angles. From these views a software application creates completely assembled digital 3D models.

The 3D scanners are small and compact enough to easily fit on the desk. They are able to acquire accurate and highly detailed 3D models of ear impressions in just a few minutes. For example, S-200 scanner model from 3Shape is able to scan up to app. 200,000 points, and the final 3D model contains app. 25,000 triangles.

Even if the 3D scanners are in general expensive pieces of equipment, some integrated low-cost packages can be also found on the market.

2.6 Rapid prototyping systems

Rapid prototyping is a generic name given to a class of technologies used to produce physical objects from a digital model [69]. These technologies are also known under different names like three dimensional printing, solid freeform

fabrication, additive fabrication or layered manufacturing (in order to form a physical object the materials are added and bounded layer by layer).

Rapid prototyping is a completely automated process. The digital model is transformed into cross sections, and then each cross section is physically recreated. Different technologies have advantages and weaknesses related to the processing speed, accuracy of reproduction, materials that can be used, surface finish, size of the object, and system price.

One of the most widely used rapid prototyping technology is *stereolithography*. With this technology the objects or parts of them can be reconstructed from plastic materials. The layers are built by tracing a laser beam on the surface of a vat of liquid photopolymer [69]. The liquid solidifies very quickly when it is hit by the laser beam, and the layers bound together due to the self-adhesive property of the material. Some of the advantages of stereolithography are the accuracy of reproduction and the larger size of objects that can be reproduced.

Stereolithography bas been successfully deployed in production-ready systems for automated hearing aid shell production. An example of such system is Viper SLA in *Figure 2.6* capable to construct very accurate and fine detailed hearing aid shells on production basis.

With the help of rapid prototyping systems the production of hearing aid shells is converted from a manual process to a digitally automated process.

Figure 2.6 Left: Viper SLA rapid prototyping systems; Right: Hearing aid shells produced with this system

2.7 Video otoscopy

An *otoscope* or *auriscope* is a medical device used to visualize the external ear canal. The examination of the ear canal with an otoscope is called otoscopy. In the most basic form an otoscope consists of a handle and a head containing a light source and a magnifying lens. Disposable plastic ear speculums can be attached in the front end of the head. The speculum is the part of the otoscope inserted in the ear canal. Its conical shape limits the insertion depth in order to protect the ear drum of injuries. The examiner can visualize the inside of the ear canal through the lens.

The v*ideo otoscope* (VO) is an optical device very similar to a standard otoscope where the eye is substituted by a miniaturized high resolution color camera at the focal point of a rod lens optical system. The rod lens is surrounded by a fiber optic bundle with the role of transmitting the source light [19]. Such a device transfers images of the ear canal to the internal CCD sensor of the camera and outputs them to a Video Monitor or to Image-Video Capturing device. For most VO systems the high intensity light is produced remotely by a fan-cooled halogen light bulb. Transmission of the light through the fiber optic bundle avoids heat generation at viewing point [19].

The examination of the ear with a video otoscope is called video otoscopy and this practice continues to gain acceptance as an integral component of hearing health care practice today [18].

The video otoscopes come in different forms and shapes from a large number of manufacturers including *Welch Allyn*, *MedRx*, *Siemens Hearing Instruments*, *GN Otometrix*, and others. The miniaturization of different parts of a video otoscope allows manufacturers to build very small, portable and self-contained units as *CompacVideo Otoscope* from *Welch Allyn* in Figure 2.7 a). This kind of video otoscopes have completely internal optical system, light source and video camera and are powered by rechargeable batteries hosted by the handle. They offer all the advantages of other sophisticated units while keeping a small size and a relative low price.

Image freezing buttons, conectivity with video monitors, VCRs, printers or computers through video capturing devices are common features for most of the video otoscopes.

a) Welch Allyn CompacVideo Otoscope

b) GN Otometrics OTOCam Video Otoscope System

Figure 2.7 Examples of Video Otoscopes

Video otoscopes have many applications in audiologists practice including examination of the ear canal and ear drum, physician communication, hearing aids selection and fitting, cerumen management, patient education [18].

With the help of video otoscopy the specialists can make recommendations regarding the type of hearing aid best suited for a patient, can detect the factors that may cause problems in the impression-taking process, or can pre-select and verify an oto-block placement site before taking the ear impression [19].

Video otoscopy is the first essential step performed in the fitting and selection of custom hearing aids.

2.8 Discussion

Among different types of hearing aids, CIC present many advantages. They are invisible for the others (cosmetically appealing), and assure a natural sound reception. A good CIC hearing aid has to fit very well in the ear canal in order to give maximum performance and also to be comfortable for the wearer.

The production of hearing aids shells is a complex and time consuming process mainly because it is based on ear impressions. Taking the ear impression is a very invasive procedure for the patients and requires extremely qualified skills of the operator. If this process is not properly done, there is always the risk of producing traumas of the ear canal or ear drum. In general, taking the ear canal impression is an unpleasant experience for the patients. Despite of these negative aspects, it is the key step in the creation of a customized hearing aid. This is because the shape accuracy of the hearing aid shell is as good as the ear canal impression accuracy.

In order to produce a physical shell for a hearing aid, the ear canal impression is scanned normally with 3D laser based scanner. Even if today many producers offer ear impressions scanner, these are in general expensive systems. The digital model obtained after 3D scanning process is used to create an accurate replica of the ear canal using a rapid prototyping system.

On the other side, the video otoscope becomes standard equipment in the ear specialist office. It is widely used for the inspection of the ear canal, diagnose, hearing instrument selection and fitting. The shape of the otoscope head makes examination of the ear canal very safe for the patients and doesn't require specialized skills. Today the video otoscopes became very popular because they can offer both the advantages of a very small size and affordable prices.

If we consider the video otoscope is a special camera able to take images inside the ear canal, then the question that comes is if it's possible to use these images for building the 3D model of the ear canal. Building 3D models of real scenes from sequences of images (known as Structure from Motion problem) has been largely studied in the last two decades, and some techniques reached their maturity and are successfully used in many real systems including medical area. If it would be possible to model the ear canal directly from otoscopic images, then two out of the three steps required to build a custom shell are eliminated: 1) taking the ear canal impression and 2) scanning the impression. The result will be a simpler and cheaper system based on standard equipment that normally can be found in many of the ear specialist offices. But the greatest advantages are on the patient side where a risky and very specialized procedure (ear impression taking) may be replaced with a very usual and less invasive one (video otoscopy).

The reason of this short review is to emphasize the motivation of writing this thesis. The first question we try to find the answer here is if it’s possible to use the otoscopic images and Structure from Motion techniques in order to create a 3d model of the ear canal. This also includes the conditions under which this is possible. Another important issue that will be covered in the second part of this thesis is to see how accurately the tube-like objects can be reconstructed with SFM methods.

CHAPTER 3

The Structure from Motion problem

Structure from Motion refers to the 3D reconstruction of a rigid (static) object (or scene) from 2D images of the object /scene taken from different positions of the camera.

A single image doesn't provide enough information to reconstruct the 3D scene due to the way an image is formed by projection of a three-dimensional scene onto a two-dimensional image. As an effect of the projection process the depth information is lost. Anyway, for a point in one image, its corresponding 3D point is constraint to be on the associated line of sight. But it is not possible to know where exactly on this line the 3D point is placed. Given more images of the same object taken from different poses of the camera, the 3D structure of the object can be recovered along with the camera motion.

In this chapter the relation between different images of the same scene is discussed. First a camera model is introduced. Then the constraints existing between image points corresponding to the same 3D point in two different images are analyzed. Next it will be shown how a set of corresponding points in two images can be used to infer the relative motion of the camera and the

structure of the scene. Finally, a specific structure from motion algorithm is presented.

The relations between world objects and images are subject of Multiple View Geometry, used to determine the geometry of the objects, camera poses, or both. An excellent review of the Multiple View Geometry can be found in [20].

As the 3D reconstruction process of an object is based on images, it is important to understand before how the images are formed. Thus a mathematical model of the camera has to be introduced.

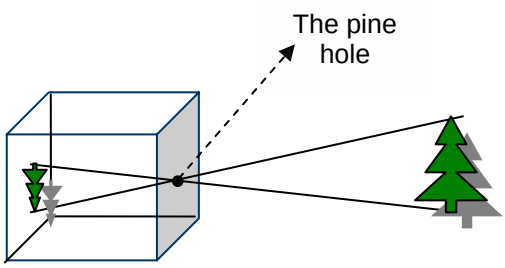


Figure 3.1 Pinhole camera

3.1 Camera model

The most basic camera model but used on a large scale in different computer vision problems is the perspective camera model. This model corresponds to an ideal pinhole camera and it is completely defined by a projection center *C* (also known as focal point, optical center, or eye point) and a focal plane (image plane).

The pinhole camera doesn't have conventional lens, it can be imagined as a box with a very small hole in a very thin wall, such as all the light rays pass through a single point (see *Figure 3.1*).

Some basic concepts are illustrated in *Figure 3.2*. The distance between projection center and the image plane is called focal length. The line passing through the center of projection and it is orthogonal to the retinal plane is called

optical axis (or principal axis), and defines the path along which light propagates through the camera. The intersection of the optical axis with the focal plane is a point *c* called principal point. The plane parallel to the image plane containing the projection centre is called the *principal plane* or *focal plane* of the camera.

The relationship between the 3D coordinates of a scene point and the coordinates of its projection onto the image plane is described by the *central* or *perspective projection.* For the pinhole model, a point of the scene is projected onto the retinal plane at the intersection of the line passing through the point and projection center with the retinal plane [2], as sown in *Figure 3.2*. In general, this model can approximate well most of the cameras.

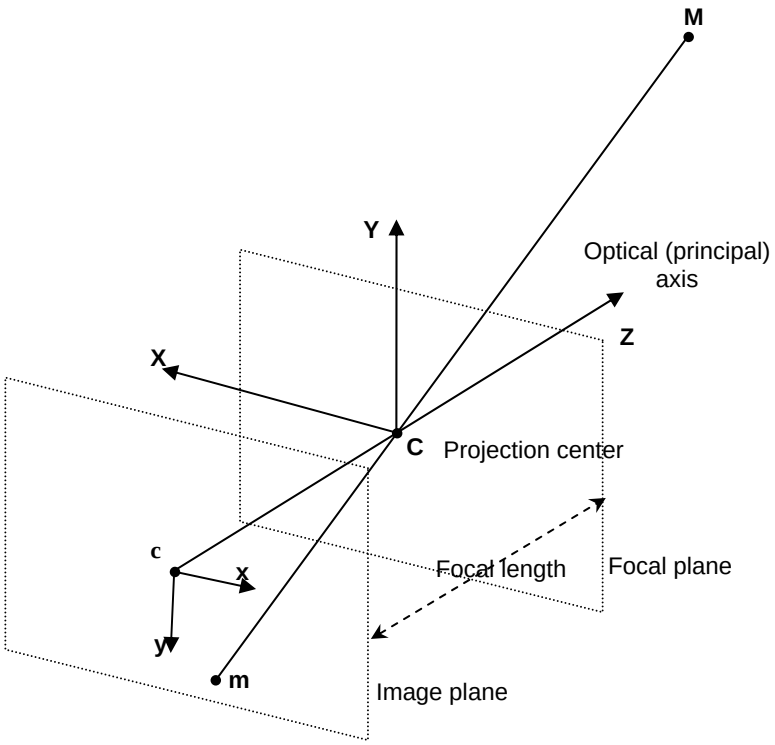


Figure 3.2 Pinhole camera geometry

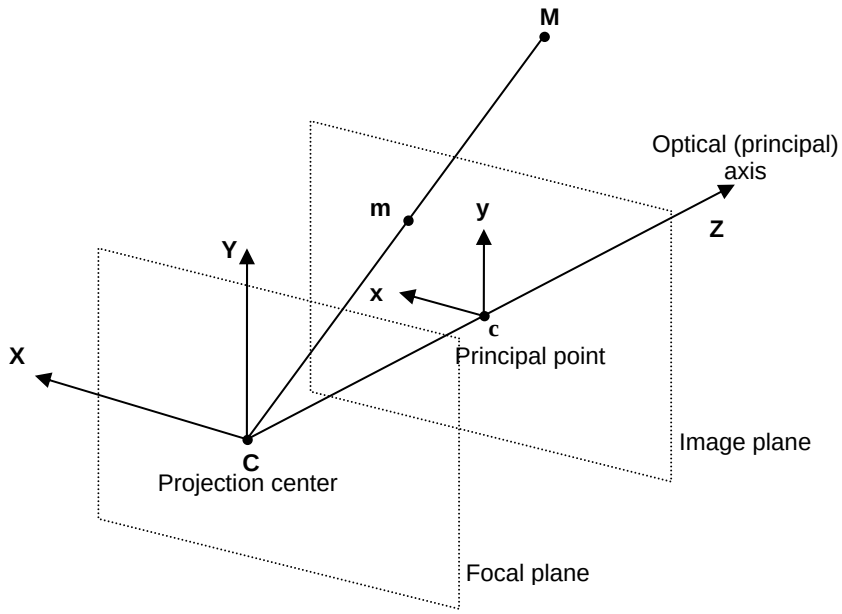


Figure 3.3 Pinhole camera geometry. The image plane is replaced by a virtual plane located on the other side of the focal plane

It is not important from geometric point of view on which side of the focal plane it is located the image plane. This is illustrated in *Figure 3.2* and *Figure 3.3.*

In the most basic case the world coordinate system origin is placed at the projection center, with the plane *x-y* parallel to the image plane, and Z-axis is identical to the optical axis.

If the 2D coordinates of the projected point *m* in the image are *(x, y)*, and the 3D coordinates of the point *M* are *(X, Y, Z)*, then applying Thales theorem for the similar triangles in *Figure 3.4* results in:

$$\frac{y}{Y} = \frac{f}{Z}, \text{ and similarly } \frac{x}{X} = \frac{f}{Z} \qquad (3.1)$$

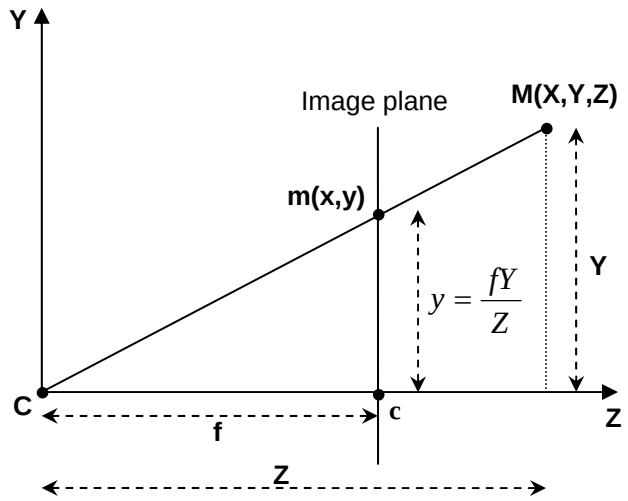


Figure 3.4 The projection of camera model onto YZ plane

Any point on the line *CM* project into the same image point *m*. This is equivalent to rescaling of point represented in homogenous coordinates.

$(X,Y,Z) \sim s(X,Y,Z) = (sX, sY, sZ)$. "~" means "equal up to a scale factor".

$$x = f\frac{X}{Z} = f\frac{sX}{sZ} \text{ ,and } y = f\frac{Y}{Z} = f\frac{sY}{sZ} \tag{3.2}$$

3.2 The camera projection matrix

If the world and image points are represented by homogeneous vectors, then the equation (3.2) can be expressed in terms of matrix multiplication as

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.3}$$

The matrix $P = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ is called perspective projection matrix.

If the 3d point is $M = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$ and its project onto the image plane is $m = \begin{bmatrix} x & y \end{bmatrix}^T$, and if $\tilde{M} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ and $\tilde{m} = \begin{bmatrix} x & y & 1 \end{bmatrix}^T$ are the homogenous representation of *M* and *m* (obtained by adding 1 in the end of the vectors), then the equation (3.3) can be written in a more simple way as:

$$s\tilde{m} = P\tilde{M} \tag{3.4}$$

Introducing homogenous representation for the image points and the world points made the relation between them to be linear.

The camera model is valid only in the special case when the *z-axis* of the world coordinate system is identical to the optical axis. But it is often required to represents the points in an arbitrary world coordinate system.

The transformation from the camera CS with center in *C* to the world CS with center in O is given by a rotation R_{3x3} followed by a translation $t_{3x1} = CO$ as shown in *Figure 3.5*. These fully describe the position and orientation of the camera in the world CS, and are called *extrinsic* parameters of the camera.

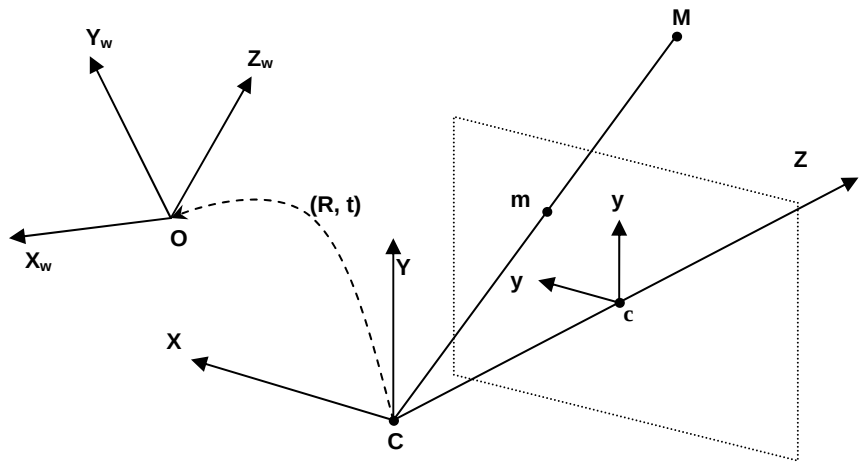


Figure 3.5 From camera coordinates to world coordinates

If a point M_C in the camera coordinate system corresponds to the point M_W in the world coordinate system, then the relation between them is:

$M_C = RM_W + t$, or in homogenous coordinates:

$$\tilde{M}_C = G\tilde{M}_W \tag{3.5}$$

where the matrix G_{4x4} is

$$G = \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \tag{3.6}$$

From (3.4) and (3.5) results that

$$m \sim PM_C = PGM_W = P_{new}M_W \tag{3.7}$$

In real cases the origin of the image coordinate system is not the principal point and the scaling corresponding to each image axis is different. For a CCD camera these depend on the size and shape of the pixels (it may happen that they are no perfectly rectangular), and also on the position of CCD chip in the camera [2]. Thus, the coordinates in the image plane are further transformed by multiplying the matrix *P* to the left by a 3 × 3 matrix *K*. The relation between pixel coordinates and image coordinates is depicted in *Figure 3.6.* The camera perspective model becomes:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim K \cdot \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.8}$$

K is usually represented as an upper triangular matrix of the form:

$$K = \begin{bmatrix} k_u & k_v \cot\theta & u_0 \\ 0 & k_v / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

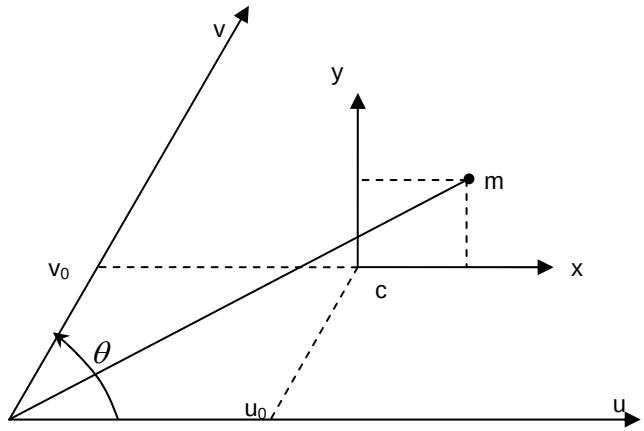


Figure 3.6 Relation between pixel coordinates and image coordinates.

where k_u and k_v represent the scaling factors for the two axes of image plane, θ is the skew between the axes, and (u_0, v_0) are the coordinates of the principal point. These parameters encapsulated in the matrix *K* are called intrinsic camera parameters. *K* it is not dependent on camera position and orientation.

Including K in (3.8) then the camera model becomes:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} k_u & k_v \cot\theta & u_0 \\ 0 & k_v / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \Leftrightarrow$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} fk_u & fk_v \cot\theta & u_0\,0 \\ 0 & fk_v / \sin\theta & v_0\,0 \\ 0 & 0 & 1\;0 \end{bmatrix} \cdot \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \Leftrightarrow$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} \alpha_u & \alpha_v \cot\theta & u_0 \\ 0 & \alpha_v / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.10)$$

where $\alpha_u = fk_u$ and $\alpha_v = fk_v$.

If we note $G = \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix}$, then this equation can be written in a simpler form

$$\tilde{m} = AP_N G\tilde{M} = A\begin{bmatrix} R & t \end{bmatrix}\tilde{M} = P\tilde{M} \tag{3.11}$$

where *P* from the above equation is the *camera projection matrix.*

The new matrix

$$A = \begin{bmatrix} \alpha_u & \alpha_v \cot\theta & u_0 \\ 0 & \alpha_v / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

contains only intrinsic camera parameters and it is called camera calibration matrix. The values u_0 and v_o correspond to the translation of the image coordinates such as the optical axis passes through the origin of image coordinate.

For a camera with fixed optics, intrinsic parameters are the same for all the images taken with the camera. But these parameters can obviously change from one image to another for the cameras with zooming and auto-focus functions.

In practice the angle between axes it is often assumed to be $\theta = \pi / 2$. Then the final camera model is:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.13}$$

3.3 Normalized coordinates

We say that the camera coordinate system is normalized when the image plane is place at unit distance from the projection center (focal length $f = 1$). If we go back to the equation (3.3) it can be seen that in this case the projection matrix P becomes:

$$P_N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.14}$$

Assuming that calibration matrix *A* from relations (3.10), (3.11) is known, then the image coordinates in a normalized camera coordinate system are:

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ 1 \end{bmatrix} = A^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3.15}$$

where normalized image coordinates corresponding to a 3D point *M(X,Y,Z)* are as simple as:

$$\hat{x} = \frac{X}{Z} \text{ and } \hat{y} = \frac{Y}{Z} \tag{3.16}$$

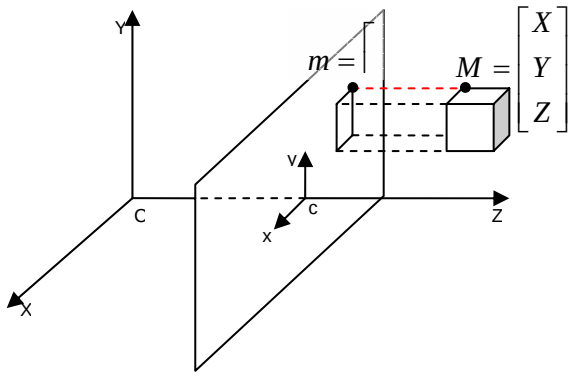


Figure 3.7 Orthographic projection

3.4 Approximations of the perspective model

The perspective projection as formulated in equation (3.2) is a nonlinear mapping. Often it is more convenient to work with a linear approximation of the perspective model. The most used linear approximations are:

- *Orthographic projection (Figure 3.7)*: is the projection through an infinite projection center. The dept information disappears in this case. It can be used when distance effect can be ignored.
- *Weak perspective projection (Figure 3.8)*. In this model, the points are first orthographically projected onto a plane Z_C (all the points have the same depth) and then the new points are projected onto the image plane with a perspective projection. This model is useful when the object is small comparing with the distance from the object to the camera.

The projection matrix for orthographic model (*Figure 3.7*) is:

$$P_{ort} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.17}$$

For the weak perspective projection (*Figure 3.8*), assuming normalized coordinates (focal length *f=1*) we can write:

$$x = \frac{X}{Z_C}, \text{ and } y = \frac{Y}{Z_C}; \tag{3.18}$$

The projection matrix for the weak perspective model is:

$$P_{wp} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & Z_C \end{bmatrix} \tag{3.19}$$

The weak perspective model becomes:

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P_{wp}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.20}$$

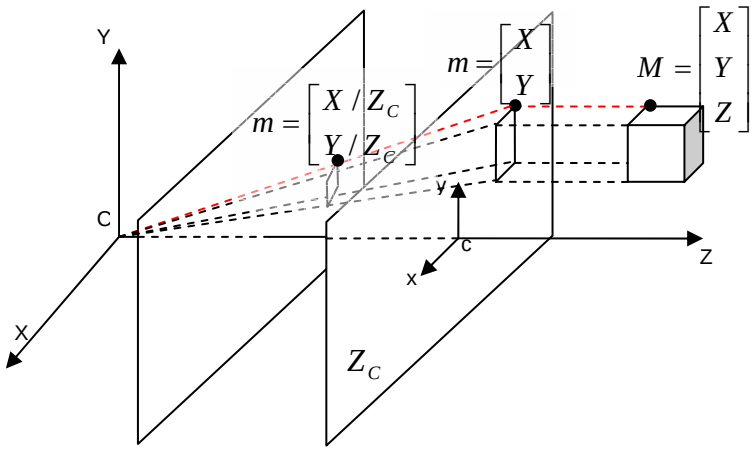


Figure 3.8 Weak perspective projection

Adding intrinsic and extrinsic camera parameters, the final weak perspective model becomes:

$$s\tilde{m} = AP_{wp}G\tilde{M} \,, \tag{3.21}$$

where A contains intrinsic camera parameters (same as in equation 3.10), and G contains extrinsic camera parameters (see equation 3.11).

3.5 Two-view geometry

Two-view geometry, also known as epipolar geometry, refers to the geometrical relations between two different perspective views of the same 3D scene.

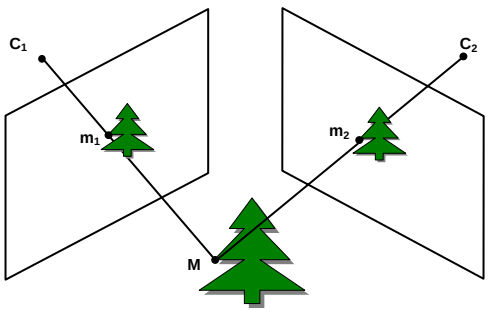


Figure 3.9 Corresponding points in two views of the same scene.

The projections m_1 and m_2 of the same 3D point M in two different views are called *corresponding points* (see *Figure 3.9).* The epipolar geometry concepts are illustrated in *Figure 3.10.*

A 3D point *M* together with the two centers of projection C_1 and C_2 form a so called *epipolar plane*. The projected points m_1 and m_2 also lie in the epipolar plane. An epipolar plane is completely defined by the projection centers of the camera and one image point.

The line segment joining the two projection centers is called *base line* and intersects the image plane in points e_1 and e_2 called *epipoles* representing the projection of the center of projection in opposite image.

The intersection of the epipolar plane with the two image planes forms the lines l_1 and l_2 called *epipolar lines.*

It can be observed that all the 3D points located on the epipolar plane project on the epipolar lines l_1 and l_2 .Another observation is that the epipoles are the same for all the epipolar planes.

Given the projection m_1 of an unknown 3D point *M* in the first image plane, the epipolar constraint limits the location of the corresponding point in the second image plane to lie on the epipolar line l_2 . The same is valid for a projected point m_2 in the second image plane; its corresponding point in the first image plane is constrained to lie on the epipolar line l_1 .

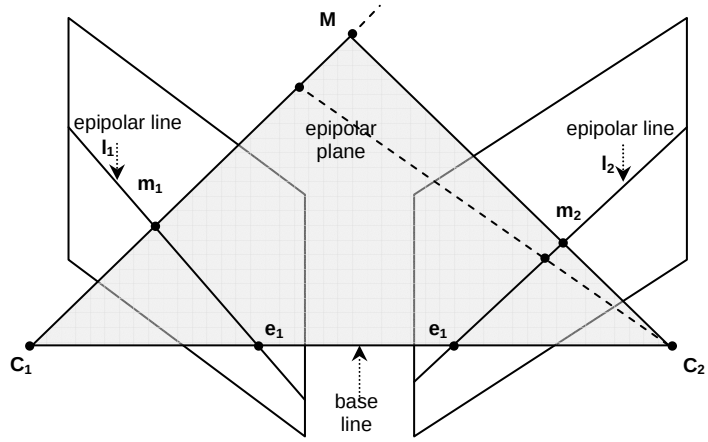


Figure 3.10 Epipolar geometry and the epipolar constraint

In order to find out the equation of the epipolar line, the equation of the optical ray going through a projected point *m* is obtained first (for a given projection matrix *P*).

The optical ray is the line going through the projection center *C* and the projected point *m.* All the point on this ray also projects on *m*. Then a point *D* on the ray can be chosen such as its scale factor is 1;

$$\tilde{m} = P\begin{bmatrix} D \\ 1 \end{bmatrix} \tag{3.22}$$

As *P* is a 3x4 matrix, we can write $P = \begin{bmatrix} B & b \end{bmatrix}$, where B_{3x3} is formed by the first 3 columns in *P*, and b_{3x1} is the last column in *P*.

The relation (3.22) becomes $\tilde{m} = \begin{bmatrix} B & b \end{bmatrix}\begin{bmatrix} D \\ 1 \end{bmatrix}$, and the 3D point *D* is obtained as:

$$D = B^{-1}\left(-b + \tilde{m}\right) \tag{3.23}$$

Then a point on the optical ray is given by the next equation:

$$M = C + \lambda\left(D - C\right) = B^{-1}(-b + \lambda\tilde{m}) \tag{3.24}$$

with $\lambda \in \left(0, \infty\right)$, or

$$\tilde{M} = \begin{bmatrix} -B^{-1}b \\ 1 \end{bmatrix} + \lambda\begin{bmatrix} B^{-1}\tilde{m} \\ 0 \end{bmatrix} \tag{3.25}$$

As it was mentioned above, the equation of the optical ray will be used in order to estimate the equation of the epipolar line. It is assumed that the projected point in the first image plane m_1 is known, and the corresponding epipolar line in the second image plane will be determined.

Let P_1 and P_2 be the projection matrices of the two cameras corresponding to the two views, and m_1 a projected point on the first image plane. The projection

of the optical ray going through the point m_1 onto the second image plane gives the corresponding epipolar line. This can be written as:

$$s_2\tilde{m}_2 = P_2\tilde{M} = P_2\begin{bmatrix} -B_1^{-1}b_1 \\ 1 \end{bmatrix} + \lambda_1 P_2\begin{bmatrix} B_1^{-1}\tilde{m}_1 \\ 0 \end{bmatrix} \qquad (3.26)$$

In a simplified form, the equation of the epipolar line l_2 can be written as:

$$s_2\tilde{m}_2 = e_2 + \lambda_1 B_2 B_1^{-1}\tilde{m}_1 \qquad (3.27)$$

The above equation describes a line going through the epipol e_2 and the point $B_2B_1^{-1}\tilde{m}_1$ - the projection of the point at infinite of the optical ray of m_1 onto the second image plane. In a similar way the epipolar line in the first image plane can be obtained.

The equation (3.27) describes the epipolar geometry between two views in the term of projection matrices, and assumes that both intrinsic and extrinsic parameters of the camera are known. When only the intrinsic parameters of the camera are known, the epipolar geometry is described by the *essential matrix*, and when both intrinsic and extrinsic parameters are unknown, the relation between the views is described by the *fundamental matrix*.

In the case of three views it is also possible to determine the constraint existing between them. This relationship is expressed by the trifocal tensor and it is described for example in [23].

3.5 The essential matrix

Let's suppose that two cameras view the same 3D point *M*, projecting onto the two image planes at $\tilde{m}_1$ and $\tilde{m}_2$. When the intrinsic parameters of the camera are known (cameras are calibrated), the image coordinates can be normalized, as explained in section 3.3.

If the world coordinates system is aligned with the first camera, then the two projection matrices are:

$$P_1 = [I \quad 0], \text{ and } P_2 = [R \quad t] \tag{3.28}$$

Substituting P_1 and P_2 in the equation (3.27) gives

$$s_2 \tilde{m}_2 = t + \lambda_1 R \tilde{m}_1 \tag{3.29}$$

The interpretation of the equation (3.29) is that the point $\tilde{m}_2$ is on the line passing through the points t and $R\tilde{m}_1$. In homogenous coordinates the line passing through two given points is their cross product, and a point lies on a line if the dot product between the point and the line is 0. Thus the equation (3.29) this can be also expressed as:

$$\tilde{m}_2^T (t \times R\tilde{m}_1) = 0 \tag{3.30}$$

The cross product of two vectors in 3d space can be expressed by the product of a skew symetric matrix and a vector. If $a = [a_1 \quad a_2 \quad a_3]^T$ and $b = [b_1 \quad b_2 \quad b_3]^T$ then the cross product $a \times b$ is:

$$a \times b = [A]_\times b = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{3.31}$$

In the context of the above definition of the cross product, the equation (3.30) can be also written as:

$$\tilde{m}_2^T [t]_\times R\tilde{m}_1 = \tilde{m}_2^T E\tilde{m}_1 = 0 \tag{3.32}$$

where the matrix *E* is called *the essential matrix.*

$$E \overset{\Delta}{=} [t]_\times R \tag{3.33}$$

One property of the essential matrix is that it has two equal singular values, and a third one hat is equal to zero. Then the singular values decomposition (SVD) of the matrix *E* can be written as:

$$E = U \sum V^T \quad \text{with } \sum = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.34}$$

If E is the essential matrix of the cameras (P_1, P_2), then E^T is the essential matrix of cameras (P_2, P_1).

If $\tilde{m}_1$ and $\tilde{m}_2$ are projected points in image planes, then the corresponding epipolar lines in the other image are:

$$\begin{aligned} l_2 &= E\tilde{m}_1 \\ l_1 &= E^T \tilde{m}_2 \end{aligned} \tag{3.35}$$

Since the epipolar lines contain the epipoles then:

$$\begin{aligned} &e_2^T E\tilde{m}_1 = 0 \text{ for all } \tilde{m}_1 \Rightarrow \\ &e_2^T E = 0 \text{ and } Ee_1 = 0 \end{aligned} \tag{3.36}$$

The essential matrix encapsulates only information about extrinsic parameters of the camera, and has five degrees of freedom: three of them correspond to the 3D rotation, and two correspond to the direction of translation. The translation can be recovered only up to a scale factor.

3.6 The fundamental matrix

When the cameras intrinsic parameters are not known, the epipolar geometry is described by the *fundamental matrix.* This matrix is derived in a similar way as the essential matrix, but this time starting from the general equation of the camera model (3.11). If the world's coordinate system is aligned with the first camera, then the projection matrices are:

$$P_1 = A_1 \begin{bmatrix} I & 0 \end{bmatrix} \text{ and } P_2 = A_2 \begin{bmatrix} R & t \end{bmatrix} \tag{3.37}$$

Substituting these general projection matrices in the equation of the epipolar line (3.27) results in:

$$s_2\tilde{m}_2 = e_2 + \lambda_2 A_2 R A_1^{-1}\tilde{m}_1, \text{ and } e_2 = A_2 t \tag{3.38}$$

The signification of the equation (3.28) is that the point $\tilde{m}_2$ is placed on the line going through the points e_2 and $A_2 R A_1^{-1}\tilde{m}_1$, and in homogenous coordinates it can be rewritten in the form:

$$\tilde{m}_2^T [e_2]_\times A_2 R A_1^{-1}\tilde{m}_1 = 0 \Leftrightarrow \tag{3.39}$$

$$\tilde{m}_2^T F \tilde{m}_1 = 0 \tag{3.40}$$

The matrix *F*

$$F = [e_2]_\times A_2 R A_1^{-1} \tag{3.41}$$

is the *fundamental matrix* and encapsulates the relation between corresponding points in the two images in pixel coordinates.

A property of the fundamental matrix is that it is singular (it has rank 2) since $\det[t] = 0$. The fundamental matrix has seven degrees of freedom (even if there are nine parameters) because of the constraint $\det[t] = 0$ and the scaling that is not significant.

3.7 Estimation of the fundamental matrix

From the equation (3.40) it can be observed that the fundamental matrix is defined only by the correspondences of the points in pixel coordinates. It means the fundamental matrix can be calculated for a given set of point correspondences in two images.

If the matrix *F* is written as:

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f33 \end{bmatrix}, \tag{3.42}$$

then the equation (3.40) can be written as:

$$\begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f33 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = 0 \tag{3.43}$$

If f is a vector containing the elements of the matrix F then from (3.43) results that:

$$\begin{bmatrix} x_2x_1 & x_2y_1 & x_2 & y_2x_1 & y_2y_1 & y_2 & x_1 & y_1 & 1 \end{bmatrix} \cdot \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0 \Leftrightarrow b^T f = 0 \tag{3.44}$$

Often the constraint $\det F = 0$ is not taken in consideration and the matrix F is estimated from a set of $n \geq 8$ point correspondences by the so called *eight point algorithm*. Each point correspondence gives an equation linear in elements of F. Then the linear set of equations is given by

$$Bf = 0 \tag{3.45}$$

where a line of the matrix B corresponds to a pair of points.

The solution for F is obtained by solving the linear system of equation in (3.45). The least squares solution of F is obtained by performing a singular value decomposition of the matrix $B^T B$. Then F is the eigenvector corresponding to the smallest eigen value.

The eight point algorithm doesn't give the optimal solution for F since the constrained $\det F = 0$ is not enforced. But this approximation can be used to initialize more complex algorithms (see for example [20]).

If SVD of the computed *F* is:

$$F = U\begin{bmatrix}\sigma_1 & 0 & 0\\ 0 & \sigma_2 & 0\\ 0 & 0 & \sigma_3\end{bmatrix}V^T = U_1\sigma_1 V_1^T + U_2\sigma_2 V_2^T + U_3\sigma_3 V_3^T, \quad (3.46)$$

then the closest rank two approximation for *F* is:

$$F = \arg\min_F \left\| F - F' \right\|, \quad (3.47)$$

where

$$F' = U\begin{bmatrix}\sigma_1 & 0 & 0\\ 0 & \sigma_2 & 0\\ 0 & 0 & 0\end{bmatrix}V^T = U_1\sigma_1 V_1^T + U_2\sigma_2 V_2^T \quad (3.48)$$

A better solution for computing F is presented in [21] where the points' coordinates are normalized before solving the homogenous system of linear equations: they are translated such as their centroid is at the origin and are scaled such as their average distance from the origin is $\sqrt{2}$. The normalized 8-point algorithm is summarized in *Table 3.1*

Given the set of corresponding points in two images $C = \{(p_i, q_i), i = 1\ldots N \mid p_i \in \mathrm{Im}age1, q_i \in \mathrm{Im}age2\}$, perform following steps:

1. Center the points in each image around their centroid and scale them such as their average distance from the origin is $\sqrt{2}$: $p_i' = T_1 p_i$, and $q_i' = T_2 q_i$
2. From the points p_i' and q_i' compute matrix *F* using 8-point algorithm.
3. Enforce the rank 2 constraint on *F*
4. Return $T_1^T F T_2$

Table 3.1 Normalized 8-point algorithm

3.8 Robust estimation of the fundamental matrix

As it was already seen, the relations between different views of the same scene is based on the correspondences between points in different views. Extracting and matching points in images is a topic largely addressed in computer vision literature, since many other algorithms are based on these correspondences. This problem, also known as feature detection and tracking, will be addressed in a separate section.

A common problem of the feature detection and matching algorithms is that in some situations they provide false correspondences. This will drastically affect the quality of the reconstruction since the computed essential/fundamental matrix is not the correct one. It is difficult to split the set of matches in inliers and outliers before having the correct solution for essential/fundamental matrix [2].

A solution for this problem was proposed in [24]. The algorithm is called RANSAC (Random Sampling Consensus) and can be used to estimate parameters of a mathematical model from a set of observed data containing outliers. The main idea is to iteratively select a random subset of the original data and build the model using this subset. Then the data set can be segmented in inliers and outliers according to this model. Repeting this procedure with randomly selected subsets, the correct solution is the one with the largest number of inliers.

When the fundamental matrix is estimated, a potential set of point matches is provided. Random samples of 8 matches are taken and based on them the fundamental matrix is computed using the normalized 8-point algorithm. Matches are considered inliers if the distance from the points to their corresponding epipolar line is not larger than a certain threshold (0.5 or 1 pixel according to [2]).

The remaining problem is how many samples should be taken, since testing all the possibilities can be impractical. In [24] it is shown that the numbers of samples can be computed such as a good sample is selected with a certain probability.

If z is the probability that at least one of the random samples is error-free the minimal number k of samples is:

$$k = \frac{\log(1-z)}{\log(1-w^n)} \tag{3.49}$$

where n is the number of points in the sample, and w is the probability that any selected data point is within the error tolerance of the model. In other words, if ε is the probability for a point to be an outlier then $w = 1-\varepsilon$ and the number of samples k is:

$$k = \frac{\log(1-z)}{\log(1-(1-w)^n)} \tag{3.50}$$

In practice the standard deviation of k (or multiplies) is added to this minimal number of samples. The standard deviation is given by

$$SD(k) = \frac{\sqrt{1-w^n}}{w^n} \tag{3.51}$$

For example, we have a set of possible matches with a probability of outliers $\varepsilon = 20\%$, and we want to estimate the fundamental matrix with a probability of z=95% that a good sample was selected, for a number of 8 points per sample.

$$n = 8, \varepsilon = 0.2, z = 0.95;$$

$$k = \frac{\log(1-0.95)}{\log(1-(1-0.2)^8)} = 16.31$$

$$SD(k) = \frac{\sqrt{1-(1-0.2)^8}}{(1-0.2)^8} = 5.43$$

A practical approach is to decide in advance the fraction of outliers the algorithm can deal with, and set the number of samples accordingly [2]. The number of samples function of the probability of outliers is shown in *Table 3.2*, for a probability of 95% that a good sample is selected.

Outliers	5%	10%	20%	30%	40%	50%	60%	70%	80%
k	3	5	16	50	177	765	4570	45658	1170206
SD(k)	1	2	5	17	59	255	1525	15241	390624

Table 3.2 The minimum number of 8-point samples along with their standard deviation to ensure a probability of 95% for the given fraction of outliers.

The algorithm that computes the fundamental matrix in a robust way can be summarized in the following steps:

1. Find a set of potential matches
2. while the probability of getting a good sample < 95% do
   - 2.1 select a sample (8 matches)
   - 2.2 compute the fundamental matrix *F*
   - 2.3 determine the inliers
   - 2.4 if the number of inliers is bigger than previous maximum, retain the configuration
3. Refine *F* based on the inliers given by the best configuration
4. Find more matches under the constraint imposed by *F*
5. Refine *F* based on all the correct matches

Table 3.3 Robust estimation of the fundamental matrix.

3.9 Triangulation

The triangulation refers to the reconstruction of the 3D point *X*, given the camera projection matrices P_1, P_2 (intrinsic and extrinsic parameters known) and a pair of corresponding points in the two images *x* and *x'* (projections of the same point *X* in the images). In the ideal case, the point *X* is located at the intersection of the two rays going through the points *x* and *x'*, as shown in *Figure 3.11*.

For an image point *x*, projection of a 3D point *X*, we have

$$sx = PX \Leftrightarrow \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} X \quad \Leftrightarrow \begin{cases} P_3 Xx = P_1 X \\ P_3 Xy = P_2 X \end{cases} \Leftrightarrow$$

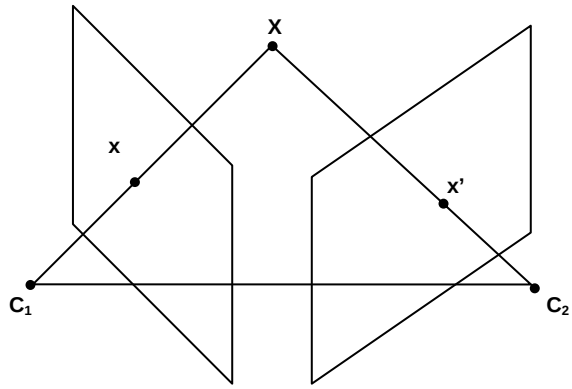


Figure 3.11 Exact triangulation

$$\begin{bmatrix} P_3 x - P_1 \\ P_3 y - P_2 \end{bmatrix} X = 0 \quad (3.37) \qquad (3.52)$$

Then the triangulation can be written as:

$$\begin{bmatrix} P_3 x - P_1 \\ P_3 y - P_2 \\ P_3^{'} x^{'} - P_1^{'} \\ P_3^{'} y^{'} - P_2^{'} \end{bmatrix} X = 0 \Leftrightarrow AX = 0 \qquad (3.53)$$

This least squares problem can be solved through singular value decomposition. If $A = U \sum V^T$ then the solution is the last column of *V*.

Now let's consider two corresponding points m_1 and m_2, projections of the 3D point *M* in two images. In reality the two rays going through the points m_1 and m_2 don't intersect due to the noise in the images, and the 3D point *M* is often chosen in practice as the mid point of the common perpendicular to the two rays, as shown in *Figure 3.12*.

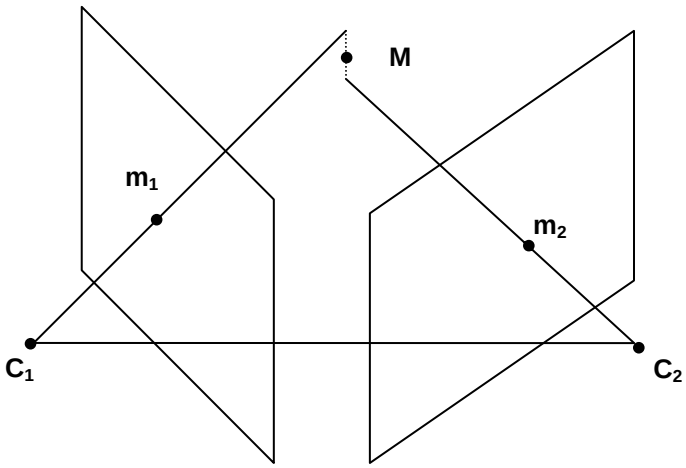


Figure 3.12 The reconstruction of a point in 3D from the projections in two images

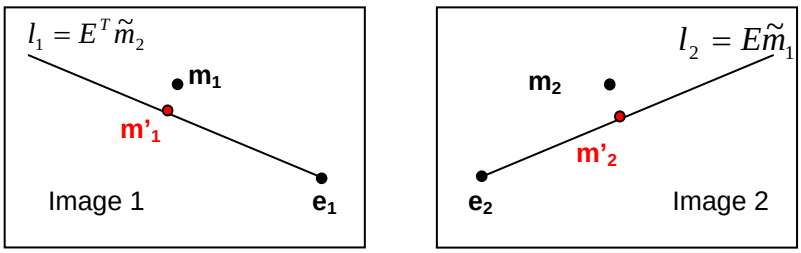


Figure 3.13 Optimal reconstructions in epipolar plane

For real images, the points m_1 and m_2 are not located on their corresponding epipolar lines. To find the optimal 3D point for a given epipolar plane, the closest point $m_1^{'}$ and $m_2^{'}$ are selected on the epipolar line (see *Figure 3.13)* and then the 3D point is computed through exact triangulation. This solution guarantees minimal reprojection error in the given epipolar plane.

3.10 3D reconstruction

The reconstruction of 3D points from corresponding image points depends on how much information is known about cameras. When both intrinsic and extrinsic parameters of the cameras are known, then the 3D points can be reconstructed by triangulation. When only the intrinsic parameters are known, then the structure can be recovered up to an arbitrary scale factor (also known as *Euclidean reconstruction*). If both intrinsic and extrinsic parameters are unknown then the structure can be recovered up to a projective transformation [2].

In this thesis the problem of reconstruction with calibrated cameras (known intrinsic parameters) is addressed. In this case the epipolar geometry is described by the essential matrix *E* (normalized image coordinates are assumed).

In equation (3.33) we have seen that the essential matrix encapsulates only the extrinsic parameters of the camera $E \overset{\Delta}{=} [t]_{\times} R$, when the world coordinate system is aligned with the first camera. The extrinsic parameters of the second camera (R,t) have to be solved in order to be able to recover the structure.

The essential matrix can be estimated from a number of point correspondences using normalized 8-point algorithm. The Rotation *R*, and translation *t* can be extracted from the matrix *E*, with the respect of the following theorem formulated and demonstrated in [22]:

Theorem: *A 3 X 3 matrix is decomposable into a skew-symmetrical matrix postmultiplied by a rotation Matrix if and only if one of its singular values is zero and the other two are equal.*

The solution is not unique, but there is only a correct one corresponding to positive depth values. The rotation can be fully recovered and the translation can be recovered only up to a scale factor. This is because the essential matrix itself it is known only up to a scale factor (different scales of *t* in equation 3.33 give the same essential matrix).

If the recovered rotation *R* and translation *t* are used to build the cameras' matrices as in equation (3.28), then the structure of the scene can be recovered by triangulation from known point correspondences. Knowing the translation *t* only up to a scale factor, will result in a reconstruction up to a scale.

3.11 Structure and motion from multiple views

To this point it was shown how two images of the same scene can be related. Moving further to multiple views, the structure from motion problem can be formulated as finding the structure $M_j, j = 1 \ldots n$ and the motion R_i and $t_i, i = 1 \ldots m$ that beast fit to the data m_j, given the perspective camera model. More formally, given $i = 1 \ldots m$ views of $j = 1 \ldots n$ feature points describing a rigid object, find M_j, R_i, t_i that minimize:

$$\arg\min_{\mathrm{M}_j, R_i, t_i} \sum_{i=1}^{m} \sum_{j=1}^{n} \left\| m_{ij} - A_i [R_i \mid t_i] M_j \right\|^2 \qquad (3.54)$$

This problem can be solved by a non-linear optimization algorithm like Levenberg Marquardt method. In general calibration matrix A_i is known. In order to avoid local minima, an initialization close to the real solution has to be provided.

In practice, other methods are used to obtain the initial guess for structure and motion, and the above optimization invariantly appears as the last optimization step in many structure from motion algorithms. This optimization step it is also known as *bundle adjustment.* For further details please refer to [29].

The large number of unknowns in this problem makes it very expensive from computational point of view. A very efficient implementation of this optimization method can be found in [30], where the sparse block structures in the normal equations are exploited. This publicly available software package was used in the experiments performed in this thesis.

3.12 Factorization method

The factorization method was first proposed by [25]. It provides an initial guess of the structure and motion by linearizing the perspective camera model under orthographic projection.

Let's consider number P of 3D points $s_p\left(X_p, Y_p, Z_p\right), p = 1 \ldots P$ and the corresponding image points in F frames $\left\{\left(x_{fp}, y_{fp}\right) \mid f = 1 \ldots F, p = 1 \ldots P\right\}$. The notation $\left(x_{fp}, y_{fp}\right)$ refers to the point p in the frame f.

The horizontal coordinates of the points are arranged in a matrix $X_{F \times P}$, each row corresponds to one frame, and each column to one point. Similarly the vertical coordinates are arranged in the matrix $Y_{F \times P}$. Then the two matrices are combined to form the *measurement matrix*

$$W_{2F \times P} = \begin{bmatrix} X \\ \hline Y \end{bmatrix} \tag{3.55}$$

The rows are updated by subtracting from each value the mean of the values in the same row:

$$\begin{aligned} &\bar{x}_f = \frac{1}{P} \sum_{p=1}^{P} x_{fp} \ , \ \bar{y}_f = \frac{1}{P} \sum_{p=1}^{P} y_{fp} \\ &\tilde{x}_{fp} = x_{fp} - \bar{x}_f \ , \ \tilde{y}_{fp} = y_{fp} - \bar{y}_f \\ &\tilde{X}_{F \times P} = \left[\tilde{x}_{fp}\right], \ \tilde{Y}_{F \times P} = \left[\tilde{y}_{fp}\right] \\ &\tilde{W}_{2F \times P} = \begin{bmatrix} \tilde{X} \\ \hline \tilde{Y} \end{bmatrix} \end{aligned} \tag{3.56}$$

The matrix $\tilde{W}$ is called *registered measurement matrix.*

The world reference system is placed at the centroid of points $s_p, p = 1 \ldots P$ as shown in *Figure 3.14.*

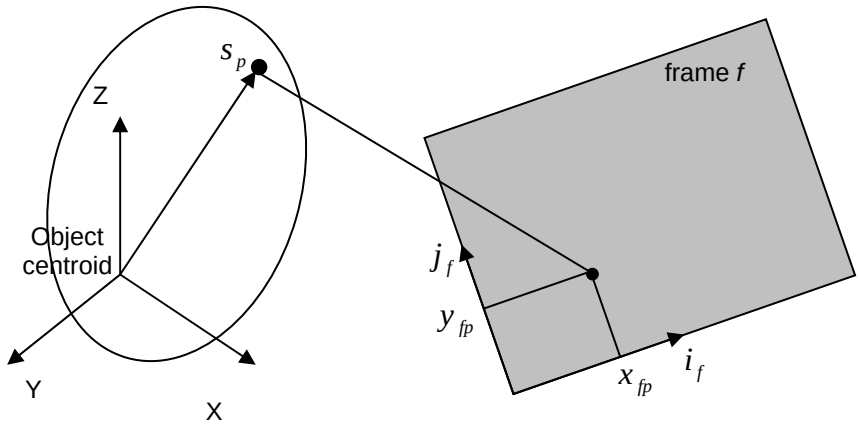


Figure 3.14 The world reference system placed at the object centroid

For a frame *f*, the camera reference system is determined by a pair of unit vectors i_f, j_f (given in the world reference) oriented in the direction of the lines and columns of the image. It can be shown that under orthographic projection

$$\tilde{x}_{fp} = i_f^T s_p \text{ and } \tilde{y}_{fp} = j_f^T s_p \tag{3.57}$$

Then the matrix $\tilde{W}$ can be written as the product of two matrices R and S

$$\tilde{W} = RS \tag{3.58}$$

where

$$R_{2F\times 3} = \begin{bmatrix} i_1^T \\ \vdots \\ i_F^T \\ j_1^T \\ \vdots \\ j_F^T \end{bmatrix}, S_{3\times P} = \begin{bmatrix} s_1 & \cdots & s_P \end{bmatrix} \tag{3.59}$$

The matrix R encodes the camera orientations (on rows), and the matrix S encodes the coordinates of the points $s_p, p = 1 \ldots P$ (when the world reference is their centroid).
The matrix $\tilde{W}$ is factorized through singular value decomposition in

$$\tilde{W} = U \sum V^T \tag{3.60}$$

If $\tilde{U}$ and $\tilde{V}$ are the first three columns of the matrices U and V respectively and $\tilde{\sum}$ is the top left 3x3 sub-matrix of $\sum$, then it can be shown that the least square solution fit is:

$$\hat{R} = \tilde{U}\sqrt{\tilde{\sum}} \text{ and } \hat{S} = \sqrt{\tilde{\sum}}\tilde{V}^T \tag{3.61}$$

This solution is not unique since for any invertible 3x3 matrix Q it holds that $\hat{R}Q$ and $Q^{-1}\hat{S}$ are also a valid decomposition. Moreover, $\hat{R}$ and $\hat{S}$ are in general different of the true solution. In general the matrix Q is found by imposing the rows of $\hat{R}Q$ to be as orthonormal as possible. Then the solution becomes:

$$R = \hat{R}Q \text{ and } S = Q^{-1}\hat{S} \tag{3.62}$$

Many other factorization algorithms were proposed in literature, and they all relay one some kind of linearization of the camera model. For a deeper understanding of the subject the reader is referred to [26, 27, 28].

3.13 The proposed structure from motion algorithm

Putting all the pieces together, a general and simple structure from motion algorithm can be summarized. It is assumed that the images are captured with a calibrated camera (with known intrinsic parameters). Camera calibration is briefly discussed in the next section. More complex SFM algorithms can perform the reconstruction task with uncalibrated cameras (see for example [2]). The idea here is to keep the algorithm as simple as possible, as the scope of this thesis is only to see if SFM can be applied to the reconstruction of the ear canal, and not to propose a certain method for doing that.

The main steps of the algorithm are:

1. Detect and track features points in all the images
2. Assuming calibrated camera, transform the coordinates of the feature points in normal coordinates
3. Find an initial guess for structure and motion using the factorization algorithm
4. Refine the structure and motion with bundle adjustment

The detection and tracking of feature points is discussed in *Chapter 4.*
The above algorithm was used for the experiments performed in *Chapter 5***.**

3.14 Camera calibration

Camera calibration refers to the estimation of the intrinsic and/or extrinsic parameters of the camera. The intrinsic parameters correspond to the physical camera parameters (focal length, principal point, skewness), while the extrinsic ones describes the position and orientation of the camera in the world coordinate system. In practice, these parameters can be estimated by knowing the correspondence between a set of 3D points and their projections in the images, and solving a system of linear equation based on the camera model.
Many calibration techniques exist. A common approach for most of them is to use a calibration object with known geometry, and then to compute calibration parameters from a set of images of this calibration object. For example, in [31] the calibration object is a cube with circular patterns on its sides, while in [32, 33] the calibration object is a planar pattern of squares.

One of the problems with the perspective camera model is that it only approximates the image formation process for a real camera. In general this model is good enough to describe most of the cameras. However, the optics of the real cameras can be very complex, and in many cases the light rays have to go through multiple lenses in order to form the image. In this situation the light rays coming from a certain point of the scene do not project to the prescribed geometric position [2]. If in ideal case the light rays pass through the optical center linearly, in reality the optics introduce a non linear distortion to the optical path [31]. When a high accuracy is required, the camera model has to consider these distortions induced by the camera optics, and to correct them.

The most commonly used corrections are for radial distortions and decentering distortions [34]. The radial distortion is the displacement of the image points radially in the image plane, relative to the center of the image. The decentering distortion appears when the centers of curvature of the lenses are not collinear, and has both a radial and a tangential component [35].

Thus, a point in the image plane undergoes further transformations.

Following notations are made:

$\begin{bmatrix} u_0 & v_0 \end{bmatrix}^T$ is the principal point;
$\begin{bmatrix} u_d & v_d \end{bmatrix}^T$ are the distorted coordinates;
$\begin{bmatrix} u_c & v_c \end{bmatrix}$ are the corrected coordinates.

The radial distortion is:

$$\begin{bmatrix} \delta u^r \\ \delta v^r \end{bmatrix} = \begin{bmatrix} \bar{u}_d \left(k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6 + \cdots\right) \\ \bar{v}_d \left(k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6 + \cdots\right) \end{bmatrix} \tag{3.63}$$

The tangential distortion is:

$$\begin{bmatrix} \delta u^t \\ \delta v^t \end{bmatrix} = \begin{bmatrix} \left(2 p_1 \bar{u}_d \bar{v}_d + p_2 \left(r_d^2 + 2\bar{u}_d^2\right)\right)\left(1 + p_3 r_d^2 + \cdots\right) \\ \left(p_1 \left(r_d^2 + 2\bar{v}_d^2\right) + 2 p_2 \bar{u}_d \bar{v}_d\right)\left(1 + p_3 r_d^2 + \cdots\right) \end{bmatrix} \tag{3.64}$$

where

$$\bar{u}_d = u_d - u_0 \; \bar{v}_d = v_d - v_0 \; r_d = \sqrt{\bar{u}_d^2 + \bar{v}_d^2} \tag{3.65}$$

$k_1, k_2, \cdots$ are radial distortion coefficients and $p_1, p_2, \cdots$ are tangential distortion parameters.

Then $$\begin{bmatrix} u_c \\ v_c \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} \bar{u} + \delta u^{(r)} + \delta u^{(t)} \\ \bar{v} + \delta v^{(r)} + \delta v^{(t)} \end{bmatrix} \tag{3.66}$$

In practice, only the first two or three radial distortion coefficients and the first two tangential distortion coefficients are estimated.

Most of the calibration applications take into account these distortions and evaluate the distortions coefficients together with the other camera parameters.

The calibration procedure using planar calibration patterns became very popular being easy to use and accurate in the same time, and it is supported by many software libraries. Probably the most widely used are Matlab Camera Calibration Toolbox [70] and Intel OpenCV library [71]. Both of them use a checkerboard planar pattern. While the Matlab Toolbox seems to be a little bit less appealing because it needs a lot of user interaction in the corner extraction process, the OpenCV calibration method is completely automatic, once the user provided a set of images of the calibration pattern.

A large amount of work was dedicated to the calibration of endoscopic cameras [36-43]. Characteristic to rod lenses of the endoscopic cameras is their wide field of view. This introduces in general significant radial distortions and they have to be considered in all the applications based on images taken with endoscopic cameras.

CHAPTER 4

Features detection and tracking

It was shown in the previous chapters that finding correspondences between different images is the key point in many computer vision applications, and recovering the 3D structure of the scene and the camera motion in only one of them. Other applications are camera calibration, image registration, object recognition, robot navigation, indexing and database retrieval, to remember only a few. It is clear then that results produced by these applications cannot be better than the matching methods themselves. The process of relating images of the same scene it is known as the feature detection and matching. Feature detection refers to finding interest regions in images, while feature matching refers to the process of relating these regions in different images. When the images represent the frames of a video sequence, the matching process is referred as feature tracking.

As the feature detection is the first step in so many computer vision applications, a very large number of feature detection algorithms have been developed in time. Relating all the points in images can be very difficult/impossible and computational expensive task. As a result, in practice only a relative small number of correspondences are detected in different images. Thus the reconstruction is rather a sparse set of 3D points. In some cases this is however not sufficient to reconstruct full surface models. For

specific applications a dense model can be obtained by 3D interpolation. There are also methods to obtain dense reconstructions starting from a small set of correspondences (see for example [ch3-2]).

4.1 Definition of a feature

There is no an exact definition of a feature. Rather this notion is general and is related to a certain application. Then a feature can represent any kind of information that can be extracted from an image and it is relevant for solving the given task. In general, features can encapsulate the result of a general neighborhood operation applied to an image, or can represent certain structures present in the image, like points, edges, or connected regions.

A feature has to be:

- Localized
- Meaningful
- Detectable

An important property of the features is their *repeatability rate* and it refers to the percentage of the correspondences found in different images. In order to be "good", a feature has to be distinctive enough to be detected in more than one image. Another property of a feature is its computational complexity. This is important, for example, in real time applications, where there is a time constraint and, as a consequence, the features have to be easy to extract. In some cases, the computational complexity of features can limit the detection process only to certain regions in the images.

Features should be reasonably tolerant to a certain level of noise in image. As the illumination conditions can change in different images, they should be invariant to these changes. Ideally the features have to be also invariant to scaling, rotation and perspective distortions.

Feature detection is a low level image processing operation. When the feature detection is related to a neighborhood operation, then a local decision is made at every image point, weather there is or not a certain type of feature.

A plus of robustness may be added in some applications if two or more different types of features are extracted from the images.

4.2 Types of features

Features extracted from images can be

- **Corners / interest points:** In the literature, the notions of corner and interest point are often used interchangeable. In a traditional way, a corner is defined by the intersection of two edges, or a point with two different edge directions in its neighborhood. An interest point, on the other side, can be any point with a well-defined position in the image and that can be robustly detected. The early algorithm performed corner detection by finding edges in a first step and then looking for the points on the edges with rapid changes in direction. Modern algorithms do not relay anymore on the edge detection, for example the corners can be detected by looking for high levels of curvature in image gradients. Other points than traditional corners can be detected by this kind of methods. They are interest points, even if by tradition they are still named corners.
- **Edges.** An edge is defined as a boundary between two regions in the image. In practice edges are detected as sets of points with high gradient magnitude.
- **Blobs** (also called regions of interest): The notion of blob refers to a meaningful region in the image, and sometimes the blobs correspond to objects in the image. Usually blobs are associated to a certain point (e.g. the center of mass or the local maximum of an operator response).
- **Ridges**: are used to describe elongated objects, and they occur normally along the center of such objects. Ridges can be considered a generalization of the medial axis. In general it is more difficult to extract these features than corners, lines or blobs.

Once a feature has been detected, an image patch (neighborhood) around the feature can be used to define some attributes or feature descriptors, forming a so called feature vector. The feature detection step itself can provide some of these attributes, e.g. the magnitude of the gradient. The descriptor has to be distinctive, robust to noise, and geometric and photometric deformations. The matching is often based on a distance between the feature vectors, e.g. the Mahalanobis or Euclidean distance. The dimension of the descriptor has a direct impact on the time this task takes.

When talking about feature detectors and their applicability in solving different problems, one should consider the possible transformations of the images that

can occur in a sequence of images. The transformations can be either geometrical or photometrical. The geometrical transformations can be: rotation, similarity (rotation and uniform scale), and affine (rotation and non-uniform scale, e.g. the perspective effect). Photometrical transformations refer to the illumination conditions that can occur from one image to another.

Selecting a feature detector depends not only on the content of the images but also on the way the images are captured. For example, in a sequence of images captured with a video camera, it is not very probable that the image changes too much from one frame to another. In this case the scale and perspective effects may be not considered. On the other hand, if the images are taken separately from very different points of view, then the shape of the features can significantly change, and the affine transformations should be taken into account. Affine invariant feature detectors are in general complex and may require high computational costs. Sometimes, selecting features that are only rotational and scale invariant offer a good compromise.

4.3 Comparing image regions

In some cases, the feature matching process requires to compare image regions. This is typically done using two measures: dissimilarity based on sum-of-square-differences (SSD) and similarity based on normalized cross-correlation (NCC) [2].

For a region W in image I and corresponding region $T(W)$ in image J, the dissimilarity is defined as:

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

with $w(x, y)$ is a weighting function typically constant and equal to 1, or a Gaussian function to emphasize the central area of the region.

The *similarity* is:

$$S = \frac{\iint_W \left(J\left(T\left(x, y\right)\right) - \bar{J}\right) \cdot \left(I\left(x, y\right) - \bar{I}\right) w(x, y)dxdy}{\sqrt{\iint_W \left(J\left(T\left(x, y\right)\right) - \bar{J}\right)dxdy} \cdot \sqrt{\iint_W \left(I\left(x, y\right) - \bar{I}\right) w(x, y)dxdy}} \tag{4.2}$$

where

$$\bar{J} = \iint_W J(T(x,y))dxdy \text{ , and}$$
$$\bar{I} = \iint_W I(x,y)dxdy \qquad (4.3)$$

are the mean intensity in the regions. Subtracting the mean intensity at every location of image regions makes this measure to be invariant to intensity changes.

4.4 Harris corner detector

One of the most widely used interest point detector is Harris corner detector [45]. The basic idea is simple: if we consider a small window centered on the point, then shifting the window in any direction should results in large changes in intensity. Anyway, there is no change in the case of a smooth region and, if the point is located on an edge, there is no change along the edge direction.

For a shift [*u,v*] of the window, the change in intensity can be expressed by the dissimilarity as:

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2 \qquad (4.4)$$

where *w* is a weighting function normally chosen *w=1,* or Gaussian function to emphasize the center of the window.

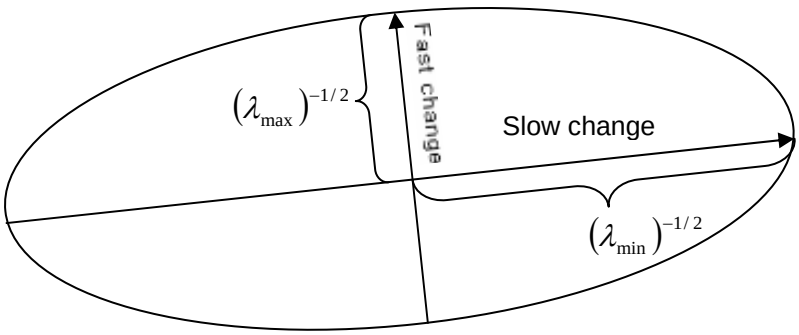


Figure 4.1 Two principal signal changes in the vicinity of a point.

When the shift is small, the equation (4.4) can be approximated by:

$$E(u,v) \cong \begin{bmatrix} u & v \end{bmatrix} \cdot M \cdot \begin{bmatrix} u \\ v \end{bmatrix} \tag{4.5}$$

where $M_{2\times 2}$ is called the second-moment matrix (or auto-correlation matrix) and it is computed from image derivatives:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{4.6}$$

The point is considered a feature or not depending on the eigenvalues of the matrix *M*. The eigenvalues correspond to the two principal signal changes in the neighborhood of the point. If $E(u,v) = 1$ is the ellipse centered on the point, then the eigenvectors give the orientation of the ellipse, and the eigenvalues give the magnitude of ellipse axes, as shown in *Figure 4.1*.

If both eigenvalues are very small, then there is no feature at the considered point. If one of them is very small and the other one is a large positive value, then an edge was found. Large, distinct positive values correspond to a corner.

In practice it is often desirable to avoid the complexity associated with the computation of the eigenvalues of the matrix *M*. Then the corner response function is defined as:

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det M - k(traceM)^2 \tag{4.7}$$

where $k = 0.04$.

The response function depends only on the eigenvalues of the matrix M.

The corner response function is a large positive for a corner, a large negative for an edge, and close to zero for smooth regions.

The corners can be detected by fixing a threshold for the response function $R > threshold$, and finding local maxima.

Harris corner detector is invariant to rotation (the shape of the ellipse depends only on the eigenvalues), and it is also invariant to changes in intensities since it is based on derivatives. Anyway, this detector is not invariant to image scale.

4.5 The KLT tracker

The KLT (Kanade-Lucas-Tomasi) tracker [57, 58, 64] is one of the most popular tracking algorithms. For a sequence of images, the algorithm detects in a first stage a set of interest points and then, each of them being then tracked in the next frames.

One of the principle this method is based on is to find features that are optimal for tracking. The detection stage is identical with Harris corner detector (based on the eigenvalues of the second moment matrix), but a different criteria is used to select features.
The authors suggest that features selected with $\min(\lambda_1, \lambda_2) > threshold$ are the ones that can be tracked well. The algorithm is able to determine an affine transformation that maps the neighborhood of a feature point in one image to the corresponding one in the next image by minimizing the dissimilarity between these image regions (see equation 4.1). When the distance between frames is small, the transformation is a simple translation.

When a feature is lost, (couldn't be tracked anymore), the algorithm is able to find a new one, in order to keep constant the number of tracked features. To keep the algorithm fast there is a limit for the maximum allowed displacement between the frames.

4.6 Scale invariant feature detectors

Sometimes, a desirable property for a feature detector is to be invariant to scale changes in images. The basic idea in finding a scale invariant feature point is to define a function on a variable size region around the feature point (e.g. the average intensity) and to find the local maximum of this function. The region size found in this way is invariant to scale changes. Then the feature is described together with its characteristic scale.

The concept of automatic scale selection was proposed by [52]. The scale space of an image is obtained by convolving the image with Gaussian kernels at different scales.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{4.8}$$

where *I*, is the image, *G* is the Gaussian kernel, and "*" is the convolution operator.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{4.9}$$

In [53] a scale invariant feature detector is introduced, namely *Harris-Laplace*. The feature points are detected with the Harris detector, and then corresponding scales are determined for each feature point using the Laplacian of Gaussian:

$$Lap(x, y, \sigma) = \sigma^2 \left(L_{xx}(x, y, \sigma) + L_{yy}(x, y, \sigma) \right) \tag{4.10}$$

The *characteristic scale* is selected by searching for a local maximum of the Laplacian over scales. The authors showed that the characteristic scale is relatively independent of the image scale and that the ratio of the characteristic scales for two images is equal to the scale factor between the images.

In [49] a similar feature detector is presented, but this time the features are located using the determinant of the Hessian matrix.

$$H = H(x, y, \sigma_D) = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} L_{xx}(x, y, \sigma_D) & L_{xx}(x, y, \sigma_D) \\ L_x L_y(x, y, \sigma_D) & L_{yy}(x, y, \sigma_D) \end{bmatrix} \tag{4.11}$$

Due to the second derivatives in the Hessian matrix, this detector gives strong responses for blobs and ridges. The shape of an elliptical region is determined with the second moment matrix of the intensity gradient, and the scale with Laplacian. This detector is known as *Hessian-Laplace* feature detector.

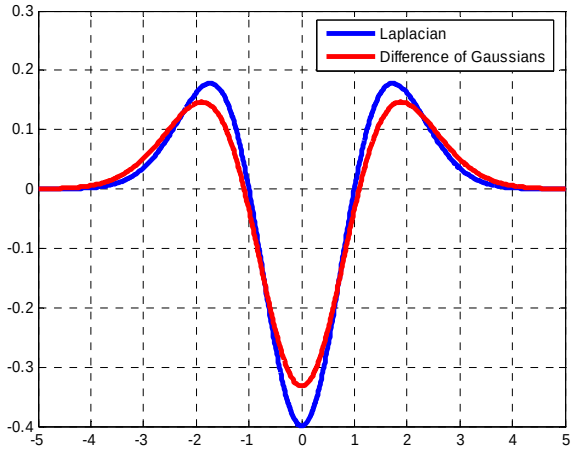


Figure 4.2 Comparing one dimensional Laplacian and Difference of Gaussians

In [54] the feature points and their associated scales are found by looking for local maximum of difference of Gaussian in scale and space:

$$DoG(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{4.12}$$

The difference of Gaussians approximates the Laplacian (see *Figure 4.2*), and can be computed more efficiently.

$$L(x, y, k\sigma) - L(x, y, \sigma) \approx (k-1)\sigma^2 \left(L_{xx}(x, y, \sigma) + L_{yy}(x, y, \sigma)\right) \tag{4.13}$$

This approximation is compensated by the gained speed of detection process. The detector is called SIFT (Scale Invariant Feature Transform) and it comes together with a very efficient feature descriptor. The combination of SIFT feature detector and descriptor is widely used in many computer vision problems.

In [51] it is shown that the features detected with *Harris-Laplace* detector are more repeatable than other detectors.

Another scale and rotation invariant interest point detector and descriptor was recently introduced in [46]. The detector is called SURF (Speeded Up Robust Features) and it is based on a measure of the Hessian matrix, and a distribution-based descriptor. As the name may suggest, the detector is focused on speed.

According to the experiments performed by the author, the performance of detector/descriptor is very similar and even outperforms many of the previosly proposed detectors/descriptors with respect to repeatability, distinctiveness, and robustness.

4.7 Affine invariant region detectors

The affine-invariant feature detectors deal with view-point changes in different images. Several affine-invariant region detectors have been proposed in literature.

In [51, 49], two affine invariant region detectors are constructed on the top of *Harris-Laplace* and *Hessian-Laplace* detectors. The shape of the affine region is determined with the second moment matrix and then it is normalized to a circular one.

In [44] two affine invariant region detectors are presented. The first one is geometry-based. Some anchor points are detected in images with Harris corner detector [45], and also edges close to the anchor point are extracted with Canny edge detector [47]. Two points are moved along the edges until they reach a position where some photometric measures of the parallelogram region spanned by them together with the anchor point go through an extremum (see *Figure 4.3*).

In the second method the anchor points are local intensity extrema. An intensity function along the rays emanating from this point is evaluated (see *Figure 4.4*). Local extrema of this function give the points of the region border.

The function evaluated along the rays is:

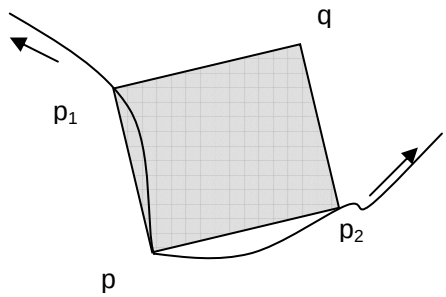


Figure 4.3. Edge based region detector

$$f_I(t) = \frac{abs\left(I(t) - I_0\right)}{\max\left(\frac{\int_0^t abs(I(t) - I_0)dt}{t}, d\right)} \tag{4.14}$$

where I_0 is the intensity at the extremum, *t* is a parameter along the ray, *I(t)* is intensity at the position *t*, *d* is just a small number to avoid division by zero.

The regions detected with this method can have arbitrary shape, but they are approximated with ellipses. If *f* is the characteristic function of a region (1 inside, 0 outside), the geometric moments corresponding to a region *R* are:

$$m_{pq} = \iint_{x,y \in R} x^p y^q f(x,y)dxdy \tag{4.15}$$

The geometric moments up to the second order are computed for the regions. Then the ellipse that approximates the region has the same geometrical moments as the original regions.

Another affine invariant region detector is the salient region detector proposed in [48], which maximizes the entropy within elliptical regions centered on a point.

A different approach was used to develop the Maximally Stable Extremal Region (MSER) detector [50] witch performed very well in comparative studies [49]. The detected regions are connected components having the property that all the pixels inside the region are either brighter or darker than

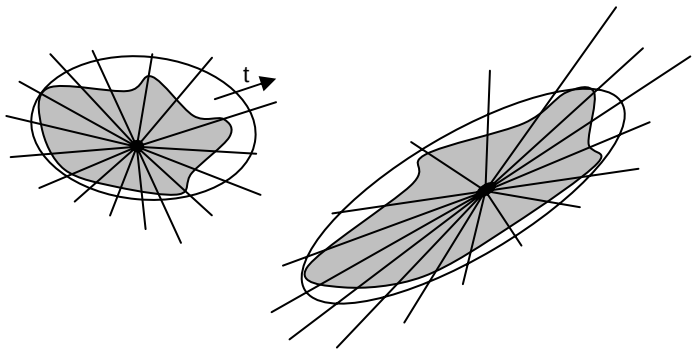


Figure 4.4 Intensity based region detector

the pixels outside the boundary. The regions are detected by optimizing the threshold selection process. A *maximally stable* region corresponds to a threshold for which the relative area function of relative change of threshold is at a local minimum.

4.8 Feature Descriptors

A large number of features descriptors have been proposed, like shape context [59], steerable filters [60], moment invariants [61], spin images [62], SIFT [54], differential invariants [63], to remember only a few of them. Choosing a proper feature descriptor depends in general on the feature detector itself. It is important that descriptors are distinctive and in the same time invariant to different image transformations. For example, the Harris corner response is invariant to rotation and photometrical changes.

In comparative studies [55, 56] the SIFT descriptor [54] proved to be superior to all the other descriptors tested. Its capacity to be very distinctive, invariant to image rotation, scale, intensity change, and to moderate affine transformations, made it to be the most widely used descriptor. The descriptor computes gradient orientation histograms for several small windows around the interest point stored in 128 elements vector, in this way being able to capture a large amount of information about the intensity patterns in the neighborhood of the point.

Its high dimensionality can be a small disadvantage when talking about feature matching, in application where the speed is very important.

It was also shown in [55] that the performance of the descriptors doesn't depend on the feature detector and, in general, region based descriptors seem to perform better than point-wise descriptors.

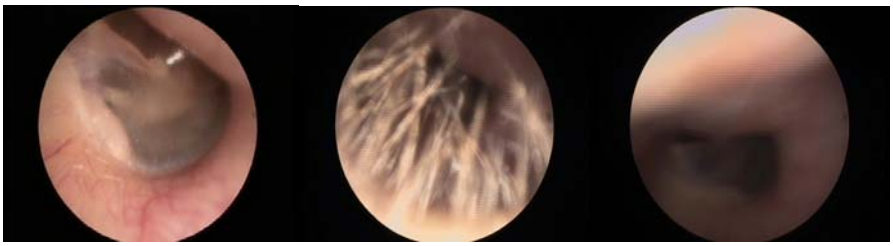


Figure 4.5 Otoscopic images: left-visible specular reflection; middle-hair blocking the field of view; right-bad illumination conditions

4.9 Features detection and tracking in otoscopic images

A simple visual inspection of the otoscopic images shows that a normal ear canal has a rather smooth and uniform colored surface. Excepting the region close to the ear drum where some very tiny blood vessels are visible (e.g *Figure 4.5-left*), the rest of the ear canal doesn't offer any visual clue. The light source placed on the tip of the otoscope points in the direction of view. This can result is specular reflections especially on the surface of ear drum, generating high intensity regions in the images that can be easily interpreted as features by the detectors. Sometimes, it is difficult to manipulate the otoscope in such a way that the tip is always in the middle of the canal. When the light source comes too close to the ear canal surface, the effect seen the images is an intensely illuminated region while the other parts become darker (*Figure 4.5-right*). Moreover, the outer region of the ear canal has hairs that can totally block the field of view (*Figure 4.5-middle*).

The otoscope is relatively big comparing to the ear canal. Being operated by hand makes it very sensitive to movements. The result is visible motion blur in many of the images. Together with an evident low contrast, all these unfavorable conditions make the otoscopic images a challenging input for the feature detection and tracking methods.

Despite of these discouraging observations, some experiments were performed in order to see how the feature detectors perform in relative "well" images. The original binaries or source codes provided by the authors of different detectors were used in these experiments.

Figure 4.6 Three frames from a sequence of 20 frames

Figure 4.7 Features detection with Harris corner detector, and tracking with KLT tracker

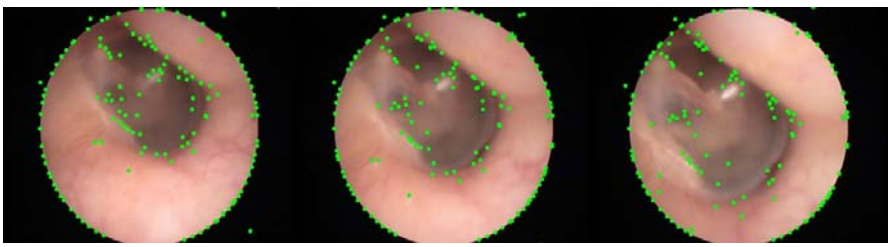


Figure 4.8 Corner selection with the standard KLT method (minimum eigenvalue)

In the first experiment the Harris corner detector was tested together with the KLT tracker (implementations available in OpenCV library) for a sequence of 20 frames, 720x576 pixels. In *Figure 4.6* frames 1, 10, and 20 are shown. The results are presented in *Figure 4.7*. Only a small number of "corners" are detected and tracked with this method, and most of them are placed (as expected) on the circular edge generated by the tip of otoscope. Other points are detected around the ear drum corresponding to some small structures on the ear drum. The natural curvature of the ear canal wall, close to the ear drum,

generates a false edge due to the view angle, and some points are detected in this region (upper-right region in the images in *Figure 4.7*). No feature is detected on the surface of the canal.

A very similar experiment is performed with the same sequence of images, but this time the features are selected with *minimum eigenvalue* method. The results are shown in *Figure 4.8.* A slightly larger number of points are detected this time, most of them in the same regions as in the previous experiment. A larger number of points appear around a high intensity spot generated by specular reflections.

The SIFT detector is tested with two similar images shown in *Figure 4.9.* The images are selected such as blood vessels structures are visible in both of them. A number of 188 key points are detected in the first image, and 150 in the second one.

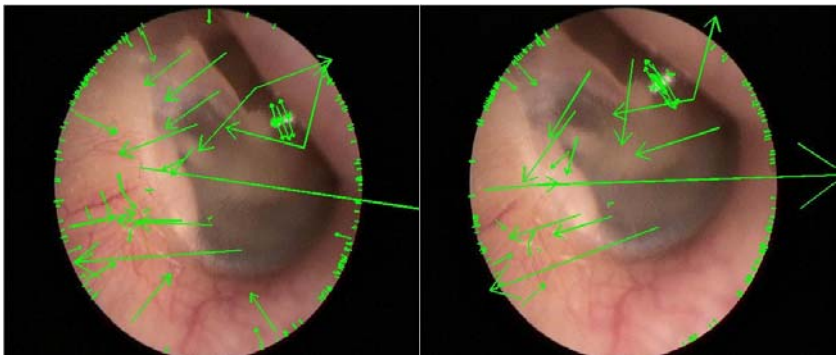


Figure 4.9 SIFT feature points detected in two different views

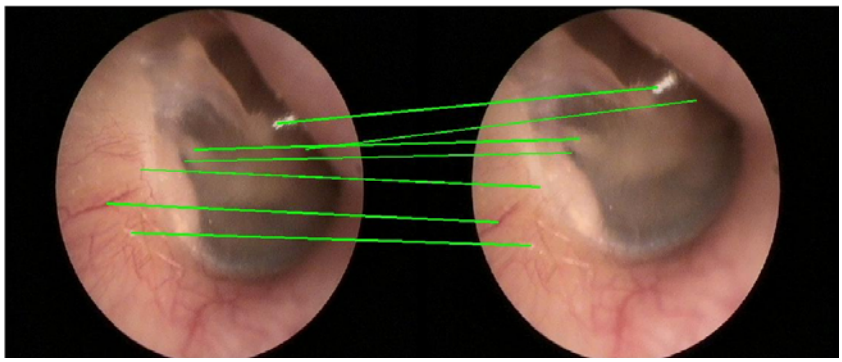


Figure 4.10 Matching SIFT features from Figure 4.9

Once again many key points are detected on the circular frame. Only 9 matches are found between these two images (*Figure 4.10*). Inspecting these matches, it can be observed that two of them correspond to small blood vessels, one of them to the specular spot, and the others correspond to some structures on the ear drum.

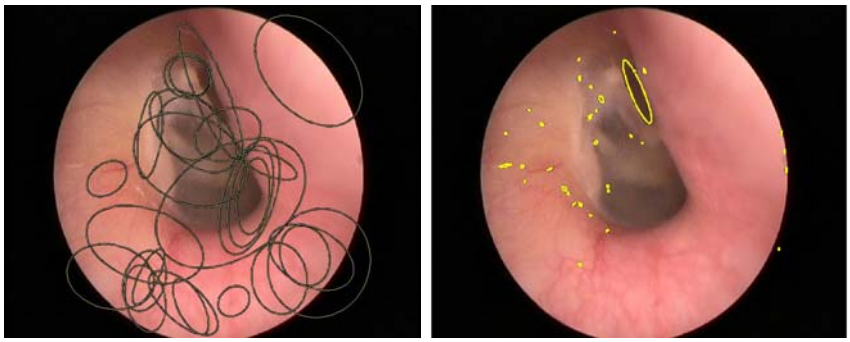


Figure 4.11 Intensity based feature detector (left); MSER detector (right)

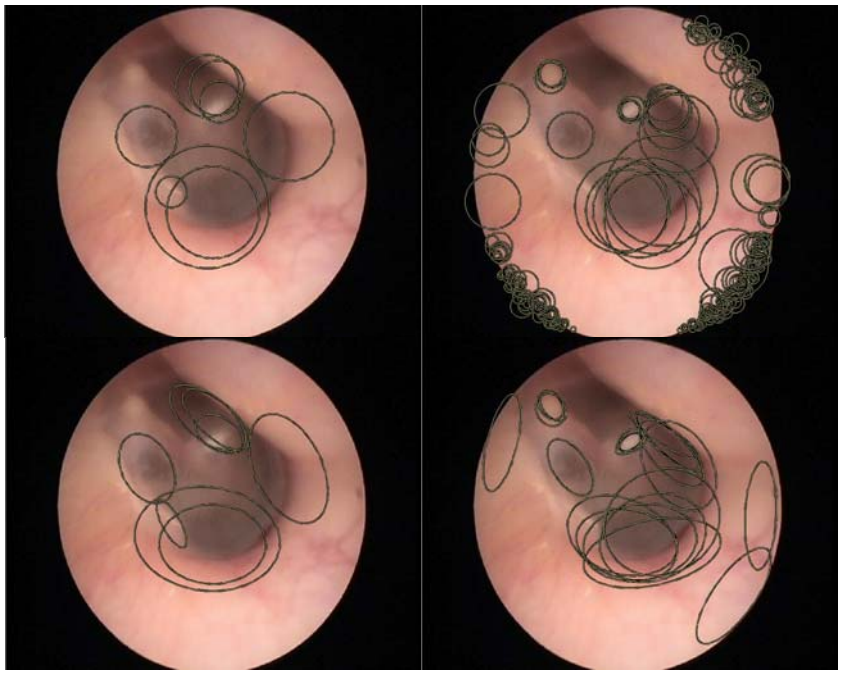


Figure 4.12 Haris-Laplace (top-left), Hessian-Laplace (top-right), Haris-Affine (bottom-left), Hessian-Affine (bottom-right) feature detector for the same test image

In *Figure 4.11* and *Figure 4.12* the results produced by other detectors are demonstrated: Intensity based detector, MSER, Harris-Laplace, Hessian-Laplace, Harris-Affine and Hessian-Afine. These results are very pure. Only the intensity based detector was able to identify more regions on the ear canal surface, but they are not well localized (large area of the ellipses), and they are generated more by the illumination variation in the images.

In the last experiment a number of artificial features (some black spots) were manually placed onto the surface of a silicon model of the ear. Three region detectors were tested in this case: Edge based detector, Intensity based detector and MSER. The results are in *Figure 4.13*. Remarkable is the large number of regions correctly detected by the intensity based detector and the precision of the MSER detector.

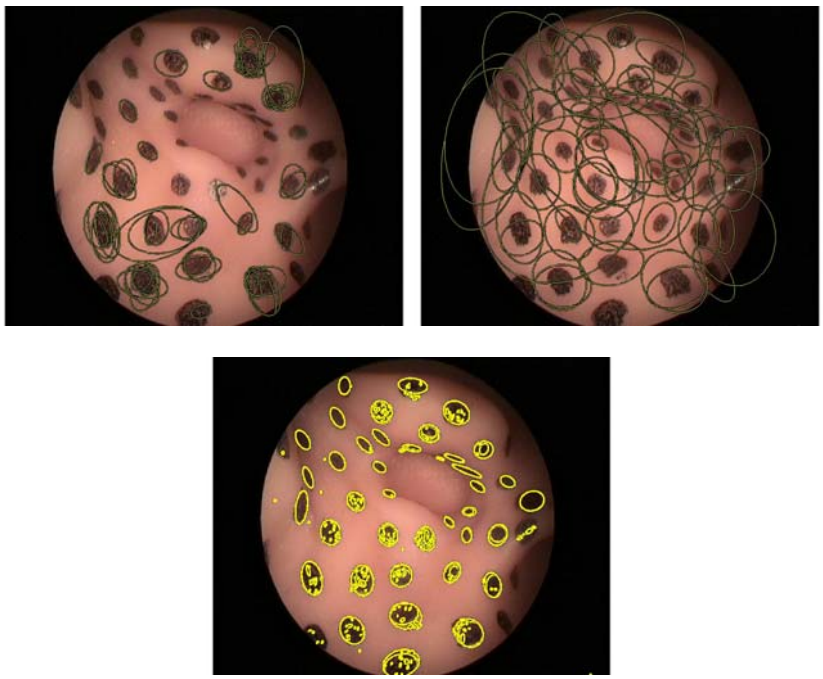


Figure 4.13 Edge based detector (left), intensity based detector (right), MSER (middle) tested for images of a silicon model of the external ear with manually added features

4.10 Discussion

It was shown that the otoscopic images don't provide valuable information for the feature detectors. A reconstruction of the ear canal cannot be possible as long as there are no features that can be detected and tracked in images provided by the video otoscope. Then it is obvious that a way to place artificial features inside the ear have to be found. Spraying some high contrast paint inside the ear canal can be such a solution. The features created in this way would be blob-like, and detectors like MSER or Intensity based one can successfully detect them. A combination of detectors can add a plus of robustness.

Another important issue is the presence of the hairs in the external part of the ear canal, blocking the field of view of the otoscope in a certain segment of the canal. Removing this hair is a must before taking the images.

Some image preprocessing techniques like contrast adjustment or color normalization could be of great help in improving the quality of images before performing feature detection and tracking step.

CHAPTER 5

Reconstruction accuracy of tube-like objects

In this chapter the reconstruction accuracy of tube-like structures will be measured. We would like to figure out if SFM methods are good enough for 3D reconstruction of this kind of objects. Several experiments with synthetic data are made. The general framework is similar for all the experiments: a number of points are placed over the surface of a known cylinder (radius, orientation), and several cameras are defined in such a way that they point inside the cylinder. The SFM method is the same for all the experiments: factorization method for obtaining an initial estimation of the model, and sparse bundle adjustment for optimization, as described in *Section 3.13*. As SFM can recover the model up to an arbitrary scale factor, the recovered model is first aligned with the real model using a 3D registration algorithm (scale adapted Iterative Closest Point). Then the accuracy of reconstruction can be measured as the registration error between two points data sets. A better way to estimate the reconstruction accuracy is to measure the radius of the recovered cylinder and to compare it with the real one. To do that, the problem of optimally fitting a cylinder to a set a 3D point has to be solved. Experiments with synthetic data are made in order to see how the reconstruction error is affected by the noise present in the images (localization accuracy of feature points), by the radius of cylinder, and by the number of feature points. In the end, an experiment with real data is presented.

5.1 Reconstruction problem validation

Before experimenting with real data, it is desirable to see if SFM methods are indeed appropriate for the given problem: reconstruction of a cylinder from a sequence of images taken from different positions of a camera pointing inside of the cylinder. At this step no quantitative analysis will be made, only the visual quality of the reconstruction will be inspected.

The experiment is performed using pure synthetic data. A number of 28 points are distributed over the surface of a cylinder with a radius of 40 units. The points form three rings at a distance of 20 units away of each other. The cylinder axis coincides with the *z* axis of the coordinate system. Two camera configurations are defined as in *Figure 5.1*. A camera is fully defined by its position in 3D space and its orientation. The direction along witch a camera is pointing is colored in red. In the first configuration (*Figure 5.1 a*), five cameras are placed along the cylinder axis. The first camera (in the bottom) is at a distance of 90 units away of the first ring of points. The other four cameras' positions and orientations are obtained by translating the previous camera with 10 units along the *z* axis.

In the second configuration (*Figure 5.1 b*) the first camera is placed at the origin of coordinate system, and for the other four cameras the positions and orientations are randomly generated.

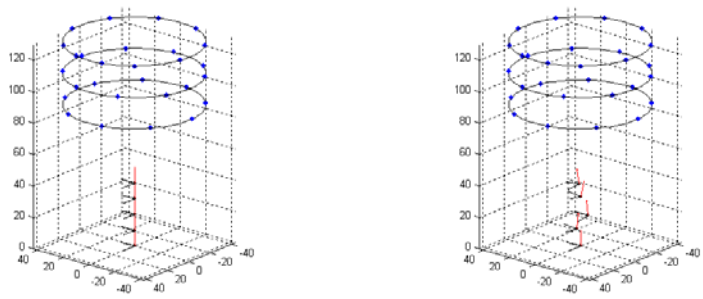


Figure 5.1 Two camera configurations used to test reconstruction of a cylinder using SFM methods

In the case of the first configuration, the camera motion is a pure translation along optical axis, while the second configuration corresponds to a general motion. The cameras are considerate to be calibrated. The reason of choosing this configuration for the cameras is that recovering the 3D model when camera motion is pure translation is ill posed for many of projective SFM methods. Of course, the factorization method used in these experiments recovers the Euclidean structure, but it is interesting to see how it behaves in this case.

The 3D points located over the surface of cylinder are projected on the frames of each of five cameras. The projection corresponds to image formation process in a real experiment, and projected points correspond to feature points. In real images feature detection may not be very accurate due to factors as image noise, or algorithm itself. To make our experiment more realistic, Gaussian noise with $\sigma = 0.005$ is added to the projected points in order to simulate the imprecision of feature detection step. The projected values corresponding to *x* axis of image frame range between (-0.4767, 0.6084), and corresponding to *y* axis range between (-0.5698, 0.5326). That means the Gaussian noise added to the projected points corresponds on average to a 0.45% localization error on both *x* and *y* axis. For example, in the case of a 512x512 pixels image, the localization error of features is 2.3 pixels on average.

The coordinates of noisy projected points are passed to the SFM algorithm and processed in the two steps. In the first step an initial Euclidean reconstruction (and also camera positions and orientations) is obtained with the factorization algorithm. It is already known that factorization method is not optimal due to the linearization of camera model. In the second step the recovered structure (and also cameras) is refined by a bundle adjustment process.

The results obtained for the two considered configurations are listed in *Figure 5.2* and *Figure 5.3*. A simple visual inspection of these results is more than enough to point out a few conclusions. Both experiments produced very similar results so we cannot conclude that a configuration behaved worse or better than the other. The reconstruction obtained with the factorization algorithm is quite bad qualitatively. While the top views show us that *x*, and *y* coordinates are estimated correctly (points follow the contour of the circle), the side views show us that the factorization method has a deficiency in the estimation of the depth information. In both cases the optimization performed by the bundle adjustment step corrected the errors and the recovered structure corresponds to the real one.

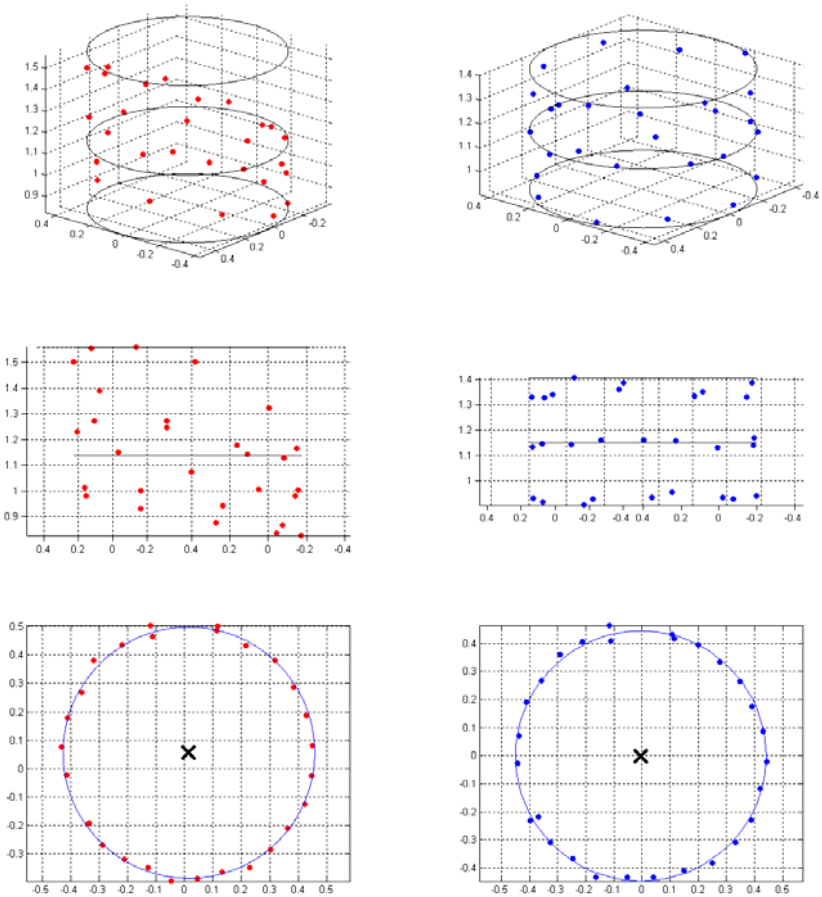


Figure 5.2. The recovered structure for the configuration of the cameras shown in Figure 5.1 a). Left column corresponds to the structure recovered after factorization method, the right column after bundle adjustment. Middle row is side view, while bottom row is top view of the recovered structures

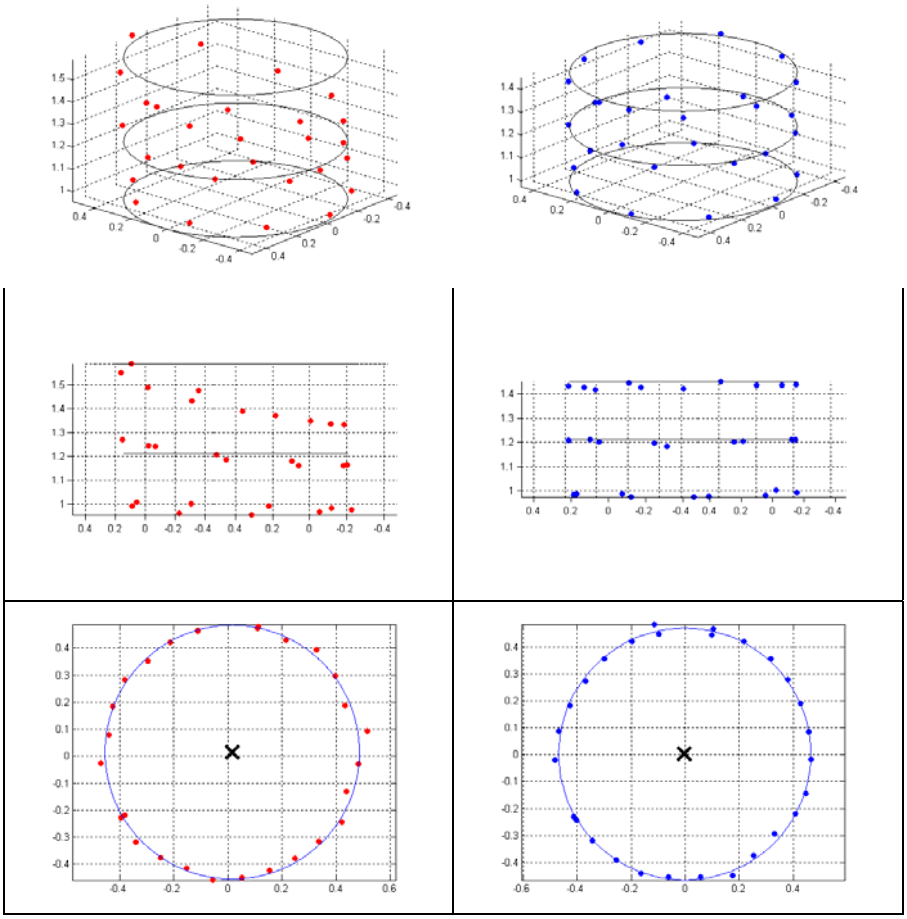


Figure 5.3. The recovered structure for the configuration of the cameras shown in Figure 5.1 b). Left column corresponds to the structure recovered after factorization method, the right column after bundle adjustment. Middle row is side view, while bottom row is top view of the recovered structures

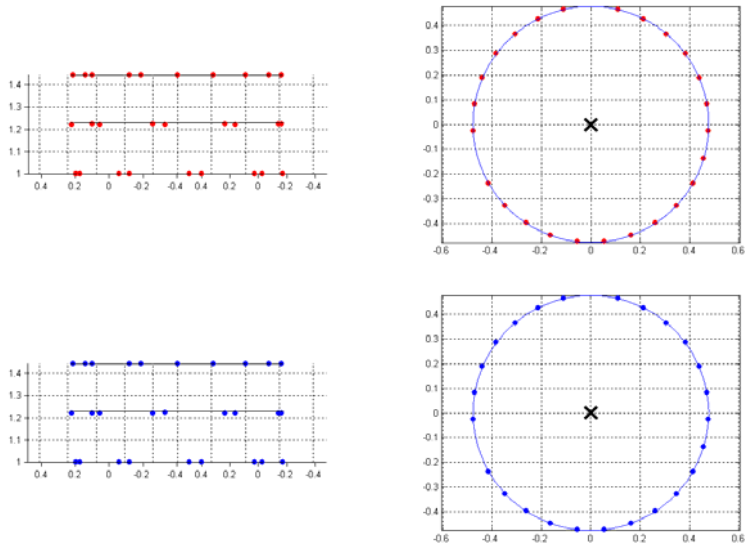


Figure 5.4. Side and top view of the reconstructed structure for the configuration in Figure 5.1 b) in the absence of noise

It is very clear from the side views that the recovered points form three groups corresponding to the three rings of the cylinder. The deviations reflect the Gaussian noise added to projections.

Figure 5.4 shows the side views and the top views of the reconstructed points in the case of second configuration in the absence of noise, before and after bundle adjustment. The reconstruction was almost perfect even after the factorization step; the small errors visible in the middle ring should be associated more with computational errors than with other causes. In this case the bundle adjustment couldn't improve the result, as it was already optimal after factorization step.

5.2 Registration of 3D point sets – ICP algorithm

The SFM algorithms recover the 3D structure of a scene only up to an arbitrary scale factor. In order to estimate the quality (accuracy) of the results, it is necessarily to register the points produced by SFM algorithm to the points of real structure.

The most popular algorithm used to register two 3D point sets is Iterative Closest Point (ICP), and there are many extensions and variations of the basic algorithm. All the algorithms require the two data sets to be roughly aligned otherwise there is a risk of converging to a local minimum. The original algorithm doesn't recover the scale factor between two data sets. The integration of the scale factor in the basic ICP algorithm is described in [65] and for convenience will be reviewed here.

The registration problem can be formulated in the following way: For two point sets A, B of 3D points corresponding to model points and data points, find a rotation matrix $R \in R^{3\times 3}$ and a translation vector $t \in R^3$ that optimally align the data points set to the model points set.

The ICP algorithm is an iterative one and can be summarized in the next steps:

Repeat until convergence

1. Find correspondences between points from the two data sets
2. Compute the rotation matrix and translation vector from the found correspondences.

The processing of finding correspondences between points is the most expensive in terms of processing resources. In the standard ICP for each point from the set of data points the corresponding one in the model set is found using nearest neighbor search. At the end of this process the set

$$C = \left\{ (i, j) \mid a_i \in A, b_j \in B \right\} \text{ is formed.}$$

The rotations and translation are obtained by minimizing the sum of squared distances between the corresponding points:

$$\left(R^*, t^*\right) = \arg\min_{R,t} \sum_{(i,j)\in C} \left\| b_j - Ra_i - t \right\|^2 \qquad (5.1)$$

If $\bar{a}$ and $\bar{b}$ are the centroids of the two data sets then the translation can be eliminated from the minimization problem by centering the points.

$$\bar{a} = \frac{1}{|C|}\sum_{(i,j)\in C} a_i \text{ and } \bar{b} = \frac{1}{|C|}\sum_{(i,j)\in C} b_i \tag{5.2}$$

Thus the optimum rotation can be computed as

$$R^* = \arg\min_R \sum_{(i,j)\in C} \left\| \left(b_j - \bar{b}\right) - R\left(a_i - \bar{a}\right) \right\|^2 \tag{5.3}$$

The solution of this problem is given by solving the singular value decomposition (SVD) of the matrix *K*:

$$K = \sum_{(i,j)\in C} \left(b_j - \bar{b}\right)\left(a_i - \bar{a}\right)^T = UDV^T \tag{5.4}$$

and

$$R^* = UV^T .$$

The translation vector t^* can be computed as

$$t^* = \bar{b} - R * \bar{a} .$$

5.2.1 Scale integration

In [65], rotation, translation, and scale factor are estimated simultaneously at each iteration. Integrating the scale factor, the minimization problem becomes:

$$\left(R^*, t^*, s^*\right) = \arg\min_{R,T,s} \sum_{(i,j)\in C} \left\| b_j - sRa_i - t \right\|^2 \tag{5.5}$$

The introduction of the scale factor in the problem doesn't affect the computation of rotation, as the matrix *K* will be the previous one multiplied with the scale factor, and the SVD problem remains the same.

Knowing R^*, the scale factor can be estimated as:

$$s^* = \arg\min_s \sum_{(i,j)\in C} \left\| \left(b_j - \bar{b}\right) - sR^* \left(a_i - \bar{a}\right) \right\|^2 \tag{5.6}$$

Making following notations

$$\tilde{b}_j = \left(b_j - \bar{b}\right) \text{ and } \tilde{a}_i = R^* \left(a_i - \bar{a}\right) \tag{5.7}$$

the scale factor becomes:

$$s^* = \sum_{(i,j)\in C} \tilde{b}_j^T \tilde{a}_i \Big/ \sum_{(i,j)\in C} \tilde{a}_i^T \tilde{a}_i \tag{5.8}$$

and

$$t^* = b - s^* R^* \bar{a}\,. \tag{5.9}$$

According to [65], computing the scale factor at each iteration, slightly increases the overall computation time, and also the number of iterations required to converge.

Convergence of the algorithm can be declared when the registration error is smaller than a given threshold, or a maximum number of iteration is performed.

5.2.2 Iterative Closest Point (ICP) algorithm test

An experiment is performed in order to test ICP algorithm. Two sets of points are generated and then passed to the ICP algorithm. The first points set (model points) consists of 100 points randomly distributed over the surface of a cylinder having a length of 100 units and a radius of 35 units. The cylinder center line coincides with the *x* axis of the coordinate system. The second points set (data points) is obtained from the first one applying a few transformations: translation with 25 units along *y* axis, rotation with $\pi/3$ around *z* axis, and then the points are scaled with a scale factor of 0.5. Finally Gaussian noise with $\sigma = 1$is added to the coordinates of the points. The two sets of points used in this experiment are presented in *Figure 5.5*, where the

points from the model data set have blue color, and the points from data set have red color. The ICP algorithm aligns the data points set to the model points set.

It was already mentioned that, in order to converge to the real minimum, the ICP algorithm needs the two datasets to be roughly aligned. From this reason, an additional step is performed before starting the main loop of the ICP algorithm. In this step, three pairs of control points with known correspondences are selected and used to compute initial rotation, translation and scale factor that roughly align the two sets of points. The control points are also randomly chosen and are represented by the green circles in F*igure 5.5.*

The results obtained after running the ICP algorithm are presented in *Figure 5.6 a) and b)*. The coarse alignment performed by the additional step is showed in *Figure 5.6 a)*, while the final (refined) result is showed in *Figure 5.6 b)*. A visual inspection of these results demonstrates that the algorithm converged to the optimal results, as it can be seen that the red circles are in general around the blue points. A quantitative evaluation of the accuracy of registration is given by the mean Euclidean distance between the final corresponding points, and its standard deviation. In the case of this experiment, the values are 1.6187 units for the mean distance, and 0.6620 units for standard deviation. These registration errors directly reflect the noise added to the points coordinates.

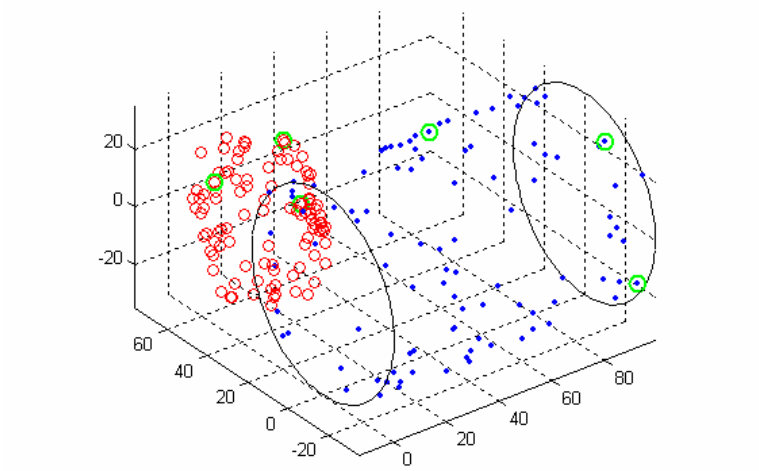


Figure 5.5 Model points set (blue), data points set (red), and control points (green) used to test the ICP algorithm

An experiment very similar with the one described above is performed, but this time in the absence of noise. Only geometric transformations are applied in order to construct the data points set. The final result obtained with the ICP algorithm can be seen in *Figure 5.7.* On average, the registration error in this case is 5.7026e-014 with a standard deviation of 2.2924e-014. The error can be fairly approximated to zero, as it is mostly generated by the computational errors.

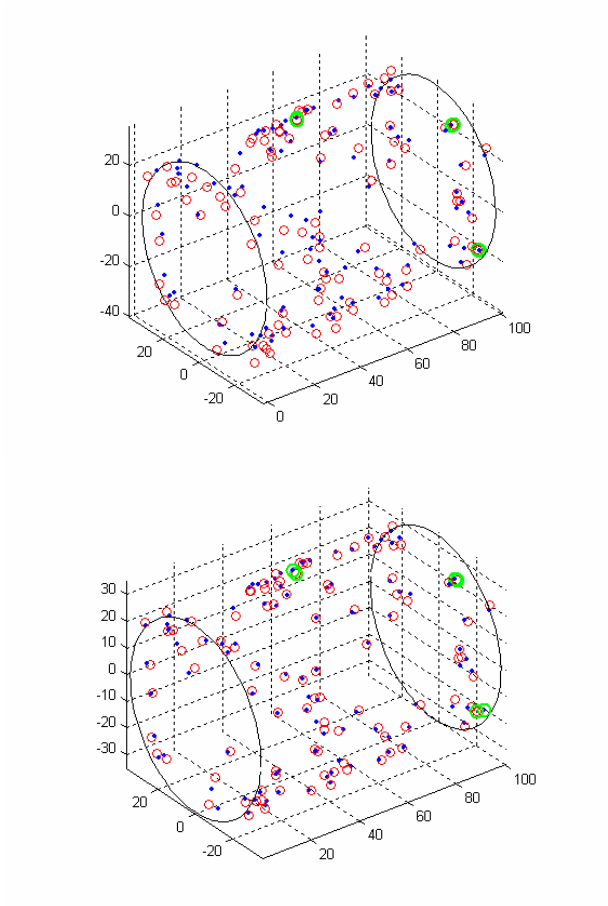


Fig. 5.6 Results obtained by the ICP algorithm: a) coarse alignment, b) refined alignment

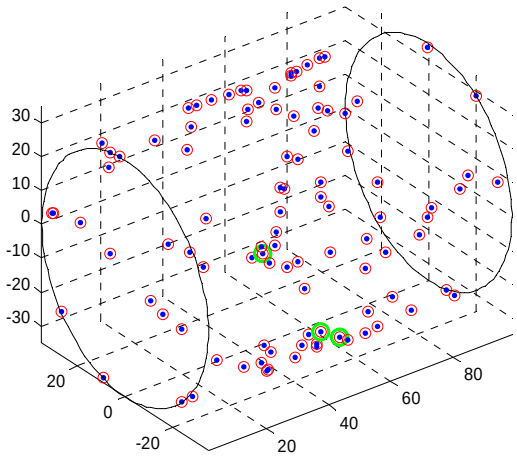


Figure 5.7 Alignment of two sets of points in the absence of noise

The registration step was required in order to measure the accuracy of the reconstruction using the proposed SFM method: factorization + bundle adjustment. Even if the average registration error can give a clue about the accuracy of the reconstruction, this measure is not the best choice for the specific problem of cylinder reconstruction. It is more interesting to see how accurately the reconstructed cylinder fits to the real cylinder. Due to the noise, the reconstructed points are not precisely placed over the surface of a cylinder. But a cylinder can be optimally fitted to the reconstructed points, averaging in this way the errors produced by individual points. Then the radiuses of both reconstructed and real cylinder can be used to define the reconstruction error as:

$$\mathrm{Re}\,constructionError = \left(1 - \frac{\mathrm{Re}\,constructedRadius}{\mathrm{Re}\,alRadius}\right) \cdot 100(\%) \tag{5.10}$$

5.3 Cylinder fitting algorithm

In this section an algorithm that fits a cylinder to a set of points is presented and analyzed.

A cylinder is completely defined by its radius, and the line that runs through its center, referred in the followings as cylinder's center line. The cylinder fitting problem can be solved by finding the center line and radius that minimize the sum of the squared distances between the set of points and the surface of the cylinder. Thus the fitting problem is performed in two steps: in the first step the central line is fitted to the points, and in the second step the optimal radius is computed. Once the center line is calculated, then the distance from a point to the cylinder surface can be obtained subtracting the radius from the distance from the point to the center line. Then the fitting problem becomes:

$$R^* = \arg\min_{R} \sum_{i=1,N} \left(dist(p_i, cl) - R\right)^2 \qquad (5.11)$$

where R^* denotes the optimal radius of the fitted cylinder, N is the number of the points, $\{p_i, i = 1, N\}$ is the set of points, and *cl* is the estimated central line of the cylinder.

In order to solve the above minimization problem, the central line has to be estimated. A line in 3D space is defined by a point on the line and a direction vector that specifies the direction of the line. As the line that optimally fits to a set of points passes through the centroid of the points, then only the direction of center line remains to be estimated.

If $$c_0 = \frac{1}{N} \sum_{i=1,N} p_i \qquad (5.12)$$

is the centroid of the points and d is the direction of the center line, then the equation of the center line is $c_0 + t * d$, where t is the parameter of the line.

The remaining unknown, direction of the center line, can be very elegantly computed using Principal Component Analysis (PCA). PCA is a technique that reduces multidimensional datasets to lower dimensions, retaining only those characteristics of the dataset that contribute most to its variance. To determine the principal components of a dataset, the eigenvectors and eigenvalues of the dataset covariance matrix have to be computed. The eigenvectors with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the dataset. For a deeper understanding of PCA technique, the reader is referred to [66, 67].

The direction vector of the cylinder's center line is given by the coefficients of the first principal component. The second and third PCs are orthogonal to the first, and their coefficients define directions that are perpendicular to the line.

Several experiments are performed in order to evaluate the performance a cylinder fitting algorithm. In the first experiment it is tested the way the number of the points influences the radius of fitted cylinder. A variable number of points are randomly placed over the surface of a cylinder having a radius of 35 units and a length of 250 units. The number of the points varies between 5 and 500. No noise was added to the points. To be noted that in this experiment the length is much bigger than the diameter of the cylinder.

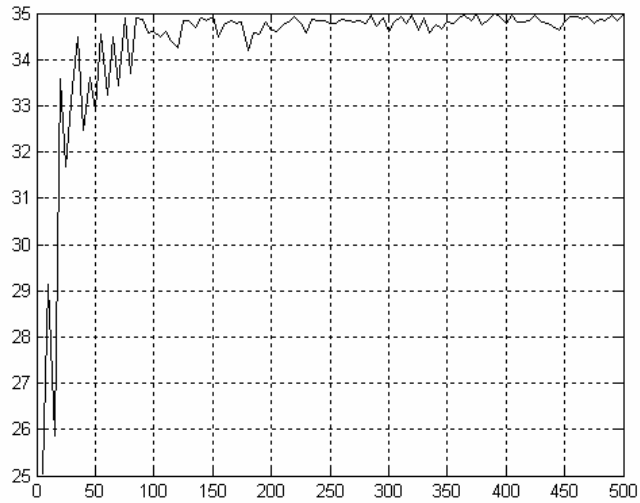


Figure 5.8 Fitted cylinder radius function of the number of points

The fitted cylinders radiuses are plotted against the number of the points in *Figure 5.8*. It is obvious that a small number of points results in a very bad estimation of the radius. This is due to the fact that the center line of the cylinder cannot be accurately estimated.

As the points are randomly distributed – and the same they are in a real configuration – there is no symmetry and centroid of the points doesn't coincide with the center of the real cylinder, and thus the estimated center line has also a direction different of the real one. These can be seen in the *Figure 5.9* – a closer look for 3 configurations corresponding to 10, 50, and 200 points. The computed radiuses of the fitted cylinders are in this case: 26.8964, 34.1071 and 34.5704.

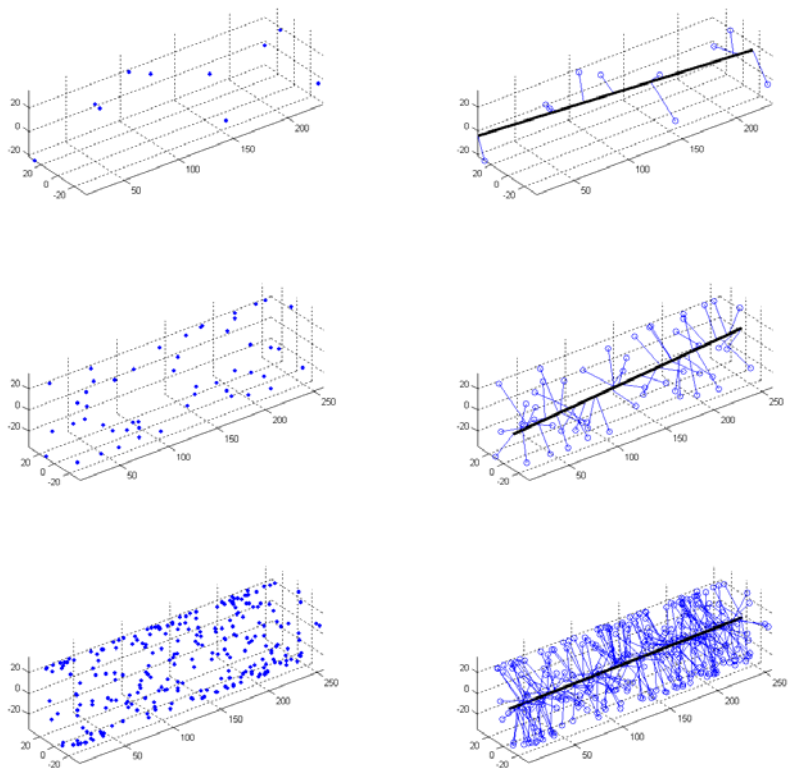


Figure 5.9 Left column: three configurations for 10, 50 and 200 points distributed over the surface of a cylinder. Right column: the center line of fitted cylinders

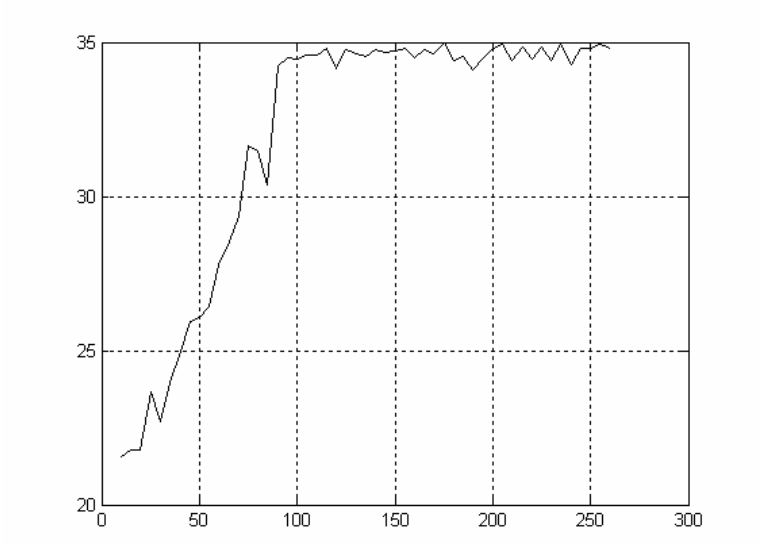


Figure 5.10 Fitted cylinder radius function of the number of the cylinder length

The second performed experiment shows the influence of the cylinder length on the fitted cylinder radius. A number of 200 points are randomly placed over the surface of a cylinder having a radius of 35 units and a length varying between 10 and 260 units. The results are depicted in *Figure 5.10*. It can be observed the estimation error of the radius is big when the length of the cylinder is smaller than its diameter. That happens because in that case the dominant direction of the variance is not anymore along the cylinder center line. This behavior can be seen in the *Figure 5.11* for three different lengths of the cylinder: 30, 70 and 90.

It was shown that not all the configurations are well tolerated by the cylinder fitting algorithm. The estimation accuracy of the fitted cylinder highly depends on the number of the points and the ratio of cylinder length and diameter. The symmetry of the points' distribution also affects the accuracy of estimation. For example, if a higher number of points are concentrated in the same area of the cylinder, they will influence more the position of the fitted center line. The cylinder fitting algorithm allows us to use the estimated radius as a measure of reconstruction accuracy. As it is, the algorithm doesn't perform well for many general configurations, and will generate additional errors when measuring the reconstruction accuracy. On the other side, only the center line of the cylinder cannot be accurately estimated. As the reconstructed points are aligned to the

model points before estimating the reconstruction accuracy, then the best practice would be to skip the line fitting step and to use the center line of the model. Thus only the radius of the reconstructed points needs to be estimated.

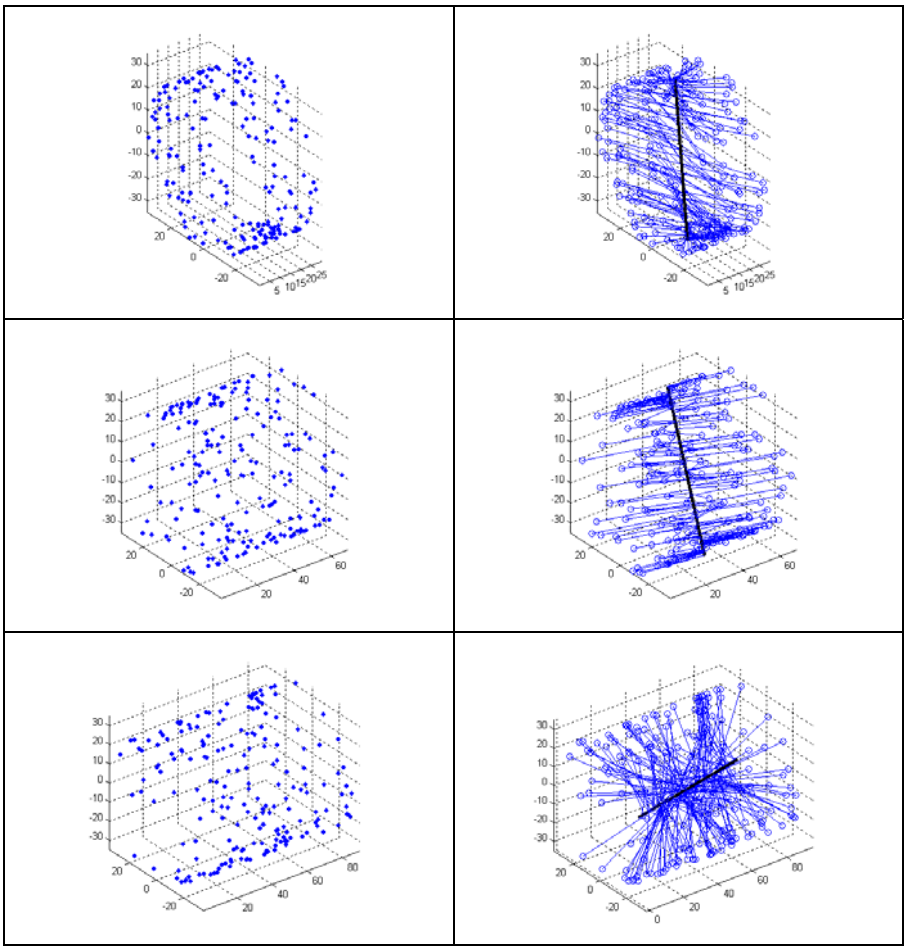
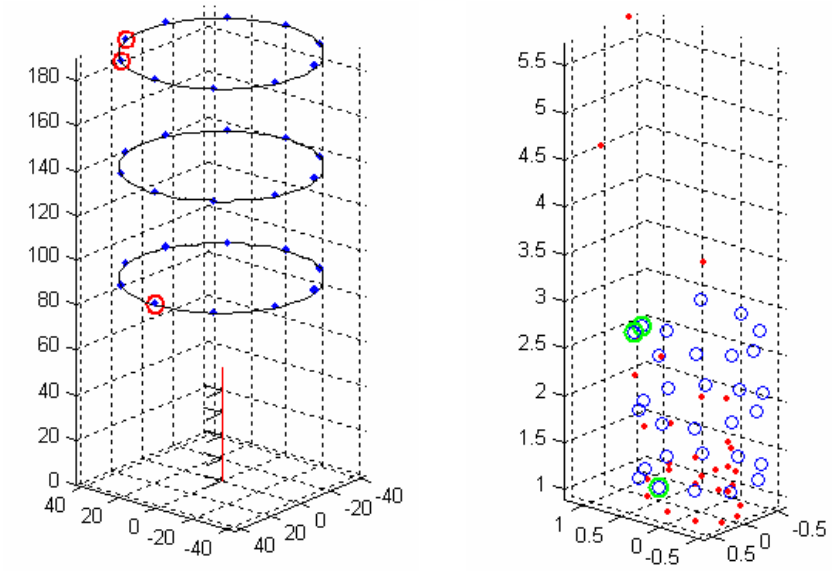


Figure 5.11 The fitted cylinder center line for three different lengths of the cylinder: 30, 70, 90.

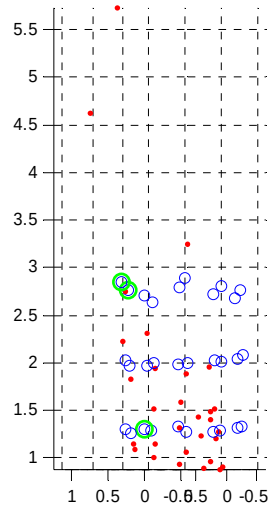
5.4 A complete reconstruction experiment using synthetic data

At this stage all the tools needed to reconstruct a cylinder and to measure the reconstruction accuracy are available. In this experiment a number of 30 points are distributed over the surface of a cylinder with a radius of 43 units and a length of 100 units. The points form 3 identical rings, each of them containing 10 equidistant points, and the distance between two consecutive rings is 50 units. Five camera frames are defined as shown in the *Figure 5.12 a).* Gaussian noise with $\sigma = 0.005$ is added to the projected points into the frame of each camera. This amount of noise corresponds to 0.53% localization error (on average) of the feature points in the image plane (or 2.75 pixels for a 512x512 image). The recovered structure after factorization method and after bundle adjustment step is shown if *Figure 5.12 b).*



a) *b)*

Figure 5.12 a) The configuration of the model points and camera frames. b) The recovered structure after factorization (red points) and after bundle adjustment (blue points)



a)

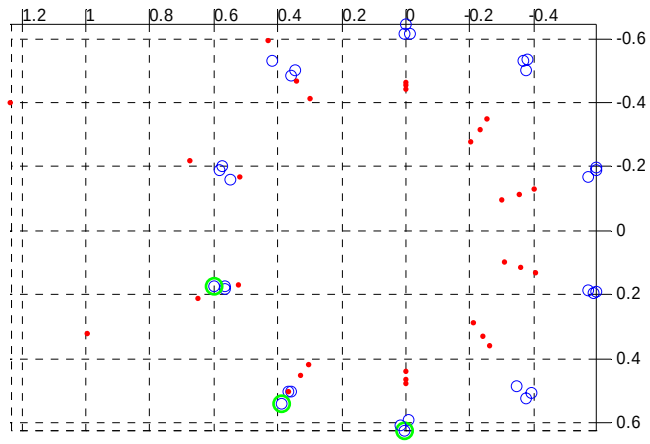
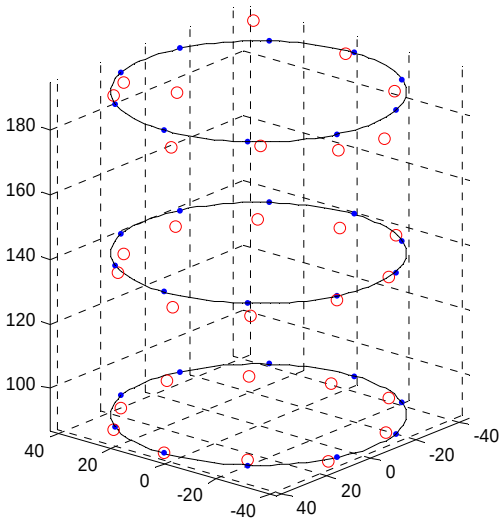
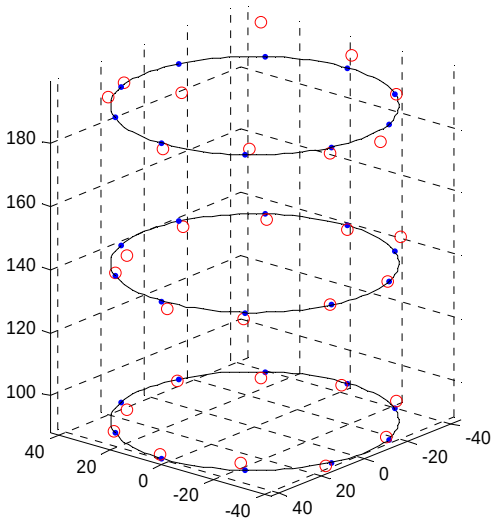


Figure 5.13. Side view and top view of the recovered structure after factorization (red points) and after bundle adjustment (blue points)

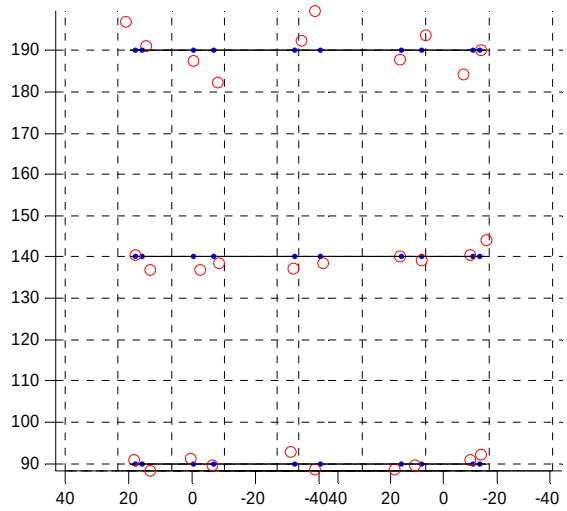


a)

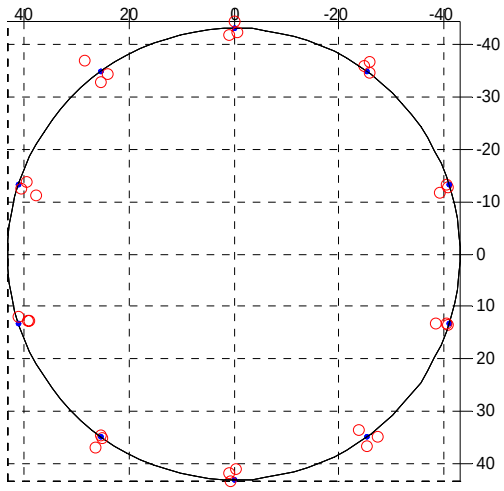


b)

Figure 5.14 The alignment of the recovered structure to the model. a) coarse alignment and b) refined alignment



a)



b)

Figure 5.15. a) Side view and b) top view of the refined alignment of the recovered structure to the model.

The top view and the side view of the recovered structure are presented in *Figure 5.13*. It can be observed again that the factorization step is not able to recover accurately the structure, especially the error in estimating the depth for some of the points is very big. But the bundle adjustment fixes the problem, and it can be seen that after this step the points follow the shape of cylinder. *Figure 5.14* shows the alignment of the recovered points to the model points after the coarse alignment step and after optimization, and The *Figure 5.15* shows the side view and top view of the refined alignment. It can be observed that the points farther away from the cameras are recovered with larger errors. The explanation is very simple. When projecting the points onto the camera frame, the coordinates of the projected points have smaller values when the distance from the point to the camera is larger. As the noise in the image plane is the same for all the points, the smaller values will be more affected. The same happens in a real scenario. Imagine that a picture is taken by pointing a camera inside a pipe. Closer sections of the pipe will appear on the sides of the image, while section far away from the camera get smaller and smaller and are placed in the center of the image.

After aligning the recovered structure to the model, the average point to point registration error is 2.9877 with a standard deviation 2.4116. The computed radius of the cylinder fitted to the reconstructed points is 42.5356. In other words the cylinder the radius of the recovered cylinder is estimated with an error of 1.08%.

5.5 The influence of the noise on the reconstruction accuracy

It is interesting to see how the noise in the images influences the accuracy of reconstructed cylinders. That means to find the maximum tolerated level of noise in the images in order to obtain o reconstruction accuracy of a certain level. The noise is directly reflected in the localization error of the feature points. The accuracy of the reconstruction is measured by the estimation error of the radius of fitted cylinder, as defined in the equation 5.10.

The required accuracy of the reconstruction depends on the application. As this work is related to the reconstruction of the ear canal, the need to find an acceptable level of tolerance in this case arises. As it was already seen, the 3D model of the ear canal is obtained by scanning an ear impression. The scanning process reconstructs very accurately the 3D model of the impression. It is also known that that the ear canal is not very rigid and its shape can change in

different situation, for example when chewing, opening mouth, etc. Different impression taking techniques assume the mouth opened, half opened, or closed. It is clear that the obtained 3D models will be also different in these situations. Anyway, the differences are small. The hearing aid shell built using this model should also fit properly in the ear canal. If is too small, there is the risk to fall down, and if it is too large it can produce discomfort for the wearer. It is clear that the tolerance level of the reproduction error is very small. To see how exactly small it is this level, people working in the hearing aids industry were asked about this problem. An exact number couldn't be obtained, but finally it was agreed that an error of 0.1 mm is acceptable. If we assume that the ear canal (or only a segment of the ear canal) is a cylinder with a diameter of 7 mm, then the reconstruction error of the cylinder radius shouldn't be bigger than 1.42%.

In the following experiment we try to determine the maximum level of noise (corresponding to the localization of the feature points in the images) such as the reconstruction error is smaller than 1.42%.

The same configuration of the points and cameras as in the previous experiment was used: 30 point (3 rings of 10 points) distributed on a cylinder with radius 43 units, and length 100 units. The noise was progressively added up to a value of 0.01, corresponding to a localization error of approximately 1.3% (6.5 pixels localization error of the features for a 512x512 image). The dependence of the localization error on the noise is depicted in *Figure 5.16.*

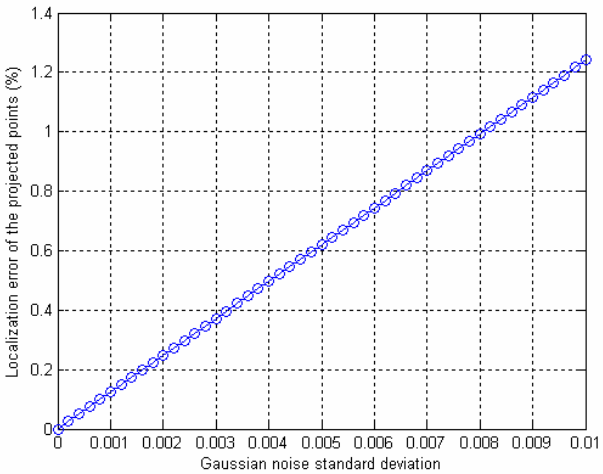


Figure 5.16 The relation between the noise and the localization error of the points projected into the camera frame

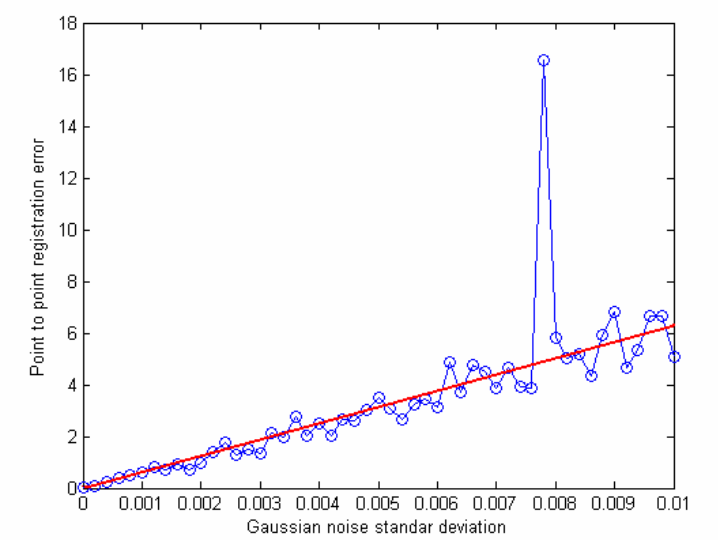


Figure 5.16 The dependence of the point to point registration error on the noise

In *Figure 5.16* it can be seen that the point to point registration error increases linearly with the noise. For each value of σ only one test was performed. That was enough to see the main trend of this dependency of the registration error on noise. A better experiment would perform more tests for each value of σ noise and average the results. Only one test was performed for each configuration due to the computational times. The point to point registration errors along with corresponding standard deviations are depicted in *Figure 5.17.*

In *Figure 5.18* the fitted cylinder radius is plotted against the localization error of the points. It can be seen that when the point localization error increases, the estimated cylinder radius has a decreasing trend. In the same figure the green curve is optimally fitted to the data with a certain degree of smoothness (a smoothing Thin Plate Spline or TPS). The black horizontal line in the figure represents the real radius, while the area between the two red lines belongs to the configurations with a reconstruction error below 1.42%. Once again, generating more tests for each configuration and averaging results can show more clearly the trend of estimated radius. In this case the fitted green curve intersects the red line at a value of 0.9 for the localization error.

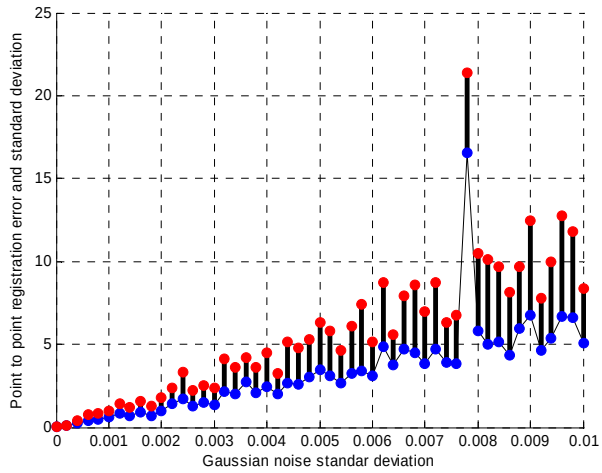


Figure 5.17 Point to point registration error (blue) and its standard deviation (black segments) function of noise.

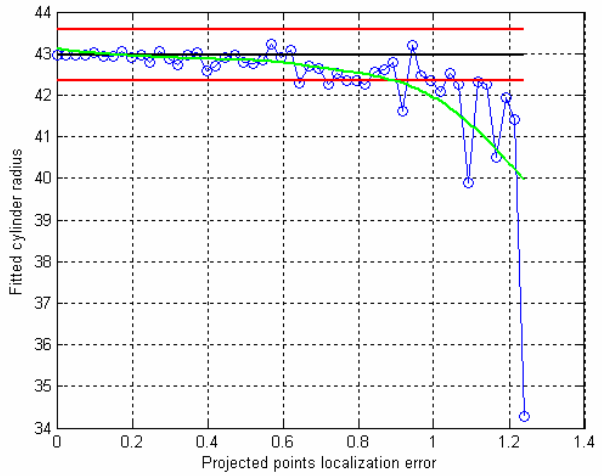


Figure 5.18 Fitted cylinder radius function of point localization error. The black line represents the real radius, and the red lines represent the accepted level of error: $\pm 1.42\%$ *of radius*

It means that, in order to reconstruct a cylinder such as the reconstruction error is less than 1.42%, the localization error of the feature points may have a maximum value of 0.9% (corresponding to 4.7 pixels for a 512x512 image). For safety reasons, a smaller value should be considered. For example a value of 0.6% corresponds to 3 pixels localization error for a 512 by 512 pixels

image. This value is not critical since many of the feature detectors have subpixel accuracy.

5.6 The influence of the cylinder radius on the reconstruction accuracy

The experiment is very similar with the previous one, but this time a fixed amount of noise is added for every configuration ($\sigma = 0.002$). The only variable parameter is the cylinder radius. In *Figure 5.19* and *Figure 5.20* the point to point registration error and the reconstruction error are plotted function of the cylinder radius. The both errors decrease while the radius increases. The explanation is simple: while the *z* coordinates of the points are the same in all the configurations, it results that the values of projections on cameras frames depend only on the radius. As the radius increases, also the projection values increase and become less sensitive to the noise.

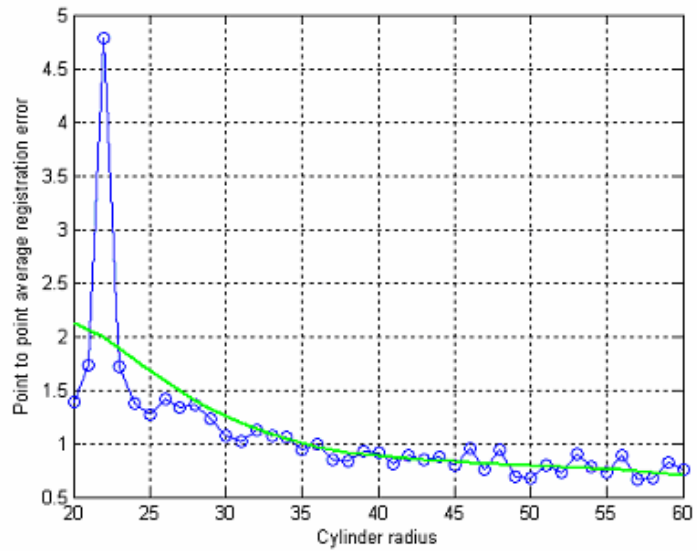


Figure 5.19 Point to point registration error function of cylinder radius

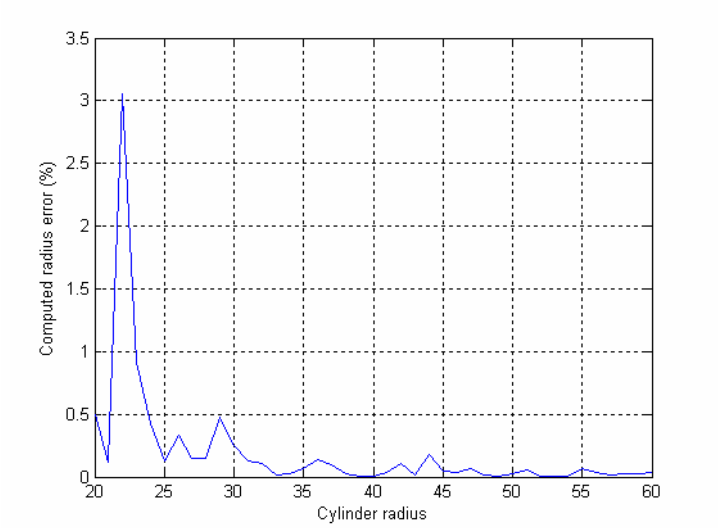


Figure 5.20 Reconstruction error function of cylinder radius

5.7 The influence of the number of points on the reconstruction accuracy

The experiment is similar with the two previous ones: same cylinder, same camera configurations. Noise with $\sigma = 0.005$ is added to the projected points. But this time the number of the points is not constant anymore. A variable number of points are randomly distributed over the cylinder' surface and the reconstruction errors are measured. In *Figure 5.21* and *Figure 5.22* the registration error and the reconstruction error are plotted against the number of the points. A larger number of the points don't necessarily improve the accuracy of the reconstruction. As long as the cylinder parameters (radius and length) don't change, the noise affects the projected points in a similar way for all the configurations. Only the position of the points in the 3D space is important, because the ones closer to the camera are less affected by noise.

It is also interesting to remark that in all the cases the reconstruction error is smaller then 1.42% (for $\sigma = 0.005$ the localization error is approximately 0.6%).

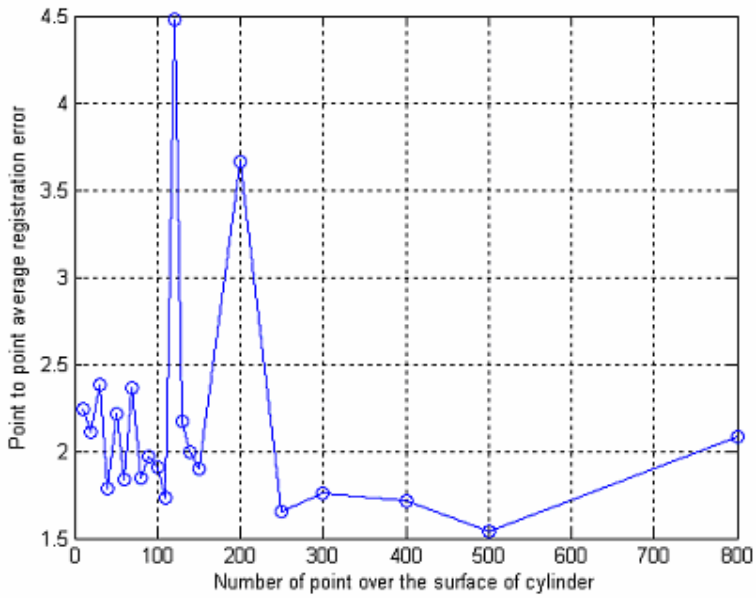


Figure 5.21 Registration error function of the number of points

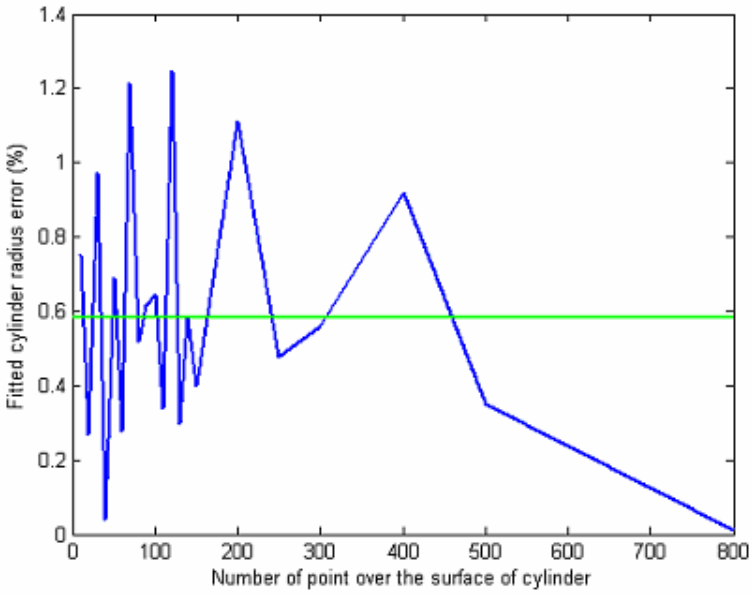


Figure 5.22 Reconstruction error function of the number of points

5.8 An experiment with real data

In this section a reconstruction of a real scene is performed with the proposed methods, and the reconstruction accuracy is measured.

The setup is very simple. It mainly consists from a camera (USB Labtec notebook web cam with adjustable focus) and a cylinder with 31 feature points. In order to obtain the 3D structure of the model, the exact positions of the points on the cylinder have to be known. Initially a grid with 1 cm square size is printed on a sheet of paper. To obtain exactly the same dimensions on the printed paper, automatically resizing features of the printer have to be disabled. The feature points are placed with a black marker on the corners of the grid at known positions, as it can be seen in *Figure 5.23*. The real coordinates of the feature points are assumed to be at the intersection of the crossing lines. There are four lines each of them having seven feature points, placed at two centimeters away of each other. The piece of paper with the points marked in the way described above is mapped to the inner part of a glass cylinder, as shown in *Figure 5.24.* As ruler marks were placed in advance on the paper, it is very easy to get the width of the unfolded paper mapped to the cylinder: 142.5 mm. Dividing this number by 2π the radius of the cylinder id obtained. Now that the radius of the cylinder is known, and the planar coordinates of the feature points (on the unfolded paper) are also known, then the 3D positions of the points on the cylinder are also obtained. The radius is in this case 22.67 mm. The cylinder model is shown in *Figure 5.25*. At this point the cylinder model is known and the reconstruction step can be performed.

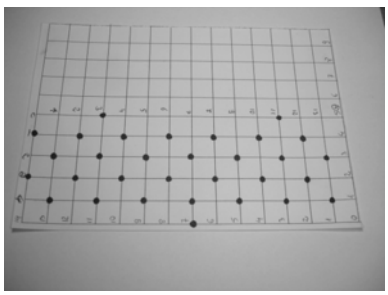


Figure 5.23 The placement of the feature points onto a rectangular grid with the size of the squares of 1 cm.

Figure 5.24 The setup used for the experiment with real data

A camera calibration is performed in order to obtain the internal camera parameters (the calibration matrix) and distortion coefficients. Camera calibration features of OpenCV (open computer vision) library are used in this case. The calibration grid is a checkerboard 6x7 squares, with square size of 7.05 mm printed onto an A4 sheet of paper. A number of 25 images of the calibration grid are taken at a resolution of 320x240 pixels. The images cover different positions of the camera, different view angles, and different orientations of the CCD sensor. The camera was fixed on a small support before taking each picture, to avoid the motion blur. The setup is shown in *Figure 5.26*. The built-in functionality of OpenCV is able to automatically detect the corners of the checkerboard, and based of their coordinates in different images it recovers accurately the camera calibration matrix together with distortion parameters.

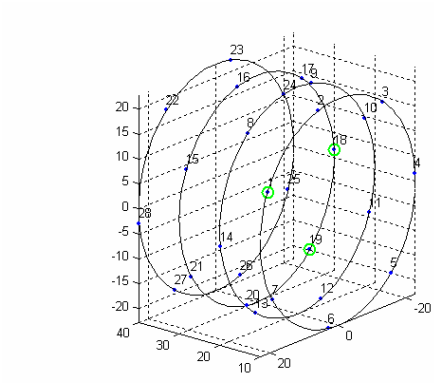


Figure 5.25 The cylinder model used in the experiment

Figure 5.26 The setup used to calibrate the camera

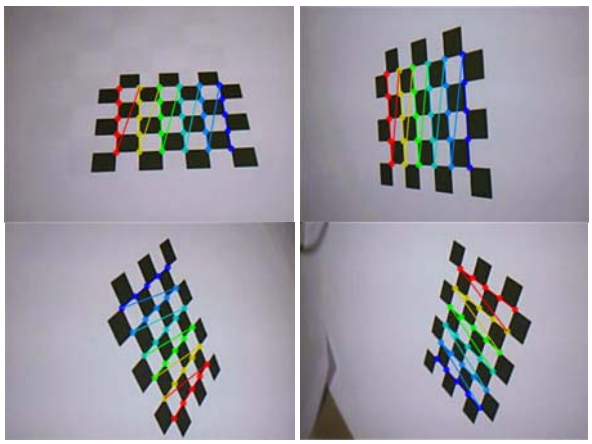


Figure 5.27 Camera calibration – detection of the calibration pattern

Several images of the calibration pattern along with detected corners are shown in Figure 5.27. After the calibration, camera internal parameters (focal length, principal point) and radial and tangential distortion coefficients are recovered. The camera calibration matrix K, principal point c, focal length f, and distortion coefficients are listed bellow:

$$K = \begin{bmatrix} 446.38 & 0 & 172.14 \\ 0 & 445.57 & 123.27 \\ 0 & 0 & 1 \end{bmatrix}$$; c = [172.14, 123.27];

f = [446.38, 445.57];

d = [0.075, 0.945, -0.0025, -0.0038], where the first two parameters correspond to radial distortions, and the last two correspond to tangential distortion.

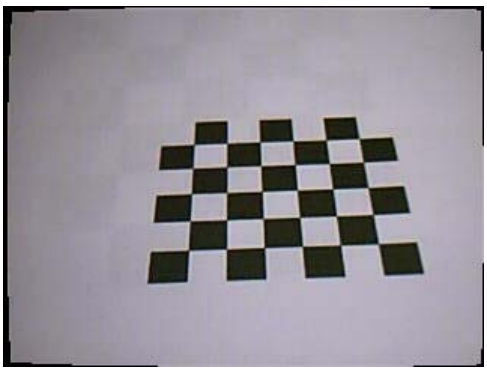


Fig 5.28 Example of undistorted image

The distortion parameters are used to undistort the taken images before processing. An example of an undistorted calibration grid image is in *Figure 5.28*.

In the next step four images of the cylinder are taken from different positions of the camera. The camera positions doesn't follow a specific pattern, they are as general as possible. The only restriction imposed is that all the feature points have to be visible in all the images. The relative positions of the cameras to the cylinder are not measured since we are interested only to see the accuracy of the recovered structure. The four images are shown in *Figure 5.29.*

In the next step the images are rectified using distortion parameters obtained after the calibration step (see *Figure 5.30*). The feature detection step is made manually for each image, trying to point the center of the features as good as possible. Manual selection of the points has also the advantage to provide the correspondences between the features in different images. The reason of performing manual annotation of the features is to make abstraction of the accuracy or robustness of a specific feature detector and tracker, and to avoid

the detection of features points which are not part of the model. For example, a blob detector would probably perform well in detecting the features in this particular example since the regions are black on a white background. A threshold properly chosen would eliminate other possible regions from the images. The feature points can be the centers of mass of the regions. A simple normalized cross-correlation associated with a distance constraint for the same feature in different images can determine the correspondences of the feature points.

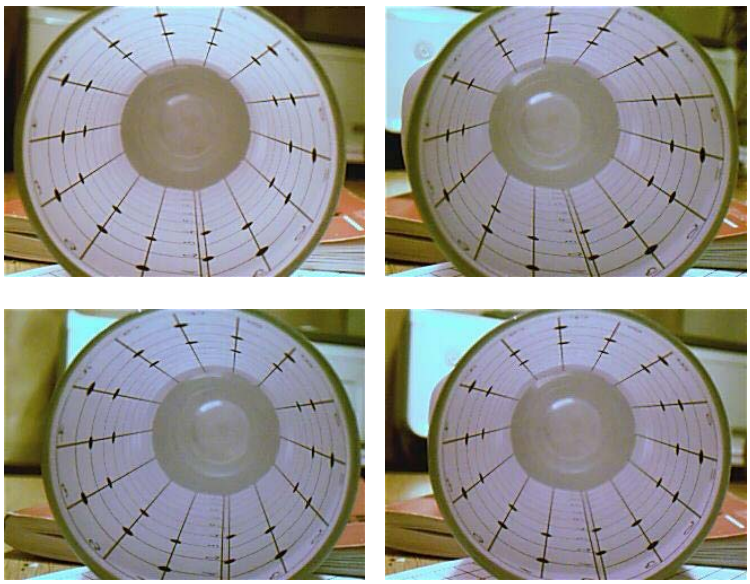


Figure 5.29 Four images of the test cylinder

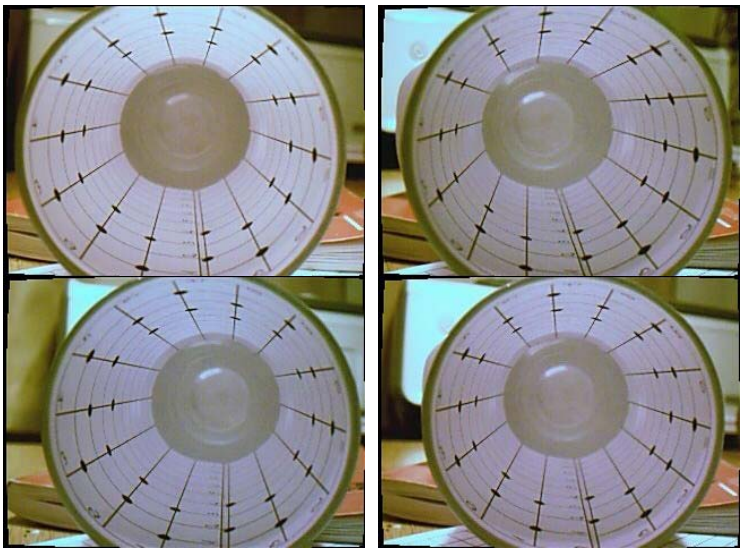


Figure 5.30 Distortion rectifications of the images

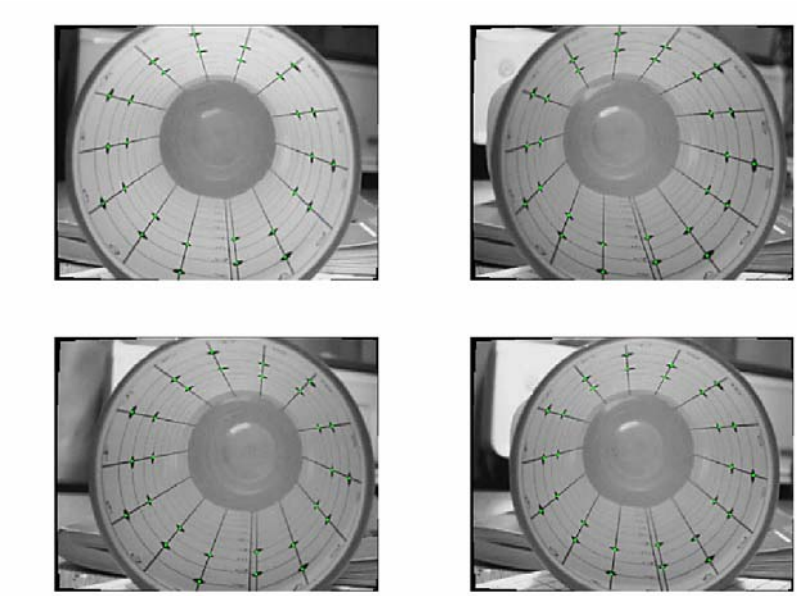


Figure 5.31 Annotated feature points in the images (green points)

The points manually selected in the four images are shown in *Figure 5.31*. In the following steps the 3D structure of the cylinder is recovered in the same way as in the experiments with synthetic data. Recovered points after factorization step and after bundle adjustment are shown in the *Figure 5.32*, coarse alignment of the recovered structure to the model in *Figure 5.33*, and the final alignment in *Figure 5.44*.

The average point to point registration error after the alignment is 0.7565 mm, with a standard deviation of 0.5664 mm, and the recovered radius of the fitted cylinder is 22.8179 mm. As the real radius is 22.67 mm, it means the cylinder radius is recovered with an error of 0.61%, less than the maximum tolerated level 1.42%.

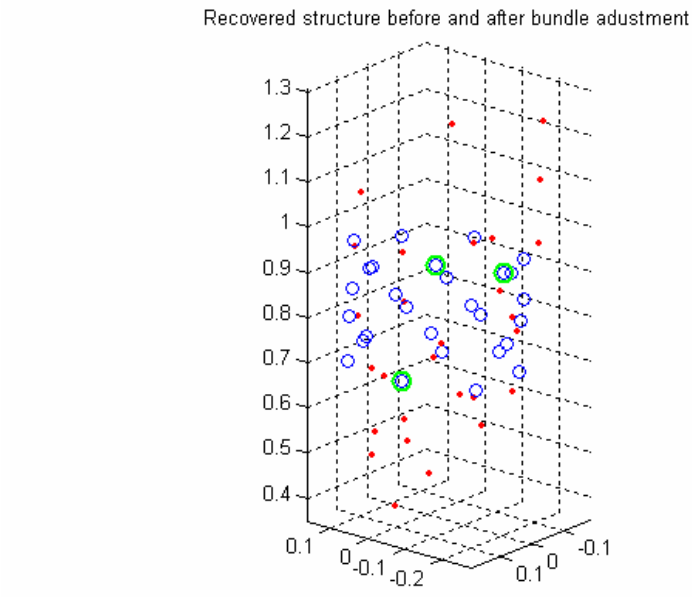


Figure 5.32 The recovered structure after factorization step (red points) and after bundle adjustment (blue points)

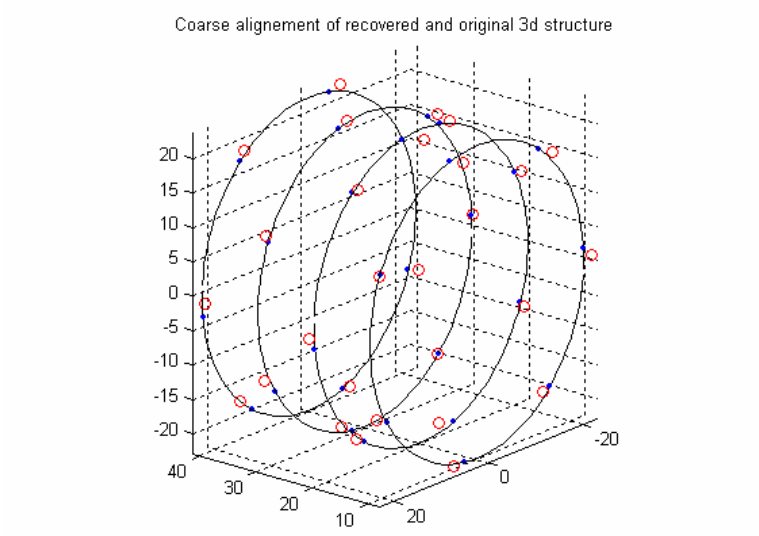


Figure 5.33 Coarse alignments of the recovered points to the model

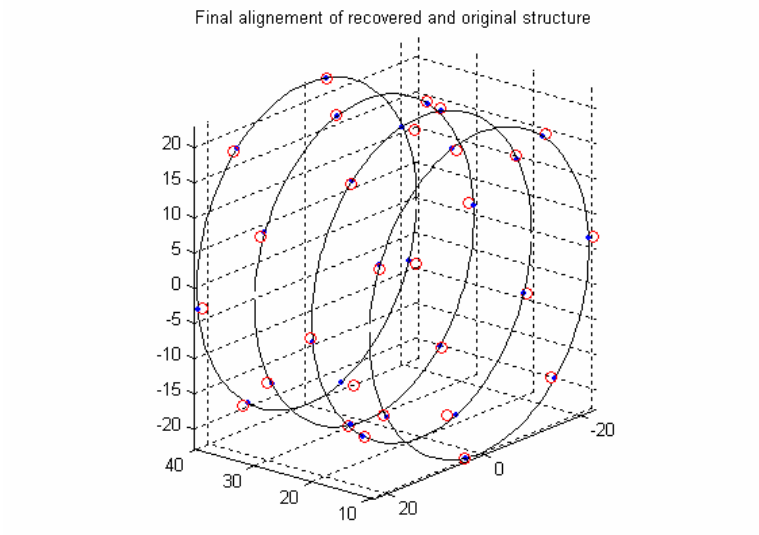


Figure 5.34 Refined alignment of the recovered structure to the model

CHAPTER 6

Conclusions

This thesis addressed the problem of 3D reconstruction of the human ear canal using feature based Structure from Motion methods. As seen in the previous chapters, these methods are based on the detection and tracking of interest points or regions (features) in different images of the same object or scene. The relations existing between same features in different images make possible the reconstruction of a sparse set of points onto the surface of object.

The otoscopic images proved to be a challenge for the feature detection algorithms. Several of state of the art feature detectors have been tested with images of the ear canal. Both interest point detectors and region detectors failed to find reliable features on the surface of the ear canal. Besides of this lack of features due to the natural smoothness of the ear canal surface, there are other negative aspects that can directly influence the performance of the feature detection algorithms. The specific illumination conditions created by the light source of the otoscope (specular reflections, poor or over illuminated regions), low contrast of images, blur generated by the motion of otoscope are the most important ones. Thus, some image preprocessing techniques like contrast adjustment or color normalization would be necessary. It was shown that the specular reflections and the circular border in the images generated by the tip of otoscope can fool the feature detection algorithms. Thus, many of the "features" are detected in regions that do not correspond to the real surface of the ear canal. Especially when an interest points detector is used, the ones

detected too close to the circular border should be discarded. The interest point detectors are not recommended, due to the natural curvature of the ear canal that can produce false edges in the images, and consequently strong responses of these detection methods.

Probably one of the most painful issues is the presence of the hair in the external part of the ear canal that practically blocks the field of view of the otoscope in some regions. The hair should be removed in advance in order to obtain useful images of the entire ear canal.

In the absence of robustly detectable features, the structure from motion methods can no be applied for the 3D reconstruction of the ear canal. Adding artificial features is then a necessary condition in order to make this solution possible. The natural opening of the ear allows, at least theoretically, the placement of artificial features inside. For example, one can imagine spraying some high contrast paint inside the canal. Such a procedure can create regions that can be easily identified by the interest region detectors, despite of the other unfavorable conditions. For example, if a dark color is used to create these regions, the specular reflections or over saturated areas in the images can be avoided by limiting the searching procedure in a certain band of image intensities.

A specific SFM algorithm was used in all the cylinder reconstruction experiments. In a first stage, the structure and camera motion are estimated with a factorization algorithm. This method is based on a linearization of the pinhole camera model under orthographic projection. An initial guess of the scene structure and camera motion obtained with this method is used to initialize a bundle adjustment algorithm. This is the optimization step refining the structure and motion such as the projection error of the reconstructed points is minimized. The experiments showed the deficiency of the factorization step in estimating the depth of the points. While for the points placed closer to the camera the depths are correctly estimated, the errors increase for the points farer away. In all the cases the bundle adjustment step was able to correct these errors. This makes me believe that the optimization step is mandatory for any SFM algorithm used for the reconstruction of cylindrical objects, when high accuracy is required.

Several experiments were performed with synthetic data in order to understand how the reconstruction accuracy is influenced by the localization error of the feature points, cylinder radius, or the number of feature points. The radius of the cylinder best fitted to the reconstructed points was used to measure the reconstruction error. After consulting people working in hearing aids industry, it was agreed that a model of the ear canal estimated with an error of 0.1 mm is

acceptable (considering that the ear canal is not rigid, and its shape can change when opening mouth, chewing or yawning). The hearing aids shells produced within this level of error fit well enough in the ear canal. Reported to the diameter of the ear canal (7 mm), the maximum cylinder reconstruction error is 1.42% of the real radius. The results of the experiments showed that a cylinder can be reconstructed within this level of error if the localization error of the feature points is about 0.6% of the image size. For example, in the case of a 512x512 pixels image, the features points have to be localized with an accuracy of 3 pixels. This value is not critical, since many feature detectors have subpixel accuracy. It was also shown that an increasing radius of the cylinder, relative to the same camera configuration, also improve the reconstruction accuracy. This result was expected since a larger cylinder also appears larger in the images, and the feature points are more accurately localized. A bigger number of feature points do not necessarily improve the quality of reconstruction. Not the number, but the positions of the points on the cylinder relative to the camera are important. The sections of cylinder farer away from the camera correspond to smaller regions closer to the center of image (camera pointing inside and along the cylinder axis). In this case, the features cannot be precisely localized.

In a real data experiment, a cylinder with a radius of 22.67 mm and 31 feature points was reconstructed with an error of 0.6%, using only four 320x240 pixels images. After the alignment to the model, the average point to point registration error was 0.75 mm with a standard deviation of 0.56mm.

The experiments performed with both synthetic and real data showed that cylindrical objects can be accurately reconstructed with SFM methods, as long as it is possible to detect and track features in multiple images within an acceptable level of accuracy.

To summarize, in my opinion there are two conditions that have to be satisfied in order to successfully apply the SFM methods to the 3D modeling of the ear canal: 1) the hairs inside the ear canal are removed and 2) some features are manually added.

There are many aspects not addressed in this thesis. It is known that the SFM methods are able to reconstruct an object only up to an unknown scale factor. If we assume that a model of the ear canal is successfully obtained with a SFM method, then recovering this scale factor is very important since it can drastically affect the accuracy of the final model. This can be a difficult task given that the real model is not known in advance. An idea is to place some control points onto the surface of the ear canal that can be easily identified in the reconstructed model. Assuming that some metric relations between these

points can be estimated, then the same metric constraints can be imposed to the model.

Another issue not addressed here is the density of reconstructed points. Feature based SFM methods in general produce a sparse set of 3D points on the surface of the object. The rapid prototyping systems require a very large number of 3D points in order to create a precise replica of the model. Dense reconstruction is in general a topic close related to SFM. It is shown for example in [2] how starting from a sparse set of corresponding points in the images, a dense reconstruction can be performed. Anyway, the number of reconstructed points is limited by the number of points in the images, and it is evident that not all the points in one image can be matched to points in other images. Considering the almost regular shape of the ear canal it is very probable that a simple 3D interpolation of a large enough set of reconstructed point can offer a very good dense estimation of the model.

Key frame selection should be also considered in longer sequences, in order to obtain the initial guess for the structure and motion. A large number of corresponding features is desirable in these frames, and a sufficient base line between them to obtain an initial structure by triangulation.

Modeling the ear canal with SFM methods is subject of further research. Creating the necessary conditions (hairs removal, artificial features addition) opens the possibility to perform real experiments. A final conclusion can be made only performing such experiments, and comparing the results with very precise models obtained by scanning ear impressions with laser rangers.

3D reconstruction of the ear canal from otoscopic images is a very large and challenging topic and it is my regret that the time constraints limited this work to the form presented here.

References

[1] M. Pollefeys, L. J. V. 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] M. Pollefeys, *Visual 3D Modeling from Images*, Tutorial Notes, University of North Carolina – Chapel Hill, USA, 2004

[3] C. Wu, Y. Chen, C. Liu, C. Chang, Y. Sun, *Automatic extraction and visualization of human inner structures from endoscopic image sequences*, Medical Imaging 2004: Physiology, Function, and Structure from Medical Images. Proceedings of the SPIE, Volume 5369, pp. 464-473 (2004), pp. 464-473, 2004

[4] R. Chellappa, G. Qian, S. Srinivasan, *Structure from Motion: Sparse Versus Dense Correspondence Methods*, in ICIP (2), pp. 492-499, 1999

[5] J. Oliensis. *A Critique of Structure-from-Motion Algorithms*, Computer Vision and Image Understanding: CVIU 80(2), 172—214, 2000

[6] T. Thormaehlen, H. Broszio, P. Meier, *Three-Dimensional Endoscopy*, Falk Symposium No. 124, Medical Imaging in Gastroenterology and Hepatology, Hannover, Germany, September 2001, Kluwer Academic Publishers, 2002, ISBN 0-7923-8774-0 0(0), 2002

[7] J. J. Caban, W. B. Seales, *Reconstruction and Enhancement in Monocular Laparoscopic Imagery*, in 'Proceedings of Medicine Meets Virtual Reality 12', 2004

[8] F. Devernay, *3D Reconstruction of the Operating Field for Image Overlay in 3D-Endoscopic Surgery*, in 'ISAR '01: Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR'01)', IEEE Computer Society, Washington, DC, USA, pp. 191, 2001

[9] D. Stoyanov, A. Darzi, G. Z. Yang, *Dense 3D Depth Recovery for Soft Tissue Deformation During Robotically Assisted Laparoscopic Surgery*, 2004

[10] D. Stoyanov, G. P. Mylonas, F. Deligianni, A. Darzi, G. Yang, *Soft-Tissue Motion Tracking and Structure Estimation for Robotic Assisted MIS Procedures*., in 'MICCAI (2)', pp. 139-146, 2005

[11] Y. Wang, D. Koppel, H. Lee, *Image-Based Rendering And Modeling In Video-Endoscopy*, in 'ISBI', pp. 269-272, 2004

[12] C. Lee, Y. Wang, D. Uecker, Y. Wang, *Image analysis for automated tracking in robot-assisted endoscopic surgery*, in 'ICPR94', pp. A:88-92, 1994

[13] K. Mori, D. Deguchi, J. Hasegawa, Y. Suenaga, J. Toriwaki, H. Takabatake, H. Natori, *A Method for Tracking the Camera Motion of Real Endoscope by Epipolar Geometry Analysis and Virtual Endoscopy System*, in 'MICCAI '01: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention', Springer-Verlag, London, UK, pp. 1-8, 2001

[14] D. Burschka, M. Li, R. Taylor, G. D. Hager, M. Ishii, *Scale-Invariant Registration of Monocular Endoscopic Images to CT-Scans for Sinus Surgery*, Medical Image Analysis 9(5), 413-439, 2005

[15] Q. Liu, R. J. Sclabassi, N. Yao, M. Sun, *3D Construction of Endoscopic Images Based on Computational Stereo*, in Bioengineering Conference, 2006. Proceedings of the IEEE 32nd Annual Northeast, 2006

[16] D. Koppel, Y. Wang, H. Lee, *Viewing Enhancement in Video-Endoscopy*, in 'WACV '02: Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision', IEEE Computer Society, Washington, DC, USA, pp. 304, 2002

[17] M. Wynne, J. Kahn, D. Abel, R. Allen, *External and Middle Ear Trauma Resulting from Ear Impressions*, Journal of the American Academy of Audiology, Vol. 11, No. 7, 2000

[18] R. Trace, *Video otoscopy: Applications in Audiology*, ADVANCE for Speech-Language Pathologists & Audiologists, Volume 6, Number 9, March 4, 1996

[19] R. F. Sullivan, *Video otoscopy: Basic and Advanced Systems*, The Hearing Review: Volume 2, Number 10; NOV / DEC, 1995 pp 12-16, 1995

[20] R. I. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003

[21] R. I. Hartley, *In defence of the 8-point algorithm*. In Proceedings of the IEEE International Conference on Computer Vision, 1995

[22] T. S. Huang, O. D. Faugeras, *Some properties of the E matrix in two-view motion estimation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(12):1310-1312, Dec. 1989

[23] P. Torr, A. Zisserman, *Robust parametrization and computation of the trifocal tensor*, Image and Vision Computing, 15(1997) 591-605, 1997

[24] M. A. Fischler, R. C. Bolles. *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*. Comm. of the ACM 24: 381—395, June 1981

[25] C. Tomasi, T. Kanade, *Shape and motion from image streams under orthography: A factorization method*, Int'l J.Computer Vision'92, 9(2):137-154, Nov. 1992

[26] S. Christy and R. Horaud, *Euclidian shape and motion from multiple perspective views by affine iterations*, INRIA Tech. Rep. RR-2421, Dec. 1994

[27] H. Aanæs, R. Fisker, K. Åström, *Robust factorization*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 9, pp. 1215-1225, Sep. 2002

[28] T. Kanade, D.Morris, *Factorization methods for Structure from Motion*, Phil. Trans. R. Soc. Lond., A(356):1153-1173, 1998

[29] B. Triggs, P. F. McLauchlan, R. I. Hartley, A. W. Fitzgibbon, *Bundle Adjustment - A Modern Synthesis*., in 'Workshop on Vision Algorithms', pp. 298-372, 1999

[30] M. Lourakis, A. Argyros, *The design and implementation of a generic sparse bundle adjustment software package based on the Levenberg--Marquardt algorithm*, Tech. Rep. 340, Institute of Computer Science---FORTH, Heraklion, Crete, Greece, 2004

[31] J. Heikkila, *Geometric Camera Calibration Using Circular Control Points*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 10, pp. 1066-1077, Oct., 2000

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

[33] Z. Zhang, *Flexible Camera Calibration By Viewing a Plane From Unknown Orientations*. International Conference on Computer Vision (ICCV'99), Corfu, Greece, pages 666-673, Sep. 1999

[34] C. C. Slama, *Manual of Photogrammetry*, 4th ed., American Society of Photogrammetry, Falls Church,Virginia, 1980

[35] J. Heikkila, O. Silven, *A Four-step Camera Calibration Procedure with Implicit Image Correction*, cvpr, p. 1106, 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97), 1997

[36] V. Asari, S. Kumar, D. Radhakrishnan, *A new approach for non-linear distortion correction in endoscopic images based on least squares estimation*, IEEE Trans. Med. Imaging 18 (4) (1999) 345-354, 1999

[37] C. Wengert, M. Reeff, P. C. Cattin, G. Székely, *Fully Automatic Endoscope Calibration for Intraoperative Use*, in 'Bildverarbeitung für die Medizin', Springer-Verlag, , pp. 419-23, 2006

[38] J. P. Helferty, C. Zhang, G. McLennan, W. E. Higgins,
Videoendoscopic distortion correction and its application to virtual guidance of endoscopy, MedImg(20), No. 7, pp. 605-617, Jul. 2001

[43] R. Shahidi et al., *Implementation, calibration and accuracy testing of an image-enhanced endoscopy system*. IEEE Transactions on Medical Imaging, 21(12):1524--1535, Dec. 2002

[40] Miranda-Luna, R., Blondel, W.C.P.M., Daul, C., Hernandez-Mier, Y., Posada, R., Wolf, D., *A simplified method of endoscopic image distortion correction based on grey level registration*, ICIP04(V: 3383-3386)

[41] H. Hideaki, Y. Yagihashi, Y. Miyake, *A new method for distortion correction of electronic endoscope images*, IEEE Trans. Med. Imaging 14 (3) (1995) 548-555

[42] W. E. Smith, N. Vakil, S. A. Maislin, *Correction of Distortion in Endoscope Images*, IEEE Transactions on Medical Imaging, vol. 11, No. 1, Mar. 1992

[43] R. Shahidi et al., *Implementation, calibration and accuracy testing of an image-enhanced endoscopy system*. IEEE Transactions on Medical Imaging, 21(12):1524--1535, Dec. 2002

[44] T. Tuytelaars, L. Van Gool, *Matching Widely Separated Views based on Affine Invariant Regions*, International Journal on Computer Vision, 59(1):61–85, 2004

[45] C. Harris, M. Stephens, *A combined corner and edge detector*, In: Proceedings of the Alvey Vision Conference, pp. 147-151, 1988

[46] H. Bay, T. Tuytelaars, L. Van Gool, *SURF: Speeded Up Robust Features*, Proceedings of the 9th European Conference on Computer Vision, May 2006

[47] J. Canny, *A computational approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, pp. 679–698, 1986

[48] T. Kadir, M. Brady, *Scale, saliency and image description*. International Journal of Computer Vision. 45 (2):83-105, Nov. 2001

[49] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Van Gool, *A comparison of affine region detectors*. IJCV 65 43–72, 2005

[50] J. Matas, O. Chum, M. Urban, T. Pajdla, *Robust wide-baseline stereo from maximally stable extremal regions*, In Proceedings of the British Machine Vision Conference, Cardiff, UK, pp 384-393, 2002.

[51] K. Mikolajczyk, C. Schmid, *Scale and affine invariant interest point detectors*. IJCV 60, pp. 63-86, 2004

[52] T. Lindeberg, *Feature detection with automatic scale selection.* IJCV 30(2), pp. 79-116, 1998

[53] K. Mikolajczyk, C. Schmid, *Indexing based on scale invariant interest points.* In: ICCV. Volume 1, pp. 525-531, 2001

[54] D. Lowe, *Object recognition from local scale-invariant features*. In: ICCV, 1999

[55] K. Mikolajczyk, C. Schmid, *A performance evaluation of local descriptors.* In:CVPR. Volume 2. pp. 257-263, 2003

[56] K. Mikolajczyk, C. Schmid, *A performance evaluation of local descriptors*. PAMI 27, pp.1615-1630, 2005

[57] C. Tomasi, T. Kanade, *Detection and tracking of point features*, Carnegie Mellon University Tec. Rep. CMU-CS-91-132, Apr. 1991

[58] J. Shi, C. Tomasi, *Good features to track*, IEEE Conf. Computer Vision and Pattern Recognition, pp. 593-600, Jun. 1994

[59] S. Belongie, J. Malik, J. Puzicham, *Shape matching and object recognition using shape contexts*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(4):509.522, 2002.

[60] W. Freeman, E. Adelson, *The design and use of steerable filters*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(9):891.906, 1991.

[61] L. Van Gool, T. Moons, D. Ungureanu, *Affine / photometric invariants for planar intensity patterns*. In Proceedings of the 4th European Conference on Computer Vision, Cambridge, UK, pp. 642-651, 1996.

[62] S. Lazebnik, C. Schmid, J. Ponce, *Sparse texture representation using affine-invariant neighborhoods*. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA, pp. 319-324, 2003.

[63] L. Florack, Bart ter Haar Romeny, J. J. Koenderink, M. A. Viergever: *General intensity transformations and differential invariants.* JMIV 4, pp. 171-187, 1994

[64] B. D. Lucas, T. Kanade, *An iterative image registration technique with an application to stereo vision.* In Proceedings of the 7th International Conference on Artificial Intelligence, pp. 674-679, Aug. 1981.

[65] T. Zinßer, J. Schmidt, H. Niemann, *Point Set Registration with Integrated Scale Estimation*;

International Conference on Pattern Recognition and Image Processing, pp. 116-119, 2005

[66] I. S. Lindsay, *A tutorial on Principal Components Analysis*, 2002

[67] J. Shlens, *A Tutorial on Principal Component Analysis*, 2005

[68] K. Robins, R. Harris, *Silicone Impression Material in the Middle Ea*r, http://www.rcsullivan.com/www/forum/harris/simitme/simitme.htm

[69] E. Grenda, *The Most Important Commercial Rapid Prototyping Technologies at a Glance*, 2006 http://home.att.net/~castleisland/rp_int1.htm

[70] J. Y. Bouguet, *Camera Calibration Toolbox for Matlab®*, http://www.vision.caltech.edu/bouguetj/calib_doc/.

[71] *Intel OpenCV Computer Vision Library (C++)*, http://www.intel.com/research/mrl/research/opencv/