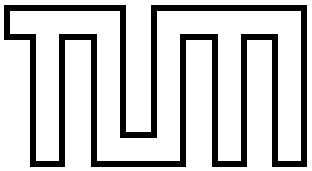


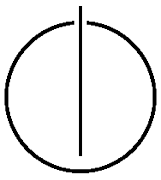
FAKULTÄT FÜR INFORMATIK

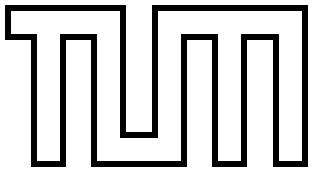
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master Thesis in Biomedical Computing

Circular Marker Tracking for Industrial Augmented Reality Systems

Hemal Naik





FAKULTÄT FÜR INFORMATIK

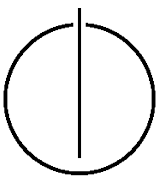
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master Thesis in Biomedical Computing

Circular Marker Tracking for Industrial Augmented Reality Systems

Verfolgung von Kreismarken für industrielle Augmented Reality Systeme

Author:	Hemal Naik
Supervisor:	Prof. Dr. Nassir Navab
Advisor:	Dr. Pascal Fallavollita
	Dr. Peter Keitler
	Dr. Yuji Oyamada
Date:	October 15th, 2013



I assure the single handed composition of this master's thesis only supported by the declared resources.

München, October 15th, 2013 Hemal Naik

Acknowledgements

I have a great deal of people to thank for the presented work. A monumental support was given from Peter Keitler, Yuji Oyamada (over Skype) and Pascal Falavollita during the entire period of the thesis. I would like to thank my superheroes, Andrew Zissermann, David Forsyth and their colleagues at Cambridge (of 90s) who rescued me with their concept of Invariants,and gave the math to solve the most crucial problem of my thesis. I would sincerely like to thank Prof. Navab, without his encouragement this work would not be possible.

Now, on a lighter note, I would thank ASUS corporation for making a laptop that understood my frustration and never gave up on me. I would like to thank English garten and its fresh air for helping me keeping my mind fresh over stressful times. Dropbox ,Zotero and GIT Hub for giving me peaceful sleep knowing that my code and papers are safe (NAS is an exception) in their cloud. I thank the anonymous person who failed me in Computer Vision 1 class in 2011, which became my inspiration to get even with the subject. My colleagues at Extend3D have been a great support. I would like to acknowledge my friends for their invaluable support. My dear friend Patrick De Boer who constantly distracted me with a record of his antics at MIT. Indian media for providing plenty of entertaining news over upcoming elections. Miss. Natalia Zarawska and Mr. Marco Esposito for providing entertaining vacations in Krakow and Tuscany. Mr. Navketan Verma and Mr. Pujit shingala for a providing entertaining discussions.

Finally, I would like to thank my family for their outstanding support for the entire duration of my course.

Abstract

Augmented Reality (AR) applications are becoming more and more popular in the Industrial domain. Any Industrial AR (IAR) application developed to aid in manufacturing or quality control is expected to deliver accurate results with robust performance. Work flow integration is rated as one of the most desirable features by industrial users. Existing marker based tracking solutions do not provide seamless integration in to the existing industrial work flow. Moreover, existing systems require additional effort from users to set-up the system.

In this thesis, we propose a new 3D object tracking approach for IAR applications using Circular markers. Circular markers are traditionally used in the industry for photogrammetric measurements. Existing tools can be used to compute accurate 3D information (position and surface normal) of the markers. However, 2D-3D correspondence problem has to be solved in order to use circular markers for monocular object tracking applications. We have proposed a new method to solve the correspondence problem from a single image using image projections of the circular markers. In this method, 3D perspective invariants are extracted from a pair of image conics, and feature descriptors are created. We also present an effective matching strategy to directly compare 2D-3D descriptors in order to establish correspondence relationship. Furthermore, the algorithm is tested on a real 3D object to validate its effectiveness. Additionally, a performance (Matching Accuracy, and speed) analysis with multiple markers and high resolution images is presented to support success of the correspondence matching method. The key contribution of this work is the feature descriptor design and the matching strategy.

CONTENTS

Outline of the Thesis

Part I: Introduction and Theory

CHAPTER 1: INDUSTIAL AUGMENTED REALITY (IAR) This chapter will introduce the reader to Industrial AR environment and its stringent requirements. This chapter also introduces the hardware to be used with the intended tracking approach.

CHAPTER 2: MOTIVATION AND PROBLEM STATEMENT In this chapter the motivation behind developing a new tracking method is presented. Additionally, the problems with the existing approach are discussed and a sound reasoning behind choice of circular markers is given. The chapter is concluded by defining the problem statement.

CHAPTER 3: TRACKING WORK FLOW AND THEORY This chapter intends to provide understanding of the geometric and *Computer Vision* (CV) concepts used in our work. Moreover, work flow for the tracking system is defined and relevant literature is discussed.

Part II: Implementation and Experiments

CHAPTER 4:INVARIANT STABILITY ANALYSIS

This chapter includes detailed analysis of conic back projection method. Also, details regarding the structure of the feature descriptor are discussed.

CHAPTER 5: TRACKING IMPLEMENTATION This chapter focuses on implementation of the discussed theoretical concepts for correspondence matching. The complete matching strategy is discussed with appropriate examples.

CHAPTER 6: EVALUATION

In this chapter we will focus on evaluating the implemented marker identification method. Different experiments are designed to comment on various aspects like usability, performance and matching accuracy.

CHAPTER 7: CONCLUSION & FUTURE WORK

This chapter presents final conclusion of our work. An overview of key contributions is given and future work is discussed.

Part I.

Introduction and Theory

CHAPTER
ONE

INDUSTRIAL AUGMENTED REALITY (IAR)

Augmented Reality (AR) is a technology which combines Virtual world and real world. This augmentation can be visualised through a display (Fixed, Hand held or Wearable) [HF04] or by projecting [WCWL12] the virtual information on a real scene. Even before the term "Augmented Reality" was coined, Ivan Sutherland [Sut65] [Sut68] demonstrated first ever AR application with *The Ultimate Display*. In the following years many researchers experimented and presented various AR applications for field of medicine, entertainment, military and manufacturing. The surge of powerful hand held equipments in past decade has helped AR make its way from research labs to the commercial domain [MET, GOO]. Azuma [Azu97] presented a comprehensive survey on AR systems and its applications. Azuma gave a valuable insight into design components of an AR system and discussed key factors that contribute to rich and effective AR experience. Industrial Augmented Reality (IAR) systems are AR systems specifically designed and customised for the Industrial environment.

Tracking is one of the most important components of any AR system. Tracking is method which computes position and orientation of camera with respect to a target coordinate frame, such that the desired augmentation can be performed based on this information. Accurate tracking is of paramount importance in Industrial applications. At the same time conditions of industrial environment are hostile towards conventional tracking solutions. The aim of this thesis is to present a new tracking solution specifically for Industrial Augmented Reality systems. However, it is very important to have a sound background about IAR systems and its challenges to understand the arguments behind developing this new method. This chapter intends to introduce the reader to the field, applications, and the equipment; before we explain the motivation of our work.

1.1. IAR Systems and Applications

Augmented Reality research has found support from industrial community since its early days. That being said, industrial interests in AR are driven by different factors from that of the academic researchers. IAR development got attention of the industry because of its potential to bring cost down, increase productivity and enhance user experience. Researchers

at *Boeing* [CMGJ99] were one of the first industrial groups to integrate Augmented Reality into manufacturing process. ARVIKA [Fri], a project initiated by Germany's Federal Ministry of Education and Research was first large scale effort to propel use of IAR systems. In the following years various research groups exposed versatile nature of AR applications. A brief account of AR applications developed for Automotive and Aeronautical industry was given by Regenbrecht[RBW05]. These applications demonstrate use of AR in different sections of industrial product life cycle model namely Product Design, Manufacturing, Commissioning, Inspection and Maintenance, and Decommissioning [Git07].

Contradictory to multiple uses and advantages professed by researchers, AR is scarcely visible in current industrial work flow. A recent survey by Fite-Georgel [FG11] tried to examine this contradiction by providing a detailed review of current state of IAR. He defined evaluation criteria for a generic IAR application and evaluated feasibility of existing IAR applications. He also commented on the hurdles faced by IAR to penetrate into the industrial market. To understand this we will put weight on the question *"What does industry expect from an IAR system?"* in the following section.

1.2. Requirements of Industry

As discussed in the earlier section we know that IAR, although being supported by more than a decade of research, has difficulties in cementing its position in industry. Therefore, we need to understand what features an IAR system must have in order to be widely accepted as part of any industrial process. Navab [Nav04] provided useful tips on developing "Killer Apps" for IAR. Navab also explained that outside the lab environment, only a practical, effective and well integrated solution can win the confidence of the industry. We have compiled a list of *must have* features for an IAR application from work of different researchers. [ML05, FG11, Nav04, Bro99].

Work flow integration : This is one of the most desired feature from any IAR application. The system should easily fit into existing work flow irrespective of the user or nature of process. This means that user can be either end user or a person involved in any part of product development life cycle. It is undesired if the application adds additional effort in a predefined industrial work flow. Generally any unnecessary change in work flow introduces loss of time which in turn increases cost. Therefore, it is advised to devote time for customising application for smooth integration in industrial work flow.

Usability : A user friendly application saves time and consequently gains faster acceptance from user. A highly innovative and effective system developed in research labs is more likely to survive in practical world if its easy to set up, learn and operate. A rich user experience for both expert and novice user is an invaluable feature for any IAR application.

Accuracy : AR might have many applications which can afford occasional compromise on accuracy, but industrial applications have no room for such compromise. In simple words less accurate system is less reliable system. IAR

applications, especially the ones designed for quality check, maintenance and end-user, can not accept unreliable result.

Robustness : Many IAR applications fail to win over its user segment due to lack of robustness. A robust application is one that maintains quality of performance and result over multiple user environments. Ability to adapt with changes in user environment (light conditions, complexity of design or line of sight etc.) without compromising in results is a vital trait for IAR.

Scalability : A practical and viable design is extremely important to bring a research prototype to the shop floor. Any IAR system that can be methodically reproduced and distributed with feasible financial investment is highly suited for industrial users. Also, depending on the application the system is expected to have reasonable physical presence in order to avoid blocking the work space.

1.3. Application and Equipment

For developing any user specific system it is necessary to include the user in the development process. In his evaluation of different IAR systems Fite-Georgel [FG11] accurately pointed out that chances of success are enhanced when industrial partners are involved in the development process. We have collaborated with Extend3D GmbH [EXT] for this research project. The problem statement is defined with reference to their Projective Augmented Reality systems. We have used their industrial expertise and experience for developing our tracking solution.

1.3.1. Projective Augmented Reality

Projective AR systems are basically AR systems which use projectors as means of visualisation of virtual content in real world. Weng [WCWL12] explains basic nature of interactive projection based AR systems. Projective AR systems typically consist of single or multi camera-projector set-up. The simplified working principle of the system can be explained in following steps:

- **Tracking** : A single or multiple Camera system tracks the object of interest from the image of the scene.
- **Projection** : Projector projects virtual information on or around desired object.

The reader should note that for simplicity of understanding we have used term 'Tracking' for combined process of image feature detection, correspondence matching and pose estimation. Each of these will be discussed thoroughly in the following chapters of the thesis. Projection based method of augmentation is different from display (fixed or wearable) based augmentation. Both may use similar means for object tracking, but they use different means for augmentation. Display based AR systems faces obstacles like depth perception, rendering quality etc. [Azu97], whereas Projection based AR systems have to deal with problems of light conditions, projection surface etc.[Bim06].

1.3.2. Application

The work of Schwerdtfeger [SK06], a laser projection system for industrial AR, can be seen as the primitive idea behind our desired application. Our research partner Extend3D [EXT] manufactures projective IAR systems namely WERKLICHT ® Pro and WERKLICHT® HD (Section 1.3.3). These products are fully functioning projective IAR systems, designed to save time and boost productivity in various areas of product development cycle. The augmented information naturally depends on the area of application (i.e. Production, Servicing etc.). For example projections are used for visual task guidance or for providing additional information about a specific object in the scene. A system is required in order to support a range of applications in different workspaces, keeping this in mind the tracking scheme is chosen has inside-out [WAC+90] configuration. The projector-camera system is calibrated, Hence geometric relationship between projector and camera is known. The reader is referred to work of Tsai [Tsa87], Zhang [Zha00] and Kimura [KMK07] for better understanding of camera and projector calibration methods. The applications largely involves tracking a 3D industrial object. Therefore, additional knowledge of 3D CAD model is used for tracking object and customise the projective augmentation accordingly.

1.3.3. Equipment

WERKLICHT® Pro : *Figure 1.1* shows equipment and its example application. As displayed in *Figure 1.1a* the equipment consists of a pair of optical camera and one laser projector. Its a classical stereo camera set-up integrated with a projector in a single housing. *Figure 1.1b* shows how the laser projection guides the worker in performing his task. The cameras track encoded square markers present on the target object to perform accurate projective guidance. The laser projection is able to provide good contrast against multiple surfaces. The operating range of the system is up to 30m^2. This equipment mainly serves manufacturing, maintenance and servicing segment of industry.

(a) Equipment (b) Application

Figure 1.1.: WERKLICHT® Pro in action

WERKLICHT® HD : As displayed in *Figure 1.2* the equipment consists of a mono camera and HD projector (1280 x 800 pix, 1920 x 1200 pix) system. This

is a more compact camera projector set-up *Image 1.2a* to cater different set of applications. This equipment also uses marker based tracking scheme. *Image 1.2b* shows that projection is capable of providing much more information with rich and colourful projections. However, the contrast of the projection may suffer due to nature projected surface. The range of the system is limited to 1-2m^2. This IAR system is considered to be handy in tasks like product design and rapid prototyping.

(a) Equipment (b) Application

Figure 1.2.: WERKLICHT® HD in action

CHAPTER
TWO

MOTIVATION AND PROBLEM STATEMENT

"If I had only one hour to save the world, I would spend 55 minutes defining the problem and only 5 minute finding solutions" - ***Albert Einstein***.

Following the words of the genius this chapter is dedicated to defining the motivation and the problem statement of this work. The motivation behind this research is derived from limitations faced by AR systems (See 1.3.3) in industrial scenario. The chapter is concluded with the problem statement for our research. The problems defined will serve as primary focus of this work. However, the solution (tracking methodology) may very well find its use, as a whole or with partial modification, in other AR or computer vision applications.

2.1. Motivation

Although projective IAR systems like WERKLICHT ® Pro and WERKLICHT® HD (1.3.3) are commercially available; industrial users show increasing appetite for better solutions. It is vital to note the shortcomings of the existing solutions. For ease of understanding we have mapped identified problems with previously explained *must have* IAR requirements (Section 1.2).

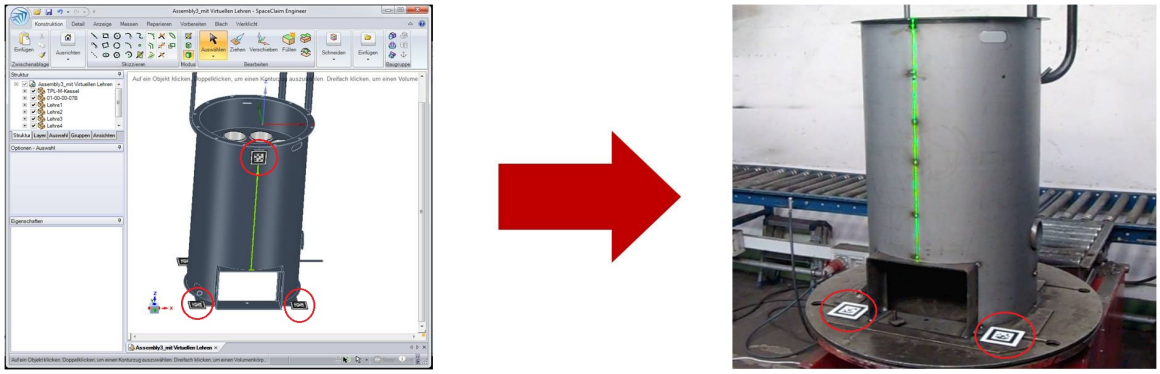


Figure 2.1.: Marker Placement

2.1.1. Marker Placement : *Work Flow Integration, Usability*

In *section* 1.3 we mentioned the marker based tracking methodology used by our projective IAR systems. A marker is an object of known geometry (3D or 2D/Coplanar) added to the scene. The special pattern on the marker helps in determining its unique identification. Its orientation in space is computed using perspective invariant features on the pattern. Accuracy of augmentation highly relies on placement of the markers in the scene. *Figure* 2.1 gives us a pictorial overview of this process. Position of the marker on target object is first decided with help of a 3D CAD model of the target object. The user has to decide the position such that the markers get maximum visibility when seen from different perspective. Careful marker placement enhances the operating space of the equipment. The image also shows a laser projection done on the industrial object with help of marker tracking.

Marker placement is a crucial step which directly reflects upon the final augmentation results. At the same time marker placement requires additional effort from the user. However, the task (welding in this case) can still be performed without the augmentation system. An important lesson to be learnt is that system should adapt to the existing work flow of the user. Industrial customers rarely show willingness towards investing extra time and effort to accommodate additional device into their work flow. Hence we can safely claim that a solution to this problem can gain projective IAR wider acceptance with industrial users.

2.1.2. Ease of Learning : *Usability*

System set up and marker placement also demands a fair bit of skill from the user. Also, need to readjust markers may arise in case of inferior augmentation. The inability to gauge reason of erroneous performance may result in unnecessary loss of production time, material and money. If targeted users are only acquainted to labour oriented jobs, then cultivating understanding of dealing with CAD data, marker positioning or manipulation is time consuming. Therefore, a *plug and play* type of system is likely to please users (expert or novice).

2.1.3. Marker and Equipment Size : *Usability, Robustness, Scalability*

Marker occlusion (line of sight) or poor visibility (at large working distances) related issues almost always make the equipment struggle to provide uninterrupted tracking results. In order to minimize this problem we may either use bigger markers or higher number of markers. Consequently, larger marker size or quantity may block workspace of the user and introduce additional operational limitations for user. IAR Systems like WERKLICHT® Pro (1.3.3) use stereo camera set-up to enhances field of view, which gives advantage in tracking, but makes equipment more bulky hurting mobility of the system.

2.1.4. Registration : *Accuracy, Robustness*

Azuma [Azu97] identifies registration of virtual information as one of the key factors of an AR system. IAR systems (*Section* 1.3.3) under consideration claim projection accuracy in

sub-millimeter range [EXT]. However this measure highly depends on knowledge of 3D position of each marker with respect to the target object. Markers are placed additionally on the target body and their relative position in the object co-ordinate system is calculated with help of CAD model. This scheme is prone to both *static* and *dynamic*[Azu97] errors. The markers are usually not designed with high precision, and its thickness is avoided or wrongly estimated which is an example of static error in 3D position of the marker. Marker displacement might happen due to vibrations on shop floor or object deformation, which can introduce dynamic error. Hence a system can boast of higher registration accuracy if accurate position of markers can be repeatedly checked or updated with ease.

2.2. Finding a feasible solution

This section will put afore mentioned problems on centre stage and discuss possible solutions. An astute observation of the problem set suggests that existing tracking approach can be improved. In current methodology multiple problems are attributed to use of specialised markers. Moreover these markers are alien to industrial environment and used only by the tracking system. Therefore, to overcome some limitations of existing approach a new tracking approach is required.

Lepetit [LF05] provides an exhaustive survey of monocular 3D model based tracking methods for rigid objects. In this study the author has discussed both marker and marker less tracking methods, and recommendations are given for selecting a suitable tracking method based on the application.

2.2.1. "Why marker less tracking is not suitable?".

The computer vision community has been very active towards developing marker less 3D object tracking solutions. These methods use localized feature points, edges or corners of object for tracking. Few traits (mentioned below) of marker less methods make them really attractive for an IAR system.

- ***Natural Marker*** : No specially designed markers required, uses features in the scene for tracking.
- ***Process Integration*** : No marker set up or manipulation required.
- ***Occlusion handling*** : Using multiple features avoid losing augmentation over line of sight problem, unless complete tracking object is not visible.
- ***Model based tracking*** : Many impressive solutions exist especially for polygonal or textured objects.

The points mentioned above make a strong case for using marker less tracking. The advantages can clearly improve *work flow integration*, *scalability* and *usability*(1.2) of the IAR system. Multiple approaches like SIFT [Low99], SLAM [CGCMC07], SFM [MYNW09,

PC02], and N3M [HBN07] can be given a good consideration. However, some of the requirements of Industrial environment add complexity towards use of marker less methods. One or more objects of same appearance might exist in the scene which may create problems for marker less methods. Additionally, manufacturing cites often have repetitive structures at different places, therefore unique identification is easier through markers. On the other hand marker less methods do not allow for selection of features to be used for tracking, which may hinder in some cases where object assembly is changed or shape is deformed over time. In such cases it is more useful to have a marker dependent solution.

2.2.2. "Why a marker based, but a new tracking approach?"

A new tracking approach without use of marker less methods naturally leads us to decide upon a new marker based approach. Traditionally marker based tracking has been widely popular in AR as both detection and pose estimation is fairly easy and fast with markers. Characteristics like *High Accuracy, Robustness* and *Fast performance* built up a strong reputation for markers in AR. Marker dependent tracking systems are able to support tracking by detection approach, which means each frame performs marker detection. Now, a study of current state of the art will help us understand need of a new marker based tracking method.

A marker can be mainly categorized in to two types *Active* and *Passive* [BKEC]. Active markers (e.g. magnetic, light) have ability to communicate its position with the tracking device, whereas passive markers are uniquely designed 2D (coplanar) or 3D patterns which have to be detected through vision methods. We intend to use a monocular system for tracking, hence a suitable choice of passive markers is required for designing a new approach .

Figure 2.2 shows various popular planar (2D) patterns that have been presented by various researchers. The marker pattern consists of perspective invariant features for accurate pose estimation. A single planar marker is generally enough to obtain 6-DOF pose estimation with good accuracy. In single or multi marker environment the special patterns function as a unique identifier of the marker. Currently, our IAR system uses encoded Circular and square markers similar to ones presented by Diego [IMH02] and Kato [KB99] respectively. However, any new planar marker based approach is likely to face issues similar to those existing in current method (2.1.1).

Figure 2.3a shows infra-red (IR) camera based tracking using 3D retro-reflective spherical markers[PK07]. These 3D markers are commercially available [ART](Inset Image2.3a) and can provide very high accuracy in 6-DOF pose estimation [LF05]. In this design each fiducial represents a single point match, at least 4 points are required for pose estimation with such systems. There markers are also referred to as *Point* markers. It should be noted that mainly such systems are used with outside-in tracking scheme. Moreover, it is not feasible to use retro reflective markers without having IR cameras. Image 2.3b shows 3D patterns created by using planar pattern for calibration of camera [Zha01]. A 3D pattern, although good for camera calibration would require more space in the scene. The change

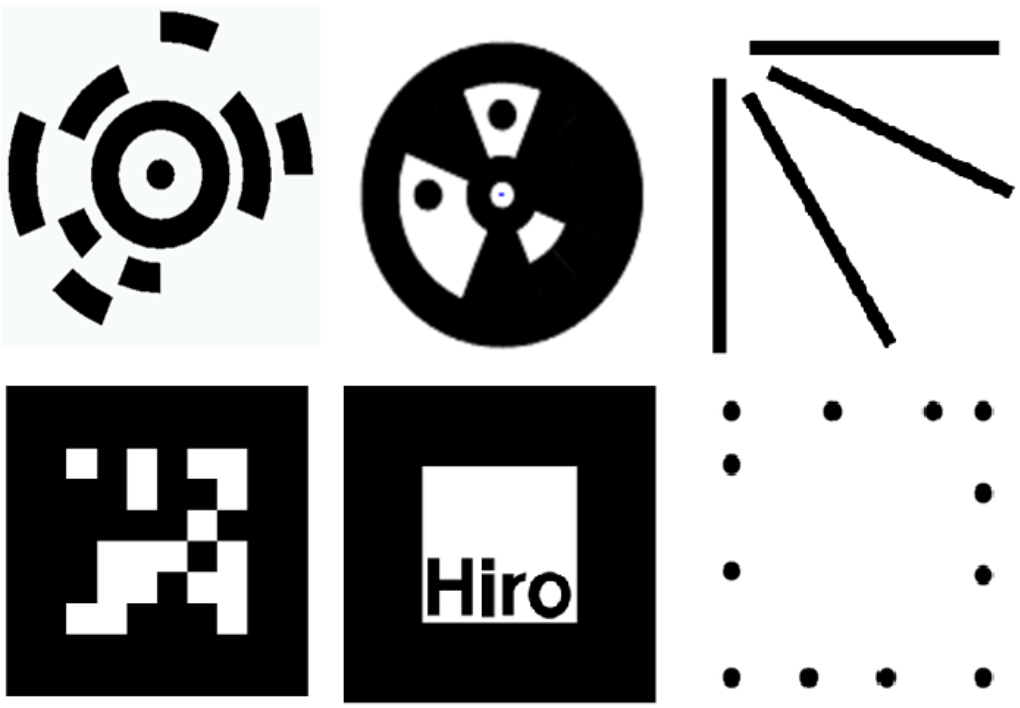
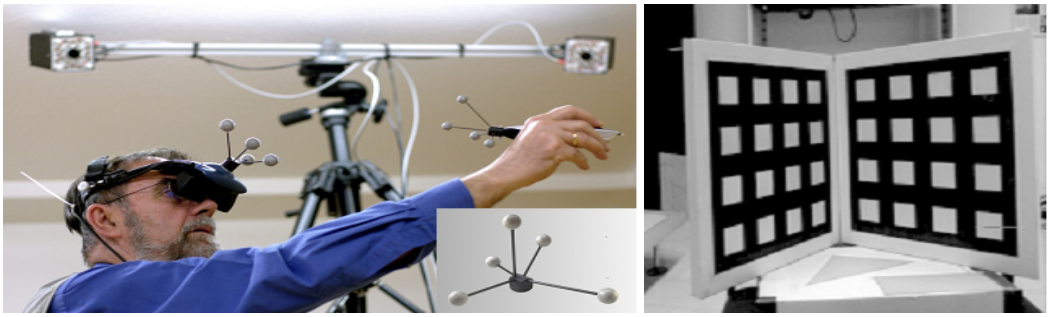


Figure 2.2.: Different Marker Patterns :
TRIP[IMH02], Circular Fiducial[NF02], Line Pencil[RM04] (top Left to Right).
ARTag [Fia05], ARToolKit[KB99], pi-marker [BAT11] (bottom left to Right)

required to accommodate this approach is not convincing as problems related to *set-up* and *registration* (Section 2.1) are likely to exist. After considering multiple options at hand we conclude that a new type of solution is required for industrial environment.



(a) IR marker Tracking [PK07] (Inset : Retro-reflective marker [ART])

(b) 3D Calibration Pattern

Figure 2.3.: 3D marker Patterns

2.2.3. "Why use Circular Markers?"

The choice of circular markers (*Image 2.4a*) has its roots in industrial work flow itself. Close range photogrammery is an integral part of industrial *Metrology* (science of measurement). It is a technique of measuring geometric properties of objects using photographic images. Circular markers are commonly used for various manufacturing and quality assurance related measurements. GOM [GOM], Capture3D [Cap], AICON [AIC] are some of the companies providing photometric solutions to multiple industries (Automotive, Aerospace, Medical etc.).

There are two types of Circular markers used in photogrammetric application. Retro reflective Circular markers are popular for high speed motion capture and 3D analysis of industrial models. These markers are expensive due to their high precision manufacturing costs. The other type of Circular markers used in photogrammetric applications are paper based markers (*Image 2.4a*). These markers are cheaper alternative for 3D measurements as their production is inexpensive and no specialized equipment (IR cameras) is required for computing measurements.

Figure 2.4 shows an example of how 3D measurements are done using Circular markers in photogrammetry. A high resolution commercial camera (e.g. SLR), some coded Circular markers and a calibrated object of known geometry are supplied with photogrammetric measurement software. Multiple Circular markers and some coded markers are attached to the object (See *Figure 2.4b*) in no specific pattern. Moreover, one calibration object of known geometry is also added to the scene (normally a bar with coded markers). Multiple images of the object are taken from various perspectives by using the camera. This images are supplied to the measurement software, using triangulation techniques 3D information (Centroid position, Surface Normals etc.) of each marker is recovered. Calibration object and coded markers are usually added to achieve precise measurement without scaling error [GOM]. Once measurements are done the coded patterns are removed from the scene. A detailed account on photogrammetric techniques and applications, including use of circular markers is given by Luhmann [LRH06].

(a) Circular Markers [Cap]

(b) Typical Set-up for measurement of circular markers

Figure 2.4.: Ciruclar markers used in Industrial photogrammetric applications

Considering wide spread use of Circular markers in industrial measurement applications, it is beneficial to use the same markers for tracking. Each marker can serve as a *Point* marker, multiple such markers in same coordinate frame create a single 3D marker. There can be several advantages of using circular markers to over come a few shortcomings of current tracking approach (See 2.1).

- *Marker placement* : Users will be comfortable with using such a system as it is already part of their work flow. Additional effort is not required from the user to learn a new marker placement method. Additionally, as an IAR manufacturer no effort is required to provide marker measurement systems.
- *Ease of Learning* : The AR system may not require any set-up if markers are already attached (*Plug and Play*). If marker set up is desired only for tracking system even then process of measuring 3D information is known, Therefore virtually no new training is required from the user.
- *Marker and Equipment Size* : Marker is small enough (∅ (mm) :5,8,10) to co-exist in working environment without obstructing work space. Multi marker pattern as a whole can be used as one single 3D marker for accurate 6-DOF pose estimation (like *Point* markers). Engineered environment is not required to accommodate such tracking system in the scene. Moreover, monocular camera tracking can allow the IAR equipment to become more compact, mobile and less expensive.
- *Registration* : 3D position of the markers is computed in object coordinate system with very high accuracy, which is a vital requirement for accurate registration of augmented data. Additionally surface normal information is also provided by the measurement software. This information can also be used to improve accuracy of the pose estimation.

2.3. Problem Statement

As discussed in previous section Circular marker based tracking shows potential to reduce limitations of the existing IAR systems. This approach is also expected to satisfy previously

defined industrial requirements (1.2). The main focus of this thesis, that is to create a Circular marker based tracking method for Industrial IAR. The two main problems to be solved are Marker detection and identification.

- **Marker Detection** : This problem refers to the challenges in terms of detection of markers in the images. Although, a circle is a primitive shape for detection, problems with industrial lighting, textured background, and specular surfaces make detection a challenging task. Moreover, small size of markers make the task of marker detection very difficult.
- **Marker Identification**: This problem is known as *Correspondence matching* problem in computer vision. This problem is introduced as result of a trade-off by not using conventional markers with specialised patterns identification. *Figure 2.5* gives pictorial representation of this problem. All markers have identical appearance in the image, therefore identifying each marker in the image with its 3D counterpart is a must for accurate pose estimation. This problem can be solved by using epipolar constraints for a stereo system, but our focus is to solve this problem for a monocular camera. Additionally, tracking by detection approach would require an efficient solution for matching problem.

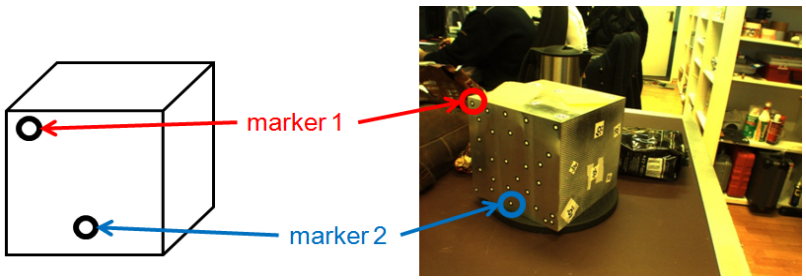


Figure 2.5.: Marker Identification Problem

Providing a method for real-time circular marker tracking is one of the key challenges of this project. Time performance gives a vital edge to the user as compared to existing off-line solutions. In the next chapters we will focus on solving the above mentioned problems, at the same time keeping the final goal of establishing a complete tracking solution in mind we will design a suitable work flow for the whole system.

CHAPTER
THREE

TRACKING WORK FLOW AND THEORY

The aim of this chapter is to define a basic work flow for the new tracking approach and provide the reader with relevant literature review. The work flow will be divided in to different functional processes. The requirements (Input) and the outcome (Output) of each process will be defined. Based on these requirements relevant theoretical concepts will be discussed. The processes directly related to the problem statement (Section 2.3) will be discussed in detail, and a suitable approach will be suggested. The reader should note that this chapter will provide only the overview of the required theoretical concepts, Implementation specific details will be covered in part II of the thesis.

3.1. Basic Work flow

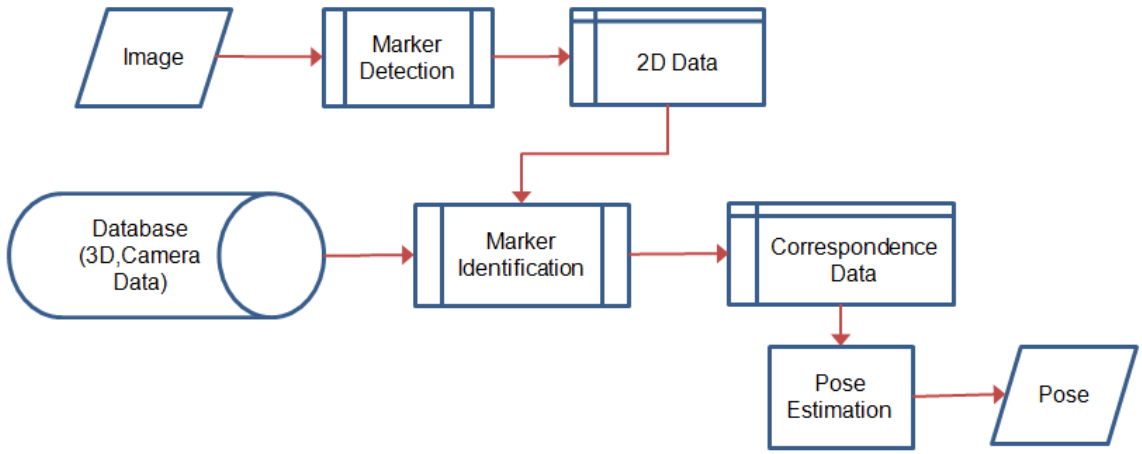


Figure 3.1.: A simplified work flow of the Tracking process

Figure 3.1 shows the basic work flow of the tracking process. The figure shows that once image is captured it is supplied to Marker Detection process to extract 2D information of

markers. After 2D information is generated, Marker Identification process will identify corresponding 2D-3D marker pair. These pairs will be used to compute pose of the object with respect to the camera. It is assumed that 3D measurements related to marker placement are performed independently and measurement data is available to the work flow. The Camera used for tracking is calibrated and intrinsics are available with the user. A short introduction to the terms, processes and their functionalities is given below.

Image : This term refers to the two dimensional photographic image of the object captured by the camera. It is given as an input for the tracking work flow. The reader should note that the image provided to the work flow is an undistorted Image.

Database : This term represents all the preprocessed information that is readily available to the tracking work flow. This information includes details regarding marker position (in *Object coordinate system*) and camera calibration (pin hole Camera model). The 3D information of the Circular markers will be referred as *3D Data*, and the term *Model Points* will be used to refer circular markers on the object. *3D Data* contains following details for each *Model Point*.

1. 3D Position : $M_i(X_i, Y_i, Z_i)$ (in mm)
2. Surface Normal : $N_i(Nx_i, Ny_i, Nz_i)$
3. Radius : R_i (in mm)

Note that all the measurements are done with respect to a reference co-ordinate system. We will term this as *Object co-ordinate system*.

Marker Detection : The function of this process is to extract the 2D information of Circular markers from the image. Based on knowledge of the marker geometry (circular) and design (white circle with black border) this process will detect markers from the image and provide information as *2D Data*.

2D Data : This term refers to the outcome obtained from the process of Marker Detection. 2D Data consists of centroid position and contour information of each marker. We will refer each marker as *Image Point*(m_i). Each Image point will have following information;

1. Centroid $m_i(x_i, y_i)$
2. Contour as C_i.

Marker Identification : The aim of this process is to solve the Marker Identification problem. The 2D Data and 3D Data are supplied as input this process. The correspondence relationship between *Model Points* and *Image Points* ($M_i \leftrightarrow m_i$) is the out come of this process. Correct correspondence matching is absolutely essential for accurate pose estimation process.

Correspondence Data : The term refers to the outcome of Marker Identification process. The 2D-3D correspondence ($M_i \leftrightarrow m_i$) information is used to compute the pose of the target object.

Pose Estimation : Using correspondence matching ($M_i \leftrightarrow m_i$), position and orientation of the camera with respect to the object is determined with this process. The output of this process is the Pose of the object.

Pose : This term refers to the final outcome of the tracking work flow. Pose allows us to transfer the Model Points from object co-ordinate system to the camera co-ordinate system. The Pose is expressed in form of a Rotation (R) and a Translation (t) matrix.

3.2. Marker Detection

In this section we will discuss the methods available to achieve the defined functionality. A series of image processing operations are performed on the image to obtain required 2D Data. It is also important to understand the challenges and the expected outcome. The understanding of these factors influences the selection of appropriate image processing methods.

Conventionally markers used in AR are designed specially to reduce exhaustive image processing efforts. Markers like ARToolKit, ARTag or TRIP (See *Figure 2.2*) are designed such that they can be easily separated from their background and identified quickly to support real-time applications. Encoded pattern of the marker is also used for verification of the detected results. However, the Circular markers are not specifically designed for tracking purposes, therefore they lack of distinctive features. The known features of the markers are listed below.

- Shape : The Circular markers are circular in shape. Their projection in the image will always be in form of circular or elliptical curves.
- Color :A Circular typical marker (See *Figure* 2.4a) consists of a white circle with a black inner edge. The markers also has an outer edge of a circular or a square shape. However, the outer edge of the marker is not considered as a distinctive feature. This is due to the fact that the outer edge will be difficult to extract when the target object surfaces are dark.

All possible information (m_i, C_i) about the marker is extracted from the image. This information is further used by the correspondence matching process.

The Industrial environment offers challenges like varying light conditions, object background and reflective surfaces. The camera used for the target application produces high resolution images of the scene. A high resolution image is able to capture the markers with higher detail. On the other end higher image resolution implies that large image data has to be processed, which may result in increased detection time. Interactive AR solutions

demand low latency rates and therefore a trade-off is required while selecting the image resolution.

Figure 3.2 shows the sequence of image processing operations proposed for the *Marker Detection* process. The colour image is converted to a binary image after *Binarization* stage. *Contour Detection* stage detects the contours from the binary image, and *Contour Rejection* stage rejects the irrelevant (non-elliptical) contours. Finally, *Ellipse Fitting* is performed on the contours to obtain *2D Data*. A short description on each stage is given in following part of this section. Reader should note that Naimark [NF02] and Diego [IMH02] have proposed similar work flow for detection of the Circular markers.

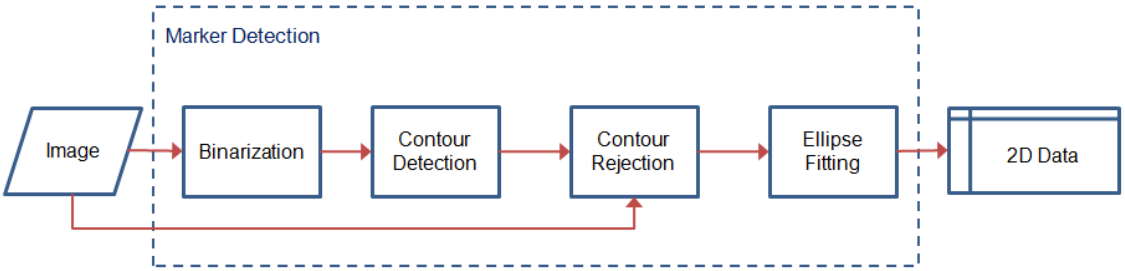


Figure 3.2.: Marker Detection work flow

Binarization : In this stage image is converted from a colour image to a binary image. This is done because it is easier to segment contours from binary images. Normal thresholding operation is used by various AR markers [KB99], however this method fails when marker patterns are less distinguishable. Adaptive thresholding [Wel93] is robust to illumination variation and therefore is a good choice to perform this conversion.

Contour Detection : In this stage all possible contours are extracted from the binary image. Nairmark [NF02] and Diego [IMH02] show use of edge detection approach for processing binary images and selecting circular markers. We suggest the approach given by Suzuki [SA85]. Additionally, Topological information of contours is also preserved.

Contour Rejection : In this stage non-elliptical contours are rejected based on various parameters like shape, size, pixel intensity value etc. In this stage information from the original image (Colour or Gray scale) can be used to access pixel intensity values for contour rejection. Naimark [NF02] explains that in daylight conditions white paper intensity can vary from 70 to 255 (in 8 bit Gray scale image), and intensity of black pixel varies from 20-50. However, in low light (e.g. basement) or high light (e.g near light source or window) intensity values of white and black might vary accordingly. Topological information obtained in earlier stage can provide an additional rejection criteria based on child-parent relations of the contours. Contours of the Circular markers are expected to have no children in their topological hierarchy.

Ellipse Fitting : Many researchers have demonstrated effective ellipse fitting methods from a given set of points. *Direct Least Square* method presented by Fitzgibbon [FPF99] is one of the popular methods. In this approach conic parameters are calculated such that distance of each contour point $p(x, y)$ from the conic is minimized. The result of fitting is influenced by quality of the input, therefore it is recommended that contour points are refined at sub-pixel level using edge gradients. A general conic representation is given as,

$$f(A_{conic}, I) = A_{conic}.I = ax^2 + bxy + cy^2 + dx + ey + f = 0. \tag{3.1}$$

Where, $A_{conic} : [a \quad b \quad c \quad d \quad e \quad f]$, $\mathrm{I} : [x^2 \quad xy \quad y^2 \quad x \quad y \quad 1]^T$. $f(A_{conic}, I_i)$ is called "algebric distance" of a contour point $p_i(x, y)$ to the conic $f(A_{conic}, I) = 0$. The fitting of a general conic is obtained by minimizing the sum of squared algebric distances of N contour points from the curve.

$$D_A(A_{conic}) = \sum_{i=0}^{N} f({I_i}^2) \tag{3.2}$$

Here, to avoid a degenerate solution $A_{conic} = 0$, Fitzgibbon proposed a constraint $4.a.c - b^2 = 1$. The advantage of this constraint is that the resultant curve is always an ellipse.

After ellipse fitting, the centroid of the ellipse can be computed in multiple ways. Luhmann [LRH06] provides multiple techniques for accurate circle measurements from contour points. Star operator and Zhou operator are the methods which belong to structural measuring approach. According to this approach estimated centre points are derived from mathematical conic equation obtained from each contour points. Eccentricity (e) [LRH06] should be taken into account while measuring centroid of the ellipse. Eccentricity is the difference (error in this case) between the measured ellipse centre and the projected circle centre.

$$e = r_m - (c/2)\left(\frac{R_m + (d/2)\sin(90-\alpha)}{Z_m - (d/2)\cos(90-\alpha)} + \frac{R_m - (d/2)\sin(90-\alpha)}{Z_m + (d/2)\cos(90-\alpha)}\right) \tag{3.3}$$

Where e : *eccentricity of projection*
d : *target diameter*
r_m : *image radius of projected target*
α : *viewing direction, angle between image plane and target plane*
R_m : *lateral offset of target to optical axis*
Z_m : *distance to target*
c : *camera focal length*

Luhmann [LRH06] explains that for small sized markers and large operating distances eccentricity (e) factor ($e < 0.5\mu m$) is considered negligible. In one of his experiments, Luhmann demonstrates that a marker of $d = 5mm$, at different object distances (Z_m = 1 m to 6 m) produces image diameters between 4-20 pixels (pixel size $10\mu m$). The maximum e recorded for this experiment was 0.27μm (other constants $c = 40mm, \alpha = 60°$). Therefore, neglecting the effect of e in centre measurements of circular markers can be considered.

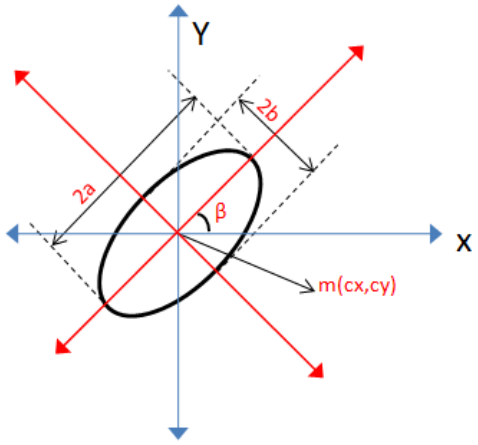


Figure 3.3.: Ellipse Parameters

Moreover, computation of e is not possible for Circular markers as several parameters (r_m, α, R_m, Z_m) are unknown. If required this correction can be done after computing the object pose.

2D Data : This term refers to the end result obtained from the *Marker Detection* process. This data contains position and structural information of each marker detected in the image. Conic representation in a *3x3 Matrix* (C) (eq. 3.4) form is derived from A_{conic} (eq. 3.1), and Ellipse parameters (E) (eq. 3.5) are calculated from the conic C. The method of calculating ellipse parameters (E) is given by Farin [FH98] (See Appendix). *Figure 3.3* gives pictorial representation of Ellipse parameters given in eq 3.5.

$$C = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}, p_i C p_i^\top = f(A_{conic}, I) = 0 : \text{Where, } p_i = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}^\top . \quad (3.4)$$

$$E = (maj_a, min_b, \beta, cx, cy) \quad (3.5)$$

Where, maj_a : length of semi-major axis (a)

maj_b : length of semi-minor axis (b)

β : Rotation angle, (Betweenm Major Axis and X-Axis)

cx : ellipse centre x image co-ordinate

cy : ellipse centre y image co-ordinate.

3.3. Marker Identification

This process is expected to solve the 2D-3D Correspondence matching problem. The correspondence problem is generally not observed in marker based tracking systems due to presence of unique identification code on the marker. Typically markerless tracking methods focus on the correspondence problem, therefore observing relevant methods can provide a useful direction towards solving the Correspondence problem for circular markers.

Figure 3.4 shows the work flow of the *Marker Identification* process. In this process *Feature Descriptor Generation* stage generates *2D-3D Feature Descriptors* from both 2D and 3D data. *2D-3D Feature Descriptors* are matched using a *Matching Strategy* to obtain the *Correspondence Data* ($M_i \leftrightarrow m_i$). In case of Circular markers,

- 2D Data : Conic Matrix C_i and 2D *Image Points* m_i.
- 3D Data : Surface normals N_i, 3D *Model Points* M_i and Radius R_i.

It is important to understand how to use the existing information to obtain the matching hypothesis. The *Image Points* are projections of *Model Points* from *Object coordinate system* to the image plane ($\mathbb{R}^3 \rightarrow \mathbb{R}^2$) . The relation between *Image Points* and *Model Points* depends on the nature of transformation. The nature of the transformation determines which geometric invariants can be extracted from the image. In the current application we deal with projective transformation and therefore only few invariants are preserved in the image. The text of Hartley and Zisserman [HZ03] is suggested for detailed readings on invariants and transformations related concepts of projective geometry.

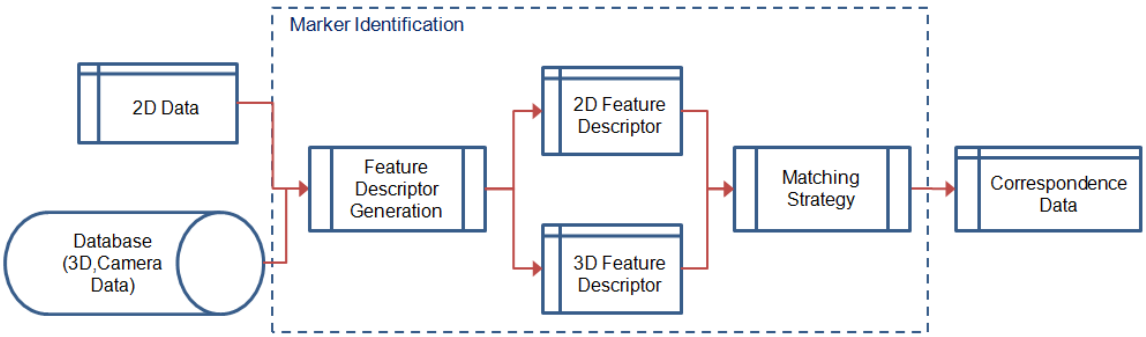


Figure 3.4.: Marker Identification work flow

3.3.1. Feature Descriptor Generation

In this stage of tracking process feature descriptors are generated from the available 2D and 3D data. The problem at hand requires extracting invariant features from the image and creating a feature descriptor. These descriptors should be invariant to object pose, perspective or camera intrinsics. For the given application of Circular markers two possible methods can be considered to generate feature descriptors,

- $2D - 2D_j$ *feature descriptor*
- $2D - 3D$ *feature descriptor*

$2D - 2D_j$ ***feature descriptor*****:** We will term this approach as a *Catalogue* based approach. In this approach the problem is reduced from 3D feature matching to $2D_j$ feature matching. Multiple images (j) of the tracking object are captured and projections of 3D features (n) on the image are recorded (in $\mathbb{R}^2$). These images are captured from known camera positions, hence 2D-3D correspondence is available. Feature descriptors (V) are computed from all *Catalogue* images and stored in the database. This computation is done only once. Representation would look like

$$\begin{bmatrix} M_1, N_1 \\ M_2, N_2 \\ .. \\ .. \\ M_n, N_n \end{bmatrix} \longrightarrow \begin{bmatrix} V_{1,i} & V_{1,i+1} & .... & V_{1,j} \\ V_{2,i} & V_{2,i+1} & .... & V_{2,j} \\ .. & .. & .... & .. \\ .. & .. & .... & .. \\ V_{n,i} & V_{2,i+1} & .... & V_{n,j} \end{bmatrix}$$

When a new image is captured, features are detected, descriptors are calculated (v_i) and matched with the descriptors in database (V_i). If matching descriptors are found, relevant 3D correspondence is obtained from the database. It can be represented as,

$$\begin{bmatrix} m_1 \\ m_2 \\ .. \\ .. \\ m_n \end{bmatrix} \longrightarrow \begin{bmatrix} v_1 \\ v_2 \\ .. \\ .. \\ v_n \end{bmatrix}$$

Kolomenkin [KPSL] proposed a *Catalogue* based method for tracking stars in order to determine orientation of the satellites. In this method an image of the stars is captured via a camera placed on the satellite and identification of each star is done by using a star catalogue. Angular distances between stars are perspective invariant, therefore they are used for identification of stars. The invariants are calculated from the stars detected in the query image and compared with catalogue images. A voting algorithm is used to produce the final matching results.

Kurz [KOB] presented a similar approach, only exception being that the catalogue images are synthetic. The target geometry information is used to create images for catalogue. Local set of features are determined from the image and matched with the features from the database. The synthetic images and descriptors are created offline to speed up the performance. *Figure 3.5* shows how a catalogue might look like when images from different perspective are captured for creating a database.

To realize this approach invariant properties for the circular markers have to be identified. The markers individually can be considered as planar due to their small size. Features can be extracted from each marker (planar conic) individually, however the matching problem remains unsolved as all the marker are exactly identical, and therefore provide similar photometric and geometric information. Moreover, two or more markers are not likely to be on the same plane so as to extract coplanar conic invariants described

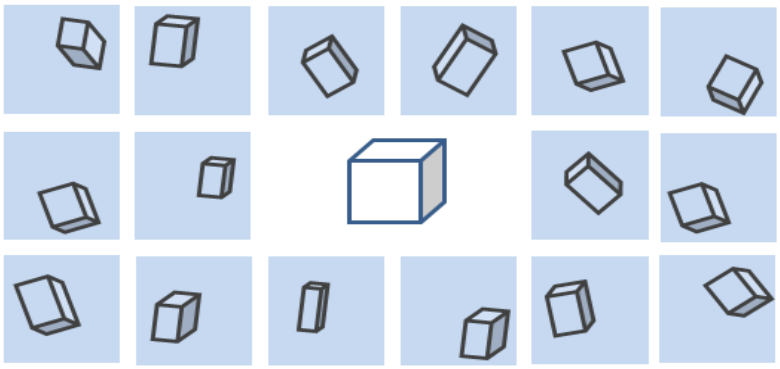


Figure 3.5.: Image catalogue example

by Forsyth [FMZ+91]. On the other hand matching the image features with features in database is an exhaustive approach. Therefore this approach may be useful in case of planar surfaces, but may not meet performance needs of our application.

$2D - 3D$ ***feature descriptor*** In this approach direct matching of 2D and 3D features is performed. A set of identical feature descriptors are generated from both *2D Data* and *3D Data*. Hinterstoisser [HBN07] provides a relevant approach N3M, where feature descriptors are created from both geometric and photometric learning of the object. A localised coordinate system is generated from each point in a small neighbourhood to create a set of N3M, using these N3Ms image features are matched directly with the features of corresponding 3D points. This approach holds true for a dense set of features, as a linear fronto-parellel projection matrix (parallelism preserved) is used to approximate a pose of the object. This approach is not suitable for Circular markers as the *Model Points* are a sparse set of points and parallelism is not preserved in sparse points.

In early 90s, Forsyth [FMZ+91] presented a concept for extracting Eucledian invariants from image projections of two non-coplanar circles. Forsyth explains that for three dimensional objects, the invariant descriptor consists of Euclidean invariants rather than Projective invariants. A three dimensional object preserves its Euclidean properties under Euclidean motions. It is harder to extract these invariants due to perspectivity, however some invariant properties can be extracted from curves existing in a single image. A 3D object can be considered as a set of rigidly coupled planes. Forsyth proposed a method which uses conic projection of a circle to extract orientation of the circle plane in 3D. The image conic can be back projected in to a three dimensional space to recover surface normal (in camera coordinate system) of the circle plane. It is important to note that the conic back projection always produces two solutions for surface normals. Additionally, if the radius of the world circle is known, the distance of the circle plane from the camera is also recovered (in *Camera coordinate* system). The mathematical information on conic back projection is provided in the Appendix. However, work of Forsyth [FMZ+91] and Diego [IMH02] is recommended for detailed understanding of conic back projection method.

Forsyth explained that following invariants can be extracted from a pair of non-coplanar

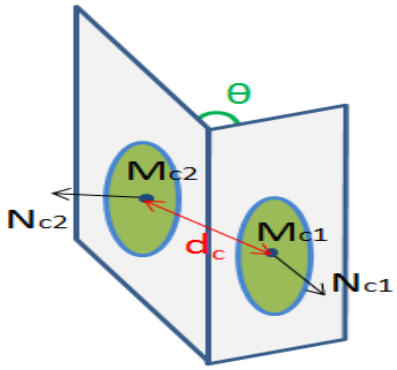


Figure 3.6.: Invariants between a conic pair

circles (Refer fig. 3.6),

1. **Angle between surface planes** (θ) : This invariant can be recovered even without the radius information. The angle between surface planes is nothing but the angle between their surface normals $\angle(N_{c1}, N_{c2})$. Normals can be recovered from conic image without knowledge of circle size in real world.

2. **Ratio of circle radius to distance between circle centers**: This invariant can also be recovered without radius value.

3. **The vector joining circle centers** : This vector $V(M_{c1}M_{c2})$ has three degrees of freedom,
   - d_c : The length of the vector (Object scale is required), $d(M_{c1}, M_{c2})$.
   - γ_{n1} :The angle between vector and normal of first circle, $\angle(V, N_{c1})$.
   - γ_{n2} : The angle between vector and normal of the second circle, $\angle(V, N_{c2})$.

It is to be noted that the surface normal and circle centre is obtained with a two fold ambiguity (two solutions for each) and therefore between a pair of conic 4 possible solutions are computed for each invariant. Experiments of Forsyth on real images convey that quantity d_c is consistent for all four solutions and only one solution exists with correct values for θ, γ_{n1} and γ_{n2}. The experiments also show that the invariant θ (although ambiguous) is less prone to errors, and the recovered surface normals are stable. Forsyth also explained that γ_{n1} and γ_{n2} are weak invariants as they are functions of recovered normals and centre positions. The quality of conic fitting method or contour point measurement affects the stability of invariants like γ_{n1} and γ_{n2}. The reader should note that ambiguity of solutions and stability of invariants is covered in following chapters.

In case of circular markers attached to rigid 3D objects, the Euclidean invariants can be extracted using their conic projections. Given that more than one marker is detected in the image, a descriptor can be generated from a set of suitable invariants for each pair of image conic. The descriptors can be generated from a single image and does not require exhaustive calculations. Moreover this approach allows maximum use of the available *3D*

Data (position, normals and radius). The additional advantage of using this approach is that descriptors are similar for both 2D (with ambiguity) and 3D data, with only constraint being that world circles are of same size. Considering above mentioned arguments use of conic based Euclidean invariants is favoured for designing the new tracking approach.

3.3.2. 2D Feature Descriptor

This term refers to the feature descriptors that are generated from the *2D Data*. The descriptor depends on the choice of method used in Feature generation stage. Since use of conic based Euclidean invariants is suggested, the descriptor are created for a pair of conics.

3.3.3. 3D Feature Descriptor

This term refers to the feature descriptors created from the *3D Data*. These descriptors are computed offline and stored in the Database. Each descriptor represents a pair of Circular markers, the 3D descriptors contains no ambiguous information.

3.3.4. Matching Strategy

This stage is one of the most important stages of the *Marker Identification* process. At this stage, the 2D and 3D descriptors are compared to generate a hypothesis of the 2D-3D correspondence. The matching hypothesis is termed as *Correspondence Data* ($m_i \leftrightarrow M_i$) which is used for the pose estimation process. A single error in correspondence matching leads to wrong estimation of the pose, therefore 100% accuracy is expected from this process. Designing a highly accurate matching strategy is one of the main focus of this thesis.

For the chosen method of conic invariant descriptors, multiple factors should be considered to design a matching strategy. As explained earlier a 2D descriptor may consist of 4 solutions for each invariant entity (i.e. $\theta, d_c, \gamma_{n1}, \gamma_{n2}$). These descriptors are to be matched with 3D feature descriptors in a computationally inexpensive manner. The complexity increases as *2D Data* may contain false marker detections, the matching should be able to reject these false positives. As the tracking object is three dimensional, not all the markers are captured in the image. However, since position of the camera is unknown all *Model Points* are equally likely candidates for the matching process. Another important detail is to understand that each descriptor is made from a pair of conics. Therefore initial descriptor matching would result in identifying a matching conic pair (i.e. A pair of circles and their possible projections). Once a pair of conic is identified, it has to be processed further to obtain a point to point correspondence between *Model Points* and *Image points*.

3.4. Pose Estimation

Pose Estimation is the final stage of the tracking process, where pose information of the object is calculated. Pose calculation from known correspondences is an extensively studied topic in the Computer Vision community, therefore we would recommend reader to follow textbook of Hartley and Zisserman [HZ03] for in depth knowledge about pose estimation methods. In this section only required information will be provided.

Figure 3.7 shows the basic work flow of this process.

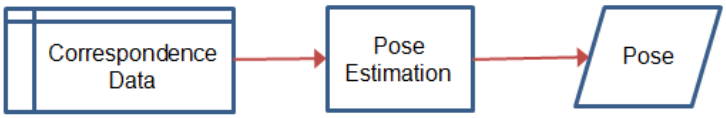


Figure 3.7.: Pose Estimation work flow

The mathematical expression for relation between *Image points* and *Model Points* is given in equation 3.6. P is a projection matrix (3x4), using which a *Model Point* can be projected on the image plane. A projection matrix is a function of the object Pose and the camera intrinsics. Pose mainly defines the position and the orientation of the object in camera coordinate system (See 3.8). Position and orientation of the tracking object is defined by $[R \mid t]$, a 3x4 matrix. Camera intrinsics are represented by a 3x3 matrix based on pin hole camera model (See 3.8).

$$s.m_i = P.M_i \tag{3.6}$$

$$P = K.[R \mid t] \tag{3.7}$$

$$K = \begin{bmatrix} \alpha_x & s & u_x \\ 0 & \alpha_y & v_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

m_i = Homogenous representation of Image Points $\begin{bmatrix} xc_i & yc_i & 1 \end{bmatrix}^\top$

M_i = Homogenous representation of Model Points $\begin{bmatrix} X_i & Y_i & Z_i & 1 \end{bmatrix}^\top$

α_x, α_y = Scale factor in x and y directions, it is function of focal length and pixel size.

$\begin{bmatrix} u_x & v_y \end{bmatrix}$ = Principal point co-ordinates on image plane.

s = Skew factor, it is non zero when x and y directions are not perpendicalar.

Correspondence Data : The $M_i \leftrightarrow m_i$ correspondence data generated by the matching strategy is used as input for Pose estimation. A Projection matrix has 12 entries and therefore 12 Degrees of freedom (DOF). If we consider the entries up-to scale then we can reduce it to 11 DOF. By solving this 11 DOF we can obtain both K and $[R \mid t]$. Each correspondence pair ($m_i \leftrightarrow M_i$) provides 2 equations and therefore 6 such pairs provide 12 equations, enough to compute a unique solution for P. Calibration matrix has 5 DOF for

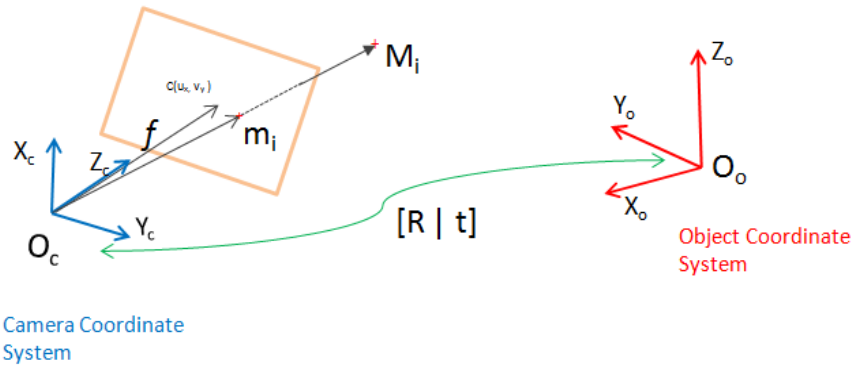


Figure 3.8.: Camera projection model : Rotation and Translation (R,t) brings a 3D point from object coordinate frame $O_o(X_o, Y_o, Z_o)$ to Camera coordinate system $O_c(X_c, Y_c, Z_c)$, and a 3D Model point M_i projected on image plane as m_i using Camera Intrinsics.

a pin hole camera model. Since Camera Intrinsics (K) are known, DOF of P is reduced to 11 - 5 = 6. This can be solved theoretically by μ = 3 correspondence pair, but with 3 correspondence pair we have 4 possible solutions [HLON91]. μ = 4 or μ = 5 gives 2 possible solutions, except when points are coplanar with no triplet being collinear the solution is unique. With $\mu \geq 6$ we can compute a unique solution for P and consequently obtain R and t. This correspondence criteria is explained in detail by Hartley in [HZ03].

Pose Estimation : The aim of this process is to compute pose of the object using correspondence information. There are various methods available for pose computation based on the number of correspondences obtained. Hartley and Zisserman [HZ03] explain computation of P using DLT algorithm. This approach works sufficiently when sufficient correspondence pair are available ($\mu \geq 6$). If camera intrinsics are known, 3 point pairs are sufficient to compute pose. This problem is also called perspective-3-point(P3P) problem. Drummond [DC02] and Harlick [HLON91] have proposed methods for solving P3P problem. These methods will recover as much as 4 real solutions for 3 point pair and unique solution for 4 or more pair (except non-collinear). DeMenthon [DD95] designed POSIT algorithm for $\mu \geq 4$, however this method does not yield reliable results in case of coplanar points.

Pose : This term is the final outcome of the tracking process, The Rotation information (R) can be represented by Euler angles or Quaternion representation. Euler angles are prone to gimble lock [DP11] problem, therefore quaternions are preferred.

Part II.

Implementation and Experiments

CHAPTER
FOUR

INVARIANT STABILITY ANALYSIS

In this chapter we will discuss the stability of Euclidean invariants generated by a pair of circles. The aim is to understand the usability of these invariants, in order to select components for the new descriptor design. Forsyth [FMZ+91] presented the concept of conic invariants, however the experiments carried out to support the proposed concept consisted of a simple arrangement of circles. Moreover, the work is two decades old, and not many researchers have explored the Invariants generated by two non coplanar circles. Diego presented TRIP [IMH02] markers that use the concept of recovering surface normals from the image conics. Size of the circles used by Forsyth and Diego for their applications were much larger than Circular markers ($\varnothing$ = 5,8,12 mm). The invariants are functions of the normal and the centre position recovered from two image conics. The error in recovery of either directly affects the quality of invariants. As this topic is less studied in the literature, we have designed synthetic experiments to understand behaviour of the conic back projection method with respect to change in camera distance and orientation.

So far we have understood that an image of a world circle can be used to define the circle plane in Camera coordinate system (with known radius and camera calibration). This process can be termed as back projection of the circle or ellipse. The back projected information includes two centre positions ($Mc_i{}^1, Mc_i{}^2$) and two surface normals ($Nc_i{}^1, Nc_i{}^2$) of the circle plane (two values for each due to ambiguity). *Figure 4.1* shows the ambiguity in a pictorial manner. $\pi 1, \pi 2$ are the recovered planes from the ellipse back projection method. For multiple circle in the image we can represent the back projected data as,

$$\begin{bmatrix} C_1 \\ C_2 \\ .. \\ .. \\ C_n \end{bmatrix} \longrightarrow \begin{bmatrix} Mc_1{}^1, Mc_1{}^2, Nc_1{}^1, Nc_1{}^2 \\ Mc_2{}^1, Mc_2{}^2, Nc_2{}^1, Nc_2{}^2 \\ .... \\ .... \\ Mc_n{}^1, Mc_n{}^2, Nc_n{}^1, Nc_n{}^2 \end{bmatrix} \tag{4.1}$$

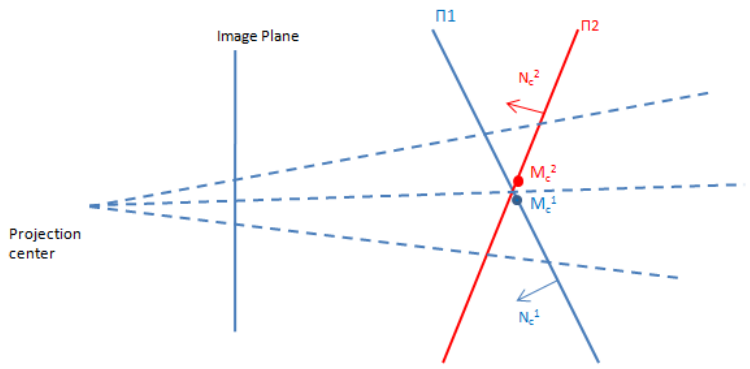


Figure 4.1.: Orthogonal view of the conic back projection ambiguity: $\pi 1, \pi 2$ are the recovered planes, ${Nc_1}^1, {Nc_1}^2$ and ${Mc_1}^1, {Mc_1}^2$ are respective normals and centre positions.

In practice outcome of the ellipse back projection method will always have some errors. The accuracy of ellipse back projection method is highly dependent on quality of the conic matrix. Image noise and lens distortion also contributes towards degrading the quality the of image conic. Poor selection of contour points or ellipse fitting method itself can also cause errors in estimation of the normals and centres. Moreover, the size of image conic is very small when object distance from camera is increased. At such distances small changes in the circle orientation are not equally reflected by the imaged conic. The stability analysis of the back projected conic information is required, the findings of such analysis can assist in effective parametrization of the matching algorithm. *Figure 4.2* gives a good overview of the ellipse back projection method. Camera intrinsics are used to compute the normalised conic matrix from image conic C_i. In the subsequent stages a rotation matrix (R') is computed for rotating the image plane such that it becomes parallel to the object surface. Surface normals and circle centre positions are computed in camera coordinate system using R' and the circle radius. The mathematical representation of the same is explained by both Forsyth [FMZ+91] and Diego [IMH02] (See Appendix).

4.1. Simulation & Experiments

We chose MATLAB as the simulation tool for our experiments. *Figure 4.2* shows that the camera intrinsics (K) are used by the process of conic back projection. Therefore, errors in estimation of calibration parameters may also degrade ellipse back projection results. This factor is avoided in simulated experiments as the camera intrinsics are fixed for generating the ground truth data. Controlling and manipulating orientation of real world objects or cameras (e.g. Rotation of 1°) is very difficult to achieve without high precision equipments. Such manual manipulation errors are also avoided by using digital simulation. Monte Carlo simulations are performed over a large number of synthetic data to gain better understanding of the back projection results. The details of experiments and their respective goals are outline in the following part of the section.

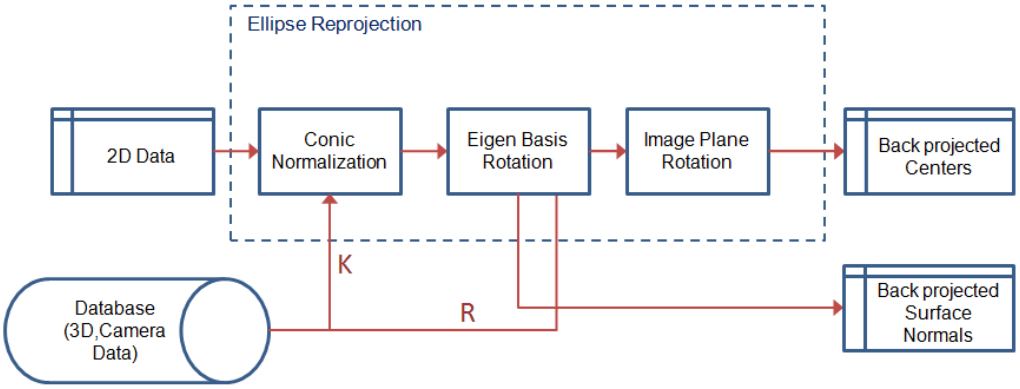


Figure 4.2.: Ellipse back projection stage

4.1.1. Parameters to examine

Focus of these synthetic experiments is to create an error model for visualising the results of surface normal and centre position estimation from an image conic. A single circle is generated with different configurations to obtain the required error model. This is valid as similar results are expected from a circle of same size and under same conditions (i.e camera orientation, image noise). Following parameters are computed in order to express the error model,

1. Normal Estimation Error (η): From the recovered normals only one of the solutions is correct. For simulations the camera orientation can be predefined and therefore recovered normals ${Nc_i}^1 = (nx_1, ny_1, nz_1)^\top$ and ${Nc_i}^2 = (nx_2, ny_2, nz_2)^\top$ can be compared directly with the ground truth $\bar{N} = (\bar{nx}, \bar{ny}, \bar{nz})^\top$. The error is computed as angle between the ground truth normal and the recovered normal. $\eta = 0$ if the recovered normal is perfect.

$$\eta_j = \cos^{-1}\left(\frac{\bar{N} \cdot {Nc_i}^j}{|\bar{N}|\left|{Nc_i}^j\right|}\right), j = 1, 2. \tag{4.2}$$

2. Centre Estimation Error (ϵ) : This error is computed as geometric distance between recovered position of conic centre ${Mc_i}^1(x_1, y_1, z_1), {Mc_i}^2(x_2, y_2, z_2)$ and ground truth $\bar{M}(\bar{x}, \bar{y}, \bar{z})$.

$$\epsilon_j = d(\bar{M}, {Mc_i}^j) = \sqrt{(\bar{M} - {Mc_i}^j)^2} = \sqrt{(\bar{x} - x_j)^2 + (\bar{y} - y_j)^2 + (\bar{z} - z_j)^2}, j = 1, 2. \tag{4.3}$$

3. Recovered Centre Separation (δ): This measure corresponds to the geometric distance between centres obtained from back projection $({Mc_i}^1, {Mc_i}^2)$ of the same conic.

$$\delta = d({Mc_i}^1, {Mc_i}^2) = \sqrt{({Mc_i}^1 - {Mc_i}^2)^2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}. \tag{4.4}$$

4. Recovered Normal Separation (ϕ) : This measure corresponds to the angle between the normals obtained from back projection $(Nc_i{}^1, Nc_i{}^2)$ of the same conic.

$$\phi = \cos^{-1}\left(\frac{Nc_i{}^1 \cdot Nc_i{}^2}{\left|Nc_i{}^1\right|\left|Nc_i{}^2\right|}\right). \tag{4.5}$$

The reader is reminded that the back projected data is recovered in camera coordinate system. Camera positions (Pose) must be available to bring ground truth Normal($\bar{N}$) and Centre position $\bar{M}$ to camera coordinate system for error measurement.

4.1.2. Data Preparation

A large set of synthetic images of a world circle are captured by varying object distance and viewing angle of the camera (independent parameters), while keeping image noise and circle size (dependent parameters) constant for a single experiment. The experiments are repeated for different configurations of dependent parameters. Input data required for the simulation is kept as close to real world application as possible. *Figure 4.3* gives a pictorial view of the data preparation process for the experiments. The recovery of normal and centre position of circle is based on the conic shape in image. Therefore, by rotating the camera around the object we can simulate different conic patterns on the image plane. A circle is a planar and symmetric object, hence rotation of the camera around the circle about one axis can generate sufficient number of patterns. The camera rotation is performed by varying the viewing angle ψ between camera axis and the normal of circle plane. *Figure 4.3* shows example of elliptical patterns created by varying ψ (camera position). The figure also shows r as distance between image plane and circle plane. The details of all the parameters involved in experiment is given below.

- 3D circle points : A circle is created by generic conic equation (eq. 3.1) for given marker size. The number of points used to generate a circle should be more than 5. This constraint is introduced by ellipse fitting algorithm. For this experiment we have used 100 sample points to create world circle.
- Marker Size $\varnothing$: This parameter uses realistic Circular marker sizes i.e. 5 mm and 12 mm.
- Camera Position : It is a predefined $[R \mid t]$ transformation used to position the camera. A 3D circle is projected on an image plane from different camera positions. Rotation component is generated by ψ,and it is varied from 0° to 70°. The translation (object distance) component is the distance from the camera r, and it is varied from 500 mm to 2000 mm. Higher viewing angles and object distances are not used due to limitations with ellipse detection.
- 2D conic points : 2D conic points corresponds to the projections of 3D circle on the image plane with known pose.
- Noise : In real conditions images captured are always with some noise which affects edge detection. To simulate this behaviour digitally generated noise is added to the

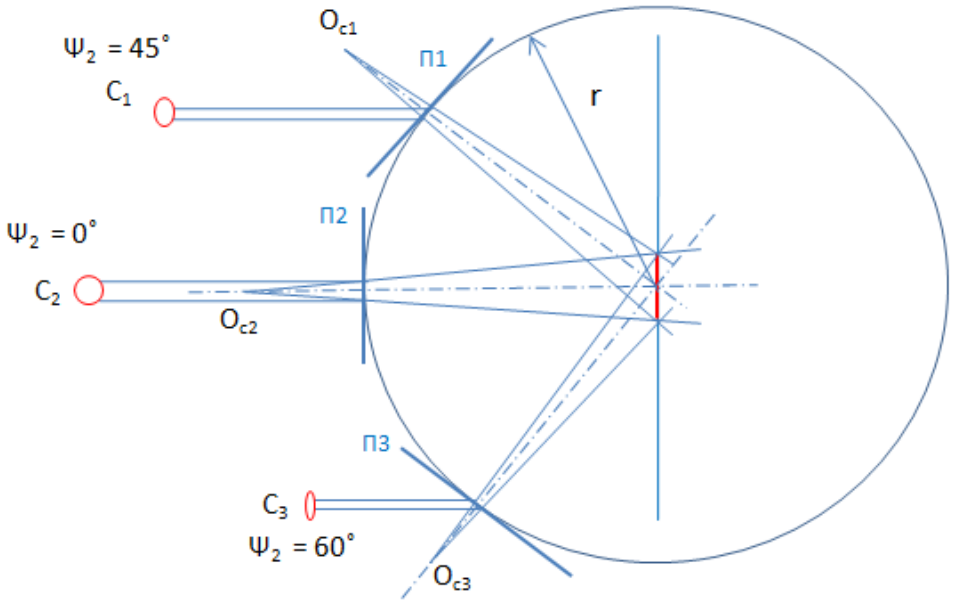


Figure 4.3.: Data preparation for experiment : $\pi 1, \pi 2, \pi 3$ represent image plane at different camera positions O_{c1}, O_{c2}, and O_{c3} respectively. C_1, C_2, C_3 represent the resultant image conic. r is the distance of the image plane from the imaged circle

.

2D conic points. Noise is in terms of pixel values, a Gaussian distribution model is assumed for simplicity. Noise is generated with mean (μ) = 0 and Standard deviation of σ_{noise} = 0.1, 0.3. Noise added to projected points before performing the ellipse fitting operation. Image noise is generally low for high resolution cameras (σ_{noise} = 0.1) and high for low quality cameras(σ_{noise} = 0.3).

- Iteration (n) : The experiments are repeated with same parameters for large number of iterations. The noise generated in each iteration is random and therefore higher number of iterations allows for reliable error modelling. Standard deviation is calculated at the end of the experiment for each error entity. The equation below shows measurement of Standard deviation in error for n iterations, where $\bar{p}_i$ is ground truth and $p_{i,j}$ is measured value in jth experiment.

$$\sigma = \sqrt{\frac{\sum_{j=0}^{j=n} \|p_{i,j} - \bar{p}_i\|^2}{n}}$$

Given below is the outline of the algorithm followed for the measurement of error in back projected data. For each experiment independent variables are varied over a defined range for a fixed set of dependent variables. The same experiments are repeated by varying the dependent variables to create error models for various configurations (different marker sizes, and image noise).

σ_{noise} = *0.1* , *n* = *100*, $\varnothing$ = *5* ;
for $n \leftarrow 1$ **to** *100* **do**
  for $r \leftarrow 500$ **to** *2000* **do**
    for $\psi \leftarrow 0$ **to** *70* **do**
      2dconicPoints ← `projectPoints(`3dcirclePoints`)` ;
      2dconicPoints ← `generateRandomNoise(`σ_{noise}`)` ;
      $C \leftarrow$ `ellipseFitting(`2dconicPoints`)` ;
      $Mc_i{}^1, Mc_i{}^2, Nc_i{}^1, Nc_i{}^2 \leftarrow$ `conicBackProjection(`C`)` ;
      $\eta_1 \leftarrow$ `computeAngle(`$Nc_i{}^1, \bar{N}$`)` ;
      $\eta_2 \leftarrow$ `computeAngle(`$Nc_i{}^2, \bar{N}$`)` ;
      $\epsilon_1 \leftarrow$ `computeDistance(`$Mc_i{}^1, \bar{M}$`)` ;
      $\epsilon_2 \leftarrow$ `computeDistance(`$Mc_i{}^2, \bar{M}$`)` ;
      $\delta \leftarrow$ `computeDistance(`$Mc_i{}^1, Mc_i{}^2$`)` ;
      errorData $[\psi, r, i] \leftarrow$ `SaveResult(`η, ϵ, δ`)` ;
    end
  end
end
`plotGraph(StandardDiviation(`errorData`),`ψ`,`*r*`)` ;

Algorithm 1: Creating error model for conic back projection method.

4.2. Data Analysis

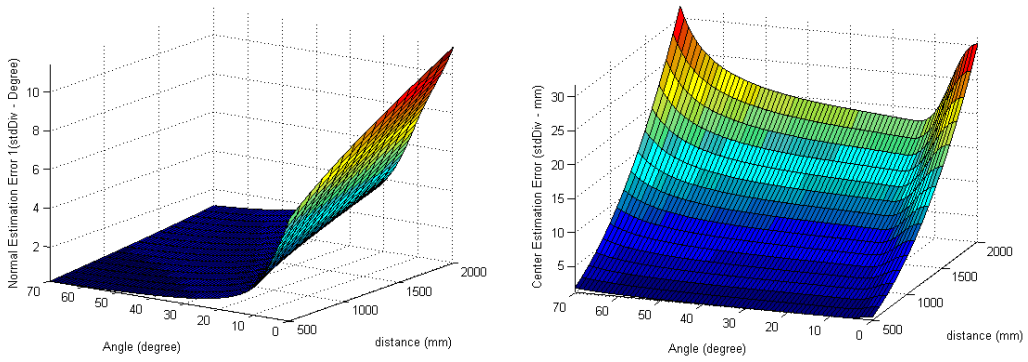
In this section the results of simulation are presented and various details are discussed. A three dimensional plot is created which depicts viewing angle (ψ) and object distance r along X and Y axis respectively. Z axis represents the standard deviation of the measured error entity. Reader should note that only important observations are commented here, detailed table of error values is provided in Appendices. One example for each error plot is given in Figure 4.4.

4.2.1. Normal Estimation Error

As we have ambiguity between the recovered normal solutions ($Nc_i{}^1, Nc_i{}^2$), only one of the two solutions which is closer to $\bar{N}$ is used for error estimation. *Figure 4.4a* shows simulation results for marker size $\varnothing$ =5 mm with image noise σ_{noise} = 0.1. The plot shows that if object distance is kept constant, for smaller viewing angles ($0 \leq \psi \leq 10$) normal recovery error (η) is higher, η drops significantly for $\psi \geq 15$, and gradually decreases from then onwards. For any given viewing angle, η increases with increment in the object

distance. It is also noted that growth of η, at lower viewing angles is higher for all object distances. Another observation suggests that, with higher noise (σ_{noise} =0.3) error pattern remains the same, however magnitude of errors are elevated. The simulation results for larger circles ($\varnothing$ =12 mm) show identical behaviour, except that the error values are smaller. The angle between the solutions itself $\angle(Nc_i{}^1, Nc_i{}^2)$ increases with increment in ψ irrespective of the object distance. This behaviour is also shown in terms of a graphical plot for interested readers.

Summary of the observation suggests that normal recovery error is high when image plane is parallel to circle plane (lower ψ). This means elliptical image conics produce normals closer to the real solution. If object distance is higher larger markers produce better results.



(a) Normal Estimation Error (η) plot for $\varnothing$ = 5 ,σ_{noise} = 0.1

(b) Centre Estimation Error (ϵ) plot for $\varnothing$ = 5 , σ_{noise} = 0.1

Figure 4.4.: Errors in Ellipse back projection

4.2.2. Centre Estimation Error

Figure 4.4b shows simulation results for marker size $\varnothing$ = 5mm and noise σ_{noise} = 0.1. The plot suggests that the Centre Estimation Error (ϵ) is high for smaller viewing angles ($0 \leq \psi \leq 10$). ϵ drops, and remains constant for a range of viewing angles ($10 \leq \psi \leq 45$), and further rises again for viewing angles $\psi \geq 45$. At any viewing angle, ϵ rises with the object distance r. Higher image noise ($\sigma_{noise} = 0.3$) produces same pattern with larger estimation errors.

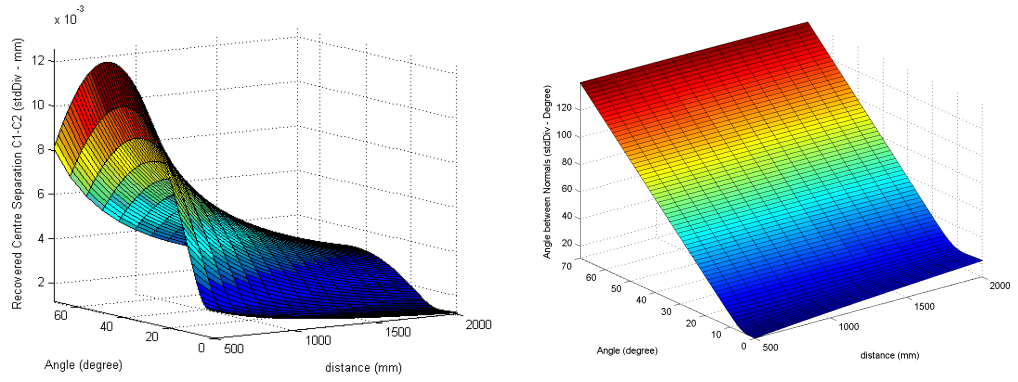
Here, a similarity can be drawn between the results of η and ϵ, both show higher errors at lower viewing angles. At lower viewing angles, the image conic appears circular in shape. This may suggest that image plane is parallel to the world circle plane. If object distance is higher, the range of viewing angles producing circular image projections becomes wider. This phenomenon may cause errors in computation of R', a rotation matrix used to align the image plane parallel to the target plane during ellipse back projection. It is also observed that centre position estimation becomes better with use of larger world circles.

4.2.3. Recovered Centre Separation

We know that the centre positions recovered from the image conic has a two fold ambiguity ($Mc_i{}^1, Mc_i{}^1$). *Figure 4.5a* shows difference between the solutions obtained from a single conic. We observe that centre separation is higher when object distance is lower, δ gradually decreases with increasing object distances. The maximum value of centre separation (δ) recorded in the plot is $\delta \approxeq 0.01$ mm. Experiments with $\varnothing = 12$ mm suggests that the centre separation (δ) values are higher for larger circles, however the maximum separation is $\leq 0.1mm$.

4.2.4. Recovered Normal Separation

The angle between the normals recovered from a single image conic explains the nature of normal ambiguity. *Figure 4.5b* shows that, at lower viewing angles normal separation $\phi \approxeq 10°$, and further increases gradually with increasing viewing angles.



(a) Recovered Center Separation $\varnothing$ = 5 , noise = 0.1(b) Recovered Normal Separation $\varnothing$ = 5 , noise = 0.1

Figure 4.5.: Separation between recovered solutions of a single conic

4.2.5. Notes for generating descriptors

A pair of circles produce multiple three different Euclidean invariants (section 3.3.1). These invariants are created by obtaining circle plane orientation (in 3D) from its conic image projections. Our objective is to find a suitable set of invariants, which can be used for creating strong feature descriptors. After learning behaviour of the conic back projection method, it is important to apply these learnings for designing a feature descriptor.

Invariant 1 : Angle between surface planes θ Simulation shows that normal recovery is less erroneous for larger viewing angles (ψ). If circle size is chosen by keeping operating distance in mind, the invariants generated will be less erroneous. The invariant θ within acceptable threshold range can be used for feature matching. However, for coplanar circles with plane parallel to image plane ($\psi = 0$), the result for θ may not be very accurate.

This comment is also supported by the result of the experiments performed by Forsyth [FMZ+91] on real objects. The reader should also note that invariant retrieved will always carry ambiguity. The ambiguous solutions for θ can be demonstrated as follows,

$$\begin{aligned} C_1 &\rightarrow {Nc_1}^1, {Nc_1}^2 \\ C_2 &\rightarrow {Nc_2}^1, {Nc_2}^2 \\ \theta_{11} &= \angle({Nc_1}^1, {Nc_2}^1) \\ \theta_{12} &= \angle({Nc_1}^1, {Nc_2}^2) \\ \theta_{21} &= \angle({Nc_1}^2, {Nc_2}^1) \\ \theta_{22} &= \angle({Nc_1}^2, {Nc_2}^2) \end{aligned} \tag{4.6}$$

Only one solution is true among the 4 solutions obtained from a pair of conics. As there is no method for discarding the ambiguity, it will exist in the 2D feature descriptor.

Invariant 2 : Vector joining circle centres This category is further divided into three invariants, two of these (γ_{n1} and γ_{n2}) are generated with combined information of surface normals and centre positions. The distance between the centres, d_c is computed in object scale. From simulations we learn that recovered centre position has higher error if image conic is too elliptical or too circular. γ_{n1} and γ_{n2} is expected to have higher errors as it involves use of all back projected information. Therefore, use of d_c is preferred over γ_{n1} and γ_{n2}. The additional advantage of using d_c is that all four solutions are consistent.

$$\begin{aligned} C_1 &\rightarrow {Mc_1}^1, {Mc_1}^2 \\ C_2 &\rightarrow {Mc_2}^1, {Mc_2}^2 \\ d_{c11} &= d({Mc_1}^1, {Mc_2}^1) \\ d_{c12} &= d({Mc_1}^1, {Mc_2}^2) \\ d_{c21} &= d({Mc_1}^2, {Mc_2}^1) \\ d_{c22} &= d({Mc_1}^2, {Mc_2}^2) \end{aligned} \tag{4.7}$$

But, we also know that the value of (δ) is small and can be neglected.

$$\begin{aligned} \delta_1 &= d({Mc_1}^1, {Mc_1}^2) \ll 0.1mm \\ \delta_2 &= d({Mc_2}^1, {Mc_2}^2) \ll 0.1mm \\ \therefore d_{c11} &\approx d_{c12} \approx d_{c21} \approx d_{c22} \end{aligned} \tag{4.8}$$

Therefore all the four solutions will result in consistent values, this invariant has higher errors but is immune to ambiguity.

The observations above suggest that a 2D feature descriptor can be designed including both d_c, θ . Combination of d_c and θ can provide uniqueness desired for matching the feature descriptors.

CHAPTER
FIVE

TRACKING IMPLEMENTATION

In part I of the thesis we discussed the need for developing Circular marker based Industrial Augmented Reality solution. The structure of new tracking system was defined and theoretical concepts for the components of the tracking system were discussed. In this chapter the practical implementation of the Tracking method is demonstrated. The basic work flow presented in 3.1 is followed for method implementation. This chapter gives an elaborate description of methods chosen for Marker Detection and Identification.

The implementation of the method is done in C++, OpenCV[Ope] and Ubitrack[UBI] are the libraries are used to perform known image processing (i.e. Image Binarization , Ellipse Fitting) and computer vision operations (i.e. Pose estimation).

5.1. Marker Detection

The basic requirement from this process is maximum detection of markers from the image and avoid false positives. The detection should be robust to variation in light conditions, and it should detect the *Image Points* m_i (marker centres) accurately. Marker detection stage uses various image processing methods, which requires specific parameter settings for optimal performance. For a tracking system it is desired that the parameters settings are generic, this allows for reliable performance in different scenes. The parameters are tuned to have best performance, however as a fail-safe measure we have designed the system such that vital parameters can be changed while application is running. We call them *Control Parameters*.

Adaptive thresholding is used for image binarization stage to compensate for illumination changes in image. The contour extraction is done from binary images with hierarchical information. Further, contours are filtered and non circular contours are rejected. The initial rejection is done using some of the Control parameters. For refined rejection information of contour hierarchy is used. The filtered set of contours are used for Ellipse fitting stage. The concept of ellipse fitting is covered in chapter 3 (section 3.2). The fitting method provides elliptical solution for any given set of points, therefore residual error is used for further rejection of bad contours. The remaining contours are selected as markers.

The conic matrix C_i and ellipse parameters E are given as output from this process. E also contains co-ordinate position of detected marker centre m_i. It is to be noted that higher fitting accuracy is achieved by using refined contour edge points extracted from original image (using binary image contours). The Control parameters are given as follows,

- Contour Size : Rejects contours based of minimum or maximum size.
- Contour Height-Width Ratio : Rejects contours lower than set threshold value.
- Contour region pixel intensity : Rejects contours based on pixel intensity of the centre.
- Adaptive Threshold window size : Parameter to set window size for thresholding operation.
- Fitting residual threshold : Rejects contours resulting in residual errors larger than set threshold value.

5.2. Marker Identification

In Chapter 3 we concluded that 3D-2D feature descriptor matching is a suitable method for Marker Identification process. It was also suggested that invariants generated from 2D image conics can be used to create correspondence ($m_i \leftrightarrow M_i$) hypothesis. A pair of image circles can generate various Euclidean invariants, some of which (d_c, θ) were suggested to generate unique 2D-descriptors after synthetic analysis in Chapter 4. The basic work flow was suggested in Chapter 3 (section 3.3). In the following section, implementation of identification process will be explained.

5.2.1. Feature Descriptor Generation

3D Data (position, normals and size) of the markers is available before starting the tracking application, whereas *2D Data* is obtained after the applications starts. Therefore, Feature Descriptor Generation process is divided into two modes namely Online and Offline.

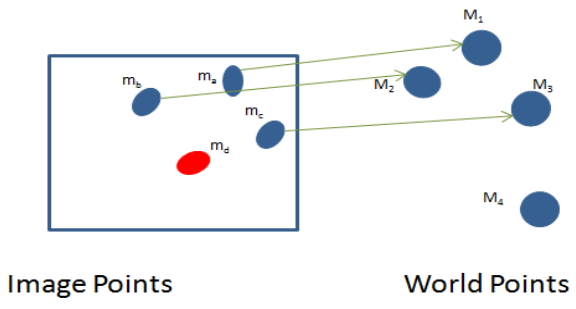


Figure 5.1.: Example for Marker Identification process

To explain the whole process in simplified manner we will follow each step of Marker Identification process with an *Application Example* given in *Figure 5.1*. Assuming we have only 4 world points and 4 image points. Matching information is given in the image. M_4 world point is not captured in the image and m_d is a false positive detected in the image. Real correspondence ,

$$M_1 \leftrightarrow m_a, M_2 \leftrightarrow m_b, M_3 \leftrightarrow m_c$$

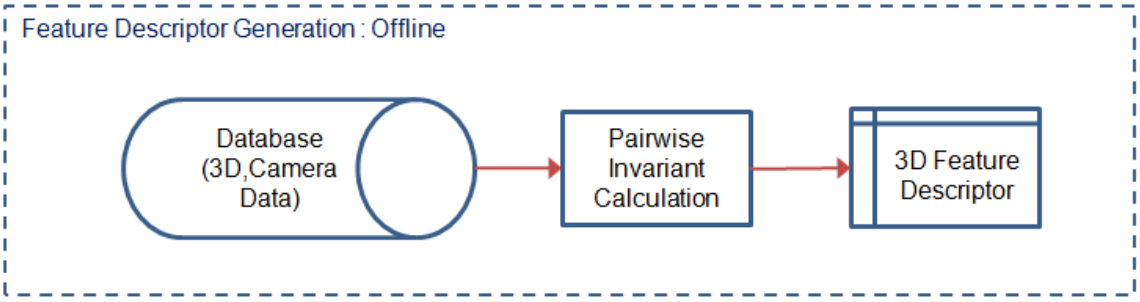


Figure 5.2.: 3D Feature Descriptor Generation in Offline mode

Offline Mode

As *Figure 5.2* suggests 3D Feature Descriptors are generated in Offline mode. This means that the *3D Data* is processed and invariants are generated before the tracking application starts. 3D Feature Descriptors are generated once and stored in the database. These descriptors are reusable unless the marker orientation has been changed. The structure of feature descriptor contains invariants (d_c, θ) generated from a pair of markers. A feature descriptor table is generated for all possible pair of markers. Structure of a 3D feature descriptor is explained below,
For ith marker 3D Data is given as,

$$M_i(X_i, Y_i, Z_i), N_i(Nx_i, Ny_i, Nz_i)$$

For marker pair $\langle i, j \rangle$ Invariants are,

$$d_c = d(M_i, M_j), \theta = \angle(Ni, Nj)$$

For marker pair $\langle i, j \rangle$ Feature Descriptor is,

$$V_p = V_{ij} = \langle d_c, \theta \rangle \tag{5.1}$$

Since each *Model Point* is considered for pairwise invariant generation, a total of $\binom{n}{2}$ feature descriptors are generated for n points (i.e. for n = 4 , no. of descriptors = 6). The equation 5.1 shows that each marker pair $\langle i, j \rangle$ is uniquely mapped to index variable p. The set of feature descriptors generated Offline for the *Application Example* (See *Figure* 5.1) are given below,

$$\text{3D Feature Descriptors} : V_{12}, V_{13}, V_{14}, V_{23}, V_{24}, V_{34}$$

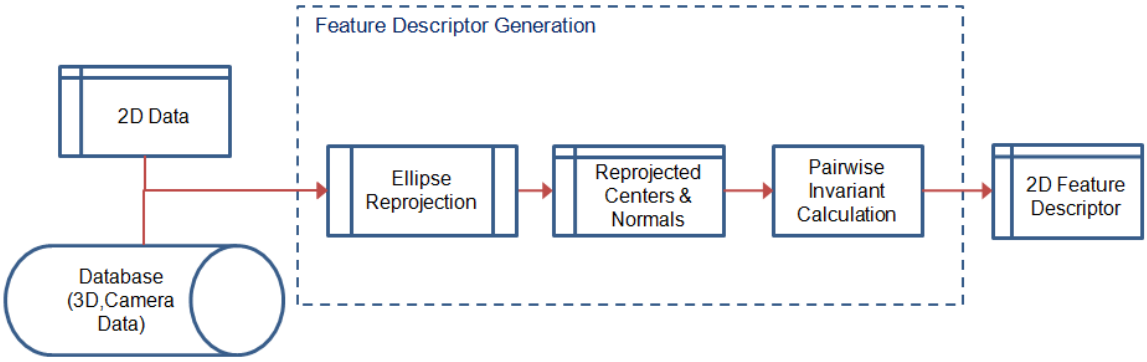


Figure 5.3.: 2D Feature Descriptor Generation in Online mode

Online Mode

2D feature Descriptors are generated in Online mode. This means that the feature descriptors are generated during run-time. *Figure 5.3* shows the steps followed to generate 2D descriptors. Once image is acquired, *2D Data* (C_i, m_i) is calculated, conic back projection is performed on *2D Data* to compute required Invariants (d_c, θ). As discussed in Chapter 4, the invariants generated for a pair of conic contains ambiguity. Invariant d_c and θ has 4 solutions (See 4.2.5) for each. All solutions of d_c are consistent and therefore they are averaged to create a single value for descriptor. As for θ only one solution is correct however without correspondence information none can be discarded. The structure of descriptor is derived from equations 4.6 and 4.8,
For marker pair $\langle i, j \rangle$ Invariants are,

$$d_{avg} = Avg(d_{c11}, d_{c12}, d_{c21}, d_{c22})$$

$$\theta = \theta_{11}, \theta_{12}, \theta_{13}, \theta_{14}$$

For marker pair $\langle i, j \rangle$ 2D Feature Descriptor,

$$v_q = v_{ij} = \langle d_{avg}, \theta_{11}, \theta_{12}, \theta_{13}, \theta_{14} \rangle \tag{5.2}$$

Pairwise 2D descriptors are calculated for all possible marker pairs using their back projected information (i.e. ${Nc_i}^1, {Nc_i}^2, {Mc_i}^1, {Mc_i}^2$). Each 2D marker pair $\langle i, j \rangle$ is uniquely mapped to index variable q as displayed in equation 5.2. The set of feature descriptors created for the *Application Example* (See *Figure* 5.1) are,

$$\text{2D Feature Descriptors} : v_{ab}, v_{ac}, v_{ad}, v_{bc}, v_{bd}, v_{cd}$$

5.2.2. Matching Strategy

This is the last stage of Marker Identification process. In this stage 2D Feature Descriptors and 3D Feature Descriptors are compared and correspondence hypothesis is formed. The matching process becomes complicated as 3D Descriptors are large in number and camera position is not initialised. Lack of initialisation forces us to consider all 3D points as possible candidates for matching. *Figure 5.4* shows the outline of the strategy developed for creating *Correspondence Data*.

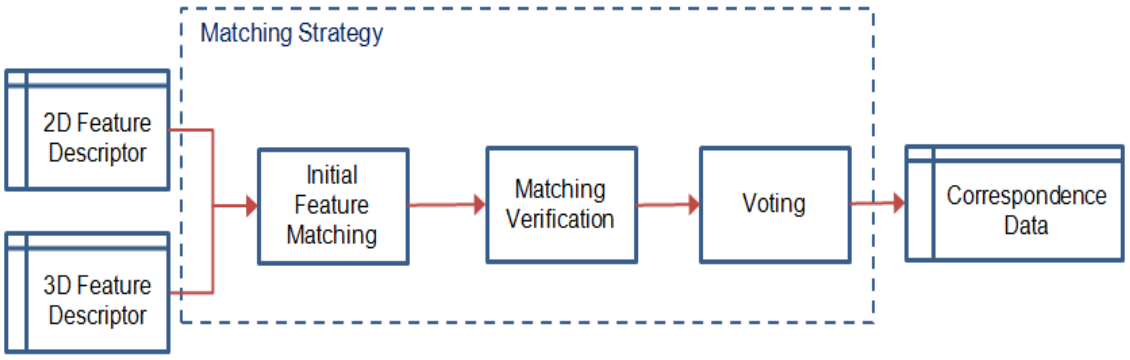


Figure 5.4.: Matching Strategy for Marker Identification process

Initial Feature Matching

The descriptor generated from *2D Data* and *3D Data* are supplied as input for feature matching stage. We know that 3D Descriptors contain information of all markers attached to the object, However 2D Descriptors are limited to markers present in the image. Initial descriptor matching results in a pair-wise matching. This is because of the fact that each descriptor represents a pair of markers. Let us assume that we have n 3D descriptors (V) and l 2D descriptors (v). Generally $l < n$ if the image does not contain large number of false positives, irrespective of this the algorithm should be able to handle such conditions.

Algorithm 2 describes the structure of descriptor matching algorithm. Each descriptor represents a marker pair, this unique combination is mapped to index variables (i.e. p, q) of each descriptor. Equation 5.1 and 5.2 describe the structure of both descriptors. Ideally for corresponding pair of markers descriptors contains same value for d_c and θ. To compare the descriptors initially d_c component is compared with matching criteria (T_{d_c}). If d_c component is satisfied, the θ component is compared; otherwise a new descriptor pair is selected. 2D Descriptor includes ambiguity for θ component, hence all solutions (i.e. $\theta_{11}, \theta_{12}, \theta_{13}, \theta_{14}$) are checked with T_θ as matching criteria. The index of descriptors are saved if the matching is successful. Matching criteria are decided based on observations from synthetic analysis in Chapter 4. T_{d_c} is less stringent as d_c component of 2D descriptors it is prone to higher errors, however angle recovery is generally more accurate and therefore T_δ can be stringent.

3D constellation of points is mostly random and therefore there are high chances of two different marker pair leading to similar descriptor values. This fact is taken into account and therefore one-to-many matching relation is allowed for initial stage of descriptor matching. Let us assume that after this stage *Application Example* (See *Figure* 5.1) produces

Goal : Find all possible V_p similar to v_q ;
Initialisation : T_{d_c} = 10 , T_θ = 5 ;
forall the *3D Feature Descriptors (V), $p \leftarrow 0$* **to** n **do**
  forall the *2D Feature Descriptors (v), $q \leftarrow 0$* **to** l **do**
    if `compared`$_c$`(`V_p`,`v_q`)` $< T_{d_c}$ **then** `// compares` d_c `component`
      if `compare`θ`(`V_p`,`v_q`)` $< T_\theta$ **then** `// compares` θ `component`
        `// All 4 solutions of` θ `in` v_q `are checked`
        ;
        `SavePairResult(`p, q`) // Save matching descriptor pair`
        ;
      end
    end
  end
end

Algorithm 2: Initial pair matching algorithm

results as given below,

$$1\rangle V_{12} \leftrightarrow v_{ab}$$
$$2\rangle V_{12} \leftrightarrow v_{cd}$$
$$3\rangle V_{13} \leftrightarrow v_{ac}$$
$$4\rangle V_{23} \leftrightarrow v_{bc}$$
$$5\rangle V_{24} \leftrightarrow v_{ac}$$
$$6\rangle V_{24} \leftrightarrow v_{bd}$$

The initial matching results of *Application Example* show that descriptor pair V_{12}, V_{24} have multiple matching candidates. This stage filters out irrelevant data and complexity of the problem is reduced significantly.

Matching Verification

In this stage of the process results of Initial Feature Matching stage are refined. We know that one-to-many matching relation exists between both sets of descriptors. This matching result should be further reduced to one-to-one type of relation. It is also to be noted that even if a correct descriptor match is found (i.e. $V_{M_iM_j} \leftrightarrow v_{m_km_l}$) ambiguity still exists in this solution. This is ambiguity is regarding point-to-point matching, as $M_i \leftrightarrow m_k$ and $M_j \leftrightarrow m_l$ or $M_i \leftrightarrow m_l$ and $M_j \leftrightarrow m_k$, both can be valid results for given descriptor match. To solve this ambiguity we use three such descriptor matching pairs ($V_{M_iM_j} \leftrightarrow v_{m_km_l}$) to generate 3 point-to-point ($m_i \leftrightarrow M_i$) correspondence hypothesis. We call this method as *Triplet Matching* and matching hypothesis is termed as *Correspondence Triplet*. *Triplet Matching* method also contributes towards discarding false descriptor matching results. The descriptor pairs for *Triplet Matching* are selected by following two simple steps,

Step 1 Find any two descriptor matching pair in which one and only one common *Model Point* (for 3D descriptor) and *Image Point*(for 2D Descriptor) exists. As only one point

is common we can form a *Correspondence Triplet*. In case of *Application Example* pair 1 ($V_{12} \leftrightarrow v_{ab}$) and 3 ($V_{13} \leftrightarrow v_{ac}$) form a *Correspondence Triplet*.

$$[1 \quad 2 \quad 3] \leftrightarrow [a \quad b \quad c]$$

Step 2 Find a new pair (not selected in step 1), that supports hypothesis of step 1. This pair is called *Triplet Verification pair*. In *Application Example* pair 4 ($V_{23} \leftrightarrow v_{bc}$) is a candidate for *Triplet Verification Pair*.

As demonstrated above, *Triplet Matching* produces 3 point-to-point correspondence hypothesis. Also the false descriptor matchings results are rejected. In case of *Application Example* pair 4 and 5 pass step 1, but lack a supporting *Triplet Verification Pair* to pass step 2. Similarly pair 2 and 3 also pass step 1 , but pair 4 produces conflict hypothesis, and verification fails for the pair.

Voting

In a system with multiple markers after Matching Verification stage we have different sets of *Correspondence Triplets*. It is possible that we have multiple triplets with conflicting correspondence hypothesis. On the other hand it is also possible that various triplets support the same $m_i \leftrightarrow M_i$ correspondence hypothesis. Voting stage is designed to create a final *Correspondence Data* to be used for pose estimation stage. In this stage voting is done for each $m_i \leftrightarrow M_i$ pair based on *Correspondence Triplet* hypothesis obtained in earlier stage. A matrix is formed and number of votes gained by each pair are entered. A pair having maximum number of votes is considered and other are rejected. Further, All pairs having vote above a certain threshold (T_{voting}) are selected. For our application we select T_{voting} as half of the maximum vote a pair has achieved in the Voting matrix. Also pairs with conflicting votes (same number of votes or below T_{voting}) are rejected in this stage. In *Application Example* each pair gains 1 vote and no other pair gains any vote.

$$1 \leftrightarrow a, \quad 2 \leftrightarrow b, \quad 3 \leftrightarrow c$$

Application Example is a simplified model to explain stages of Marker Identification process, however for Voting we will use an additional example to visualise the process clearly. In this example, A-H represent *Model Points* (M_i) and 1-5 represent detected *Image Points* (m_i). *Figure 5.5* shows the a graphical view of the resultant Voting matrix.

$$\begin{aligned} A &\leftrightarrow 1 \qquad Vote : 4 \\ D &\leftrightarrow 2 \qquad Vote : 4 \\ F &\leftrightarrow 3 \qquad Vote : 4 \\ H &\leftrightarrow 4 \qquad Vote : 4 \\ B &\leftrightarrow 5 \qquad Vote : 3 \\ E &\leftrightarrow 5 \qquad Vote : 3 \end{aligned}$$

Reader should note that the results given above show only the pairs which gained maximum vote in the matrix. The pairs gaining least vote are rejected. The maximum vote in matrix is 4 and therefore T_{voting} is 2 (0.5×4). An observation of the above given results suggests that first 4 pairs are unique and have high voting count. These pairs are selected as *Correspondence Data* for pose estimation. On the other hand B and E show an ambiguity over point 5 as both show same voting count, hence the pairs are not unique and both are rejected.

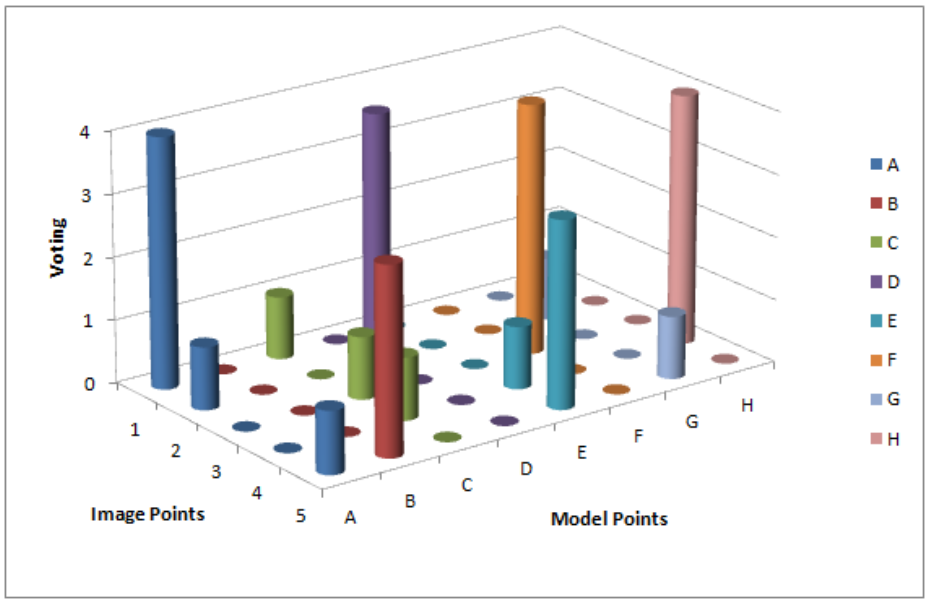


Figure 5.5.: Voting results for correspondence matching

5.2.3. Pose Estimation

As discussed in Chapter 3 reliable methods for Pose Estimation are readily available. Therefore we will use the methods available with OpenCV[Ope] library to achieve the final Pose. We have used a multi stage approach for Pose Estimation process to obtain pose result with higher accuracy.

Initial Pose Estimation

The *Correspondence Data* available from Marker Identification stage is used for Initial Pose Estimation. As discuss earlier for a unique pose estimation correspondence information ($m_i \leftrightarrow M_i$) of six points are enough. For current implementation, The *Correspondence Data* must contain at least six points pairs for estimating the object pose. We have used quaternion representation for the resultant Pose.

Refine Correspondence Data

If initial pose of the object is calculated, this step is used to refine the *Correspondence Data*. This step is performed to generate a larger set of correspondence pairs. It is very likely

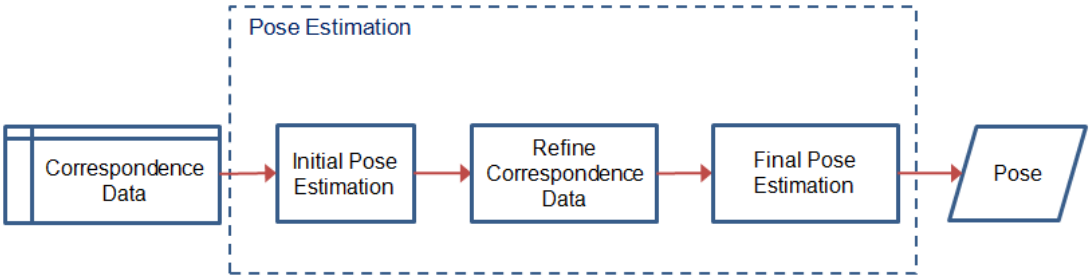


Figure 5.6.: Pose Estimation work flow

that for a given image, out of n *Image Points*, correspondence is found only for k points and $s(n-k)$ points are not matched with any *Model Point*. This may happen in case of failure in descriptor matching or weak invariant information. In this case pose calculated in earlier stage is used to find new correspondence information for s points.

The projection matrix P is calculated for that particular view using pose information. The projection matrix is used to project all the *Model Points* (M_i) of the object to the image plane (See equation 3.6). We call these set of points as *Projected Points*. If any unmatched *Projected Point* is found close to an existing but unmatched *Image Point* ($\in s$), the corresponding *Model Point* and *Image Point* pair is added to the existing *Correspondence Data*.

$$d(ProjectedPoint, ImagePoint) < T_{matching}$$

$T_{matching}$ is the acceptance criteria for the new matching, it is measured in terms of pixels. In our application we have chosen this value as 10.

Final Pose Estimation

In this stage, pose of the object is calculated again with the new correspondence results obtained in previous stage. This result is more reliable as the system is over determined as we have more equations than constraints. The output of this stage is used as final result of the tracking work flow.

CHAPTER
SIX

EVALUATION

In this chapter evaluation results of the proposed tracking approach are presented. Two 3D models are prepared with circular markers of different sizes. This chapter includes performance analysis of the tracking method based on experiments performed on a real 3D object. Additionally, simulations are used to explain the behaviour of the tracking system.

As discussed in the problem statement, the main focus of the thesis is to solve two problems, Marker Detection and Marker Identification. Marker Detection stage is implemented to extract marker information from the image using known methods. Sub-pixel edge detection, ellipse fitting and centroid measurement for conics are well studied topics in the literature. Therefore we would refrain from performing an exhaustive analysis on accuracy of marker detection process. Similarly, the process Pose Estimation is designed using existing methods. In this thesis we have introduced a new feature descriptor to perform 2D-3D correspondence matching from a single image consisting of multiple circles. Therefore, The experiments performed in this chapter largely focus on the Marker Identification (matching) process.

Data Preparation

Figure 6.1 shows the 3D car models prepared for the experiments. The car models are identical in shape, but the markers attached are of different sizes. The marker configuration of the two car models is given in the following table.

Table 6.1.: Marker configuration of the car models

Model 1	No of Points	20
	Marker Size	∅ 12mm
Model 2	No of Points	26
	Marker Size	∅ 5mm
	No of Points	21
	Marker Size	∅ 8mm

It should be noted that Model 2 has Circular markers with two different sizes. The experiments are focussed on 5mm and 12mm markers, 8mm markers mainly function as false positives on the model for some experiments. The markers are attached randomly to facilitate for unique feature descriptor generation. The *3D Data* for car models is generated from a highly accurate Photogrammetric system used in Industrial applications [GOM]. A 5 Mpx camera with image resolution of 2560 x 1920 is used for all evaluation experiments. The computing system used for all the experiments uses Intel Core i5 processor (M 460, 2.53 GHz) with 8.0 GB RAM.

Figure 6.1.: 3D car models used for Tracking

The experiments are targeted to analyse specific processes of the tracking method. A novel method for 2D-3D correspondence matching (using image conics) is introduced in this thesis, hence it is vital that experiments provide understanding of its reliability and limitations. First set of experiments will define a working area for the developed tracking system. Once the working area is defined, next set of experiments will focus on understanding the factors affecting matching strategy. Our generic goal is to provide a practical tracking system for the Industrial community, therefore experiments are also designed to answer usability related questions. Finally, we will demonstrate the performance of the tracking method with respect to time.

6.1. Experiment 1 : Working Area

Detection Range

Aim : To identify the limitations of marker detection with respect to object distance.

Set-up : A planar object with 4 markers of $\varnothing$ = 5mm and 12mm is used to define minimum and maximum detection range for a given marker size. The plane is rotated to simulate the change in viewing angle (ψ).

Parameters :

- Contour Size (20-30000 pixel)
- Contour Height to Width Ratio (1.8)
- Contour region pixel intensity (30)
- Adaptive Threshold window size (21)
- Fitting residual threshold (1.0)

The threshold values are most generic for various conditions and can be changed to optimize the performance of detection. The reader is referred to the Section 3.2 for understanding the function Control parameters.

Results : The distance measurements are displayed with mm as measurement unit. The results of the experiment are detailed in *Table 6.2*.

Table 6.2.: Detection range specification of the Circular markers

$\varnothing$ (mm)	ψ (°)	min Distance	max Distance
5	0	800	2900
	40	400	2500
	60	600	2700
12	0	1000	4000
	40	600	3600
	60	800	3000

The results show that detection range is higher for markers with larger circle size. The results also suggests that at higher viewing angle, minimum detection distance decreases. This is due to the fact that any contour bigger than the threshold size is rejected. In case of higher viewing angles projected marker size is smaller than at $\psi = 0°$. At larger object distances detection is limited due as marker size. Also, smaller markers (5mm) support large viewing angle at close distances, whereas bigger markers (12mm) support larger viewing angle at higher distances.

Tracking Range

Detection of the marker is absolutely necessary for tracking the target object. When multiple markers are attached to a 3D object, Each marker produces different projections on

Table 6.3.: Tracking range of car models

Size (∅)	min Distance	max Distance	Remark
5	600	1200	Optimum
	1200	1500	Orientation Specific
12	600	2000	Optimum
	2000	3600	Orientation Specific

the image plane. For a rigid 3D object with markers on different planes tracking range is smaller than detection range. The viewing angle of the camera affects the quality of the conic parameters, which in turn affects quality of 2D descriptors. We need at least six correspondence for computing the pose of the object. Therefore, this experiment is performed to define the tracking range for the car models.

Aim : To identify the limitations of model tracking with respect to object distance.

Set-up : car models are positioned in front of the camera. The distance and orientation of the car model are varied and operating range is determined.

Parameters : We have fixed values for matching parameters to $T_\theta = 5, T_{d_c} = 10, T_{voting} = 0.5 * MaxVote$. We have the information regarding origin of object coordinate system. Tracking is considered successful if this origin is recovered repeatedly for at least 100 image frames.

Results :

We observed that if sufficient points (≥ 6) are detected and matched, the pose is computed without any interruption. It is observed that the tracking results deteriorate as the distances from camera are increased. We have provided two sets of values to define tracking range for the system. Optimum performance range is the distance range in which the model is tracked without interruption. An additional tracking range is also provided with the results. This range is specific to object orientation, because at larger object distances tracking results are unstable. In such cases it is observed that if orientation of object is changed in order to detect additional number of markers, tracking results are improved. The results also suggest that for bigger markers working range is higher.

6.2. Experiment 2 : Marker Orientation

In this section we will analyse various parameters affecting marker identification stage of the tracking algorithm. The thresholds used for Marker Identification $(T_\theta, T_{d_c}, T_{voting})$ are selected for best performance with given set of car models. However, it is important to understand how to select threshold values such that correspondence matching results can be improved. In chapter 4 we already learned that recovery of circle plane in 3D using image conic back projection has its limitations. Therefore, 2D descriptors used for correspondence matching depend on marker constellation on the given 3D object (Density and Orientation of markers). From usability perspective it is important to understand

limitations of the algorithm regarding marker placement. The selection of marker sizes also affects the performance of the correspondence matching methods.

6.2.1. Marker Distribution and Threshold Selection

Aim : To identify limitations regarding marker selection and positioning.

Set-up : A synthetic 3D model consisting of a pair of circles (⌀) is generated. The distance and orientation of circle planes are varied in a stepwise manner. Images are captured from 1000 random camera positions for each plane configuration. Based on the image projections 2D descriptors are created and matched with ground truth data.

Parameters : The orientation of circle planes, the rotation and translation range for camera positions are varied in controlled manner for each experiment. Descriptor matching success is recorded for dependent parameters like Circle Radius, Distance from camera. The measurement unit is mm in case of distances and degrees (°) for angles.

- Distance between centres (λ) : 10 - 150 (Step size : 10).
- Angle between circles planes (β) : 0 - 90 (Step size : 10).
- Image noise factor : 0.3 (σ).
- No. of images captured at each position : 1000.
- Camera rotation angle Range (x-y-z) : -65 to 65.
- Translation Range X - Y : -100 to 100.

For the experiment the angle between circle plane is changed, hence camera positions can not be fixed for entire experiment. To accommodate this limitation, a new set of camera positions are generated for each circle plane configuration. Thresholds used to validate descriptor matching success are kept constant at $T_{d_c} = 10$ and $T_\theta = 5$.

Results :

In this experiment, success rate of descriptor matching is recorded for a range of camera distances with different marker sizes. A percentage scale is used to visualise the matching success. *Table 6.4* shows the summary of the results. The quality of matching success drops when camera distance is increased. On the similar lines, at higher camera distances bigger markers produce better matching success. The results show that angle between circle planes has higher influence on the descriptor matching success. The result of descriptor matching is an AND operation between matching results of individual descriptor d_c and θ.

Matching strategy compares the descriptor components (θ, d_c) of 2D and 3D feature descriptors. The results of the experiment shows that, the matching success decreases when the angles between the circle planes are increased. In Chapter 4 we observed that Normal Estimation Error (i.e. η) is higher at large camera distances. While computing θ component, errors of individual normal estimation are added ($\eta_1 + \eta_2$). As a result, matching criteria ($\theta < T_\theta$) is not satisfied, which leads to failure in descriptor matching. T_θ can

Table 6.4.: Matching Success analysis

∅	β	Camera Distance	Matching Success
12	0-40	500-2000	80-95
	40-90	500-2000	50-80
	0-40	2000-4000	40-60
	40-90	2000-4000	20-40
5	0-40	500-1200	75-90
	40-90	500-1200	40-75
	0-40	1200-2000	55-80
	40-90	1200-2000	35-55
20	0-80	500-2000	100
20	80-90	500-2000	30-90

be tuned to allow for higher matching possibilities at higher camera distances. The recovery of 3D position of the circle centre is less robust (i.e. Centre Estimation Error ϵ) at higher distances. To compensate for this fact d_c component has less stringent threshold ($T_{d_c} = 10mm$) to allow for higher initial descriptor matching, the results are later filtered by θ component. Relaxing the threshold for both d_c and θ matching may enhance the matching success for a pair of conic, but may affect overall accuracy of matching process. This may happen when marker density on object is higher, because false matching results will be generated with flexible thresholds. Therefore, marker density on the object should also be considered while changing the thresholds.

6.2.2. Relation of Voting Pattern with False Correspondence Matching

There are various cases in which false correspondence matching is observed. As explained earlier the cause can be poor choice of matching thresholds, dense marker arrangement or object distance from camera. However, a single incorrect matching can result in wrong pose estimation, therefore such cases should be examined to improve matching performance. *Figure 6.2* shows examples of correct pose estimation and wrong pose estimation. The green points in the image are the points used for pose estimation. *Figure 6.2b* shows that incorrect correspondences are formed, and wrong pose is computed. The reader should note that filtering descriptor matching results with thresholds has its limitations, false correspondences may still appear in final correspondence data. Multiple objects with circular marker patterns may also influence correspondence selection.

Aim : To identify role of voting threshold in correspondence results.

Set-up : The experiment set up consists of both car models with circular markers to encourage false correspondence matching. 90 images are captured with different camera positions, maximum voting count and number of matched points are recorded for each image. At least six point correspondences are required to compute the pose for the object. The images producing less than six correspondence matches are not considered for this experiment.

(a) Correct Matching (b) Incorrect Matching

Figure 6.2.: Examples of matching results

Parameters : T_{voting} = 0.5, a correspondence pair is selected if,

$$\text{Vote count} > (\text{MaxVoteCount}) * T_{voting}.$$

Outcome :

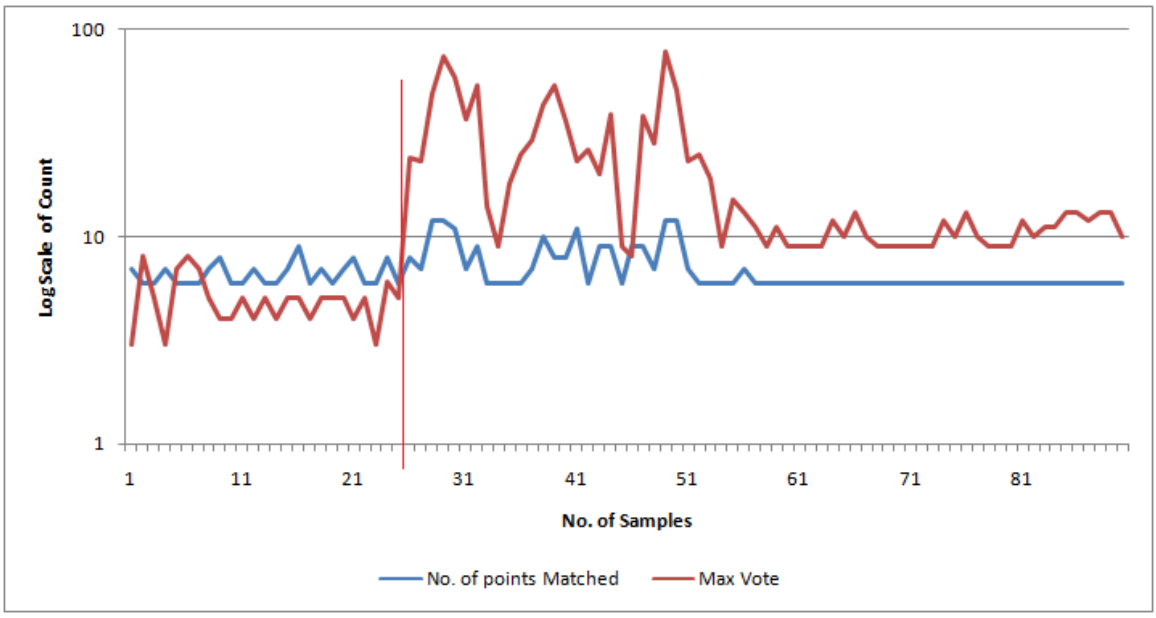


Figure 6.3.: Voting pattern for a set of images

Figure 6.3 shows the voting pattern (log scale, Y-axis) for a set of images. Pose estimation results of each image are verified manually by drawing object origin in the image. The images are separated into two groups for better visualisation of voting pattern. First group

(Image 1 - 25) contains Images showing false correspondence matching (Example Fig. 6.2b) and Second group (26-90) is of Images showing correct correspondence matching results (Example Fig. 6.2a). The graph shows separation between the two groups with a vertical red line. The Voting pattern suggests that maximum voting count is lower than the number of matching points for images showing false correspondence results. On the contrary maximum voting count increases exponentially with number of matching points in case of correct matching results. Therefore, an additional filtering criteria can be added to obtain reliable correspondence results. Correspondence hypothesis should be rejected if maximum vote count is lower than total number of matched pairs. This result also concludes that at higher operating distances it is better to have more points in scene for maximising possibility of correct results. It should be noted that each valid point pair contributes towards creating a higher voting count. If more markers are captured in the image, better rejection of false matching can be achieved.

6.3. Experiment 3 : Tracking Speed Analysis

In this part of the experiment we will present details regarding time consumed by each phase of tracking algorithm. This experiment is to check if the tracking method can fulfil performance expectations of Industry.

Aim : To identify overall performance of tracking method with respect to time, especially for Marker Identification stage.

Set-up : The experimental set up consists of two cameras, a high resolution (2560 x 1920) Industrial camera (CAM 1) and a low resolution (640 x 480) off-the-shelf web camera(CAM 2). 200 frames are captured with each camera and average time consumed by each stage is recorded. The car models are used as the target objects for this experiment.

Parameters : Time taken by each stage of the algorithm is measured for each frame and final results are produced by computing the average time of each stage. Time consumed by each stage is expressed in terms of % time of the overall tracking time.

Outcome :

Table 6.5 shows time consumed by each stage of the algorithm, with two different camera configurations. To provide the reader with better understanding, we designed two different experiments. First experiment tests time performance with different camera configurations, and no false positives in the scene (*Table 6.5*. In this case we can see that Image undistortion, Marker Detection and Pose estimation take up to 99% of computing time and only 1% time is taken by Marker Identification stage. High resolution camera performs tracking at the rate of 2-3 FPS (0.43 Sec/Frame), whereas web camera performs tracking at the rate of 7-8 FPS (0.14 Sec/Frame). It is to be noted that the complexity of matching algorithm might increase with number of model points or image points. This can be reflected from the Second experiment, where 90-140 markers are introduced in the scene. However, These markers are false positives for the matching algorithm.*Table 6.6*

Table 6.5.: Tracking performance, without false positives.

Algorithm Stage	CAM 1	CAM 2
Image Undistortion	39.53	11.79
Marker Detection	38.51	30.75
Descriptor Generation	0.10	0.33
Descriptor Matching	0.25	1.01
Pose Estimation	21.61	56.12

Table 6.6.: Tracking performance, with 90-140 false positives.

Algorithm Stage	CAM 1
Image Undistortion	12
Marker Detection	34
Descriptor Generation	0.2
Descriptor Matching	44
Pose Estimation	9.8

shows that in case of large number of false positives descriptor matching complexity increases and this stage takes up to 50% of the total tracking time. In this case tracking is done at the rate of 0.7 FPS (1.41 Sec/Frame).

CHAPTER
SEVEN

CONCLUSION & FUTURE WORK

In this work we have proposed a new tracking approach for Industrial Augmented Reality systems. We will summarize our learnings and emphasize on key points of our research contribution. At the same time limitations of the proposed tracking solution are also discussed and future research direction is outlined.

7.1. Conclusion

A tracking solution using Circular markers for a monocular camera system has a very good potential of replacing existing state of the art marker based solutions (Coded square or Circular patterns). This approach makes use of available tools (3D measurement system, circular markers) to provide reliable and robust tracking solution. Using this approach seamless work flow integration is possible with minimum work space intrusion (no specialised markers). The performance is able to meet requirements of real-time tracking with relevant hardware choices and software optimizations. In terms of usability the approach does not require the user to gain additional knowledge specific to the system. However, the user has to follow certain guidelines to achieve best possible tracking performance. These guidelines are mainly related to choice of markers for required working range and arrangement of markers based on object size.

In addition we have demonstrated a new method to perform a direct 2D-3D correspondence matching for a sparse set of 3D points using perspective invariant feature descriptors. Using this method we are able to extract 3D information of the conics existing on target object by analysing their 2D projections in the image. Further, a perspective invariant descriptor is generated from 2D conic projections. This descriptor is created from 3D data and therefore matched directly with 3D model information of the object. The method assumes that 3D information of the object is available. The recovered 3D information is of position and surface normal of the conic plane. This means that for a given set of conic projections in image 3D geometry information of world conic can be extracted. However, this method is not restricted to 3D objects and can also work on 2D objects with certain limitations. The surface normal information is usually achieved with help of depth camera and further used for matching 3D models. With our approach normal information is

achieved from normal 2D camera image given that circles exist in the Image. A unique feature of this method is that the descriptor always carries ambiguity in its structure, which is resolved using matching strategy. Also it is to be noted that the method does not require dense set of points to form local affine invariant features. On the contrary the method depends completely on perspective invariant 3D features created by sparse set of points on same or different planes.

7.2. Future Work

The circular maker based tracking approach is promising and may provide a vital solution for industrial AR in future, but as of now it has multiple limitations. These limitations are considered a primary focus for the further research work. We observed that the current system requires minimum 6 Correspondence pairs to compute a pose, however mathematically we require 3 points pairs if calibration information is available. The 3 point pose [HLON91] solving method provides 4 ambiguous solutions, one of which can be selected by using normal information extracted from 2D conics. This improvement may allow the user to operate with minimum 3 points in the scene. One other observation is that the Image undistortion and Image processing time can be reduced by developing faster methods by using GPU or parallelization. It is also possible to implement optical flow techniques to reduce frame by frame image processing effort and improve overall tracking speed. One of the most interesting extensions of the method would be introduction of machine learning and marker less tracking methods to create a new hybrid object tracking solution for Industrial AR. The application in this stage itself can be used for multiple industrial applications. The 3D data obtained is with highest possible accuracy and therefore registration errors can be minimised to a great extend. Moreover once the matching is performed eccentricity correction in the centroid measurements can be applied to get better accuracy in pose estimation [LRH06].

It is also important to understand that the approach provided in this work mandates use of known circles in scene for tracking. The future goal is to make a more generic system which can use natural curves of the image to extract 3D information. Moreover, extracting normal information from 2D image is a topic which can be further explored using methods such one proposed in this work.

Appendices

APPENDIX
ONE

CONIC PARAMETER CONVERSION

This part shows the method of computing ellipse parameters E from a Conic matrix C. A conic equation can also be written as,

$$x^\top Ax + 2x^\top B + D = 0.$$

This equation is similar to the general conic equation,

$$ax^2 + bxy + cy^2 + dx + ey + f = 0.$$

We know that conic matrix is written as ,

$$C = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}.$$

From these equations we get,

$$\begin{aligned} A &= \begin{bmatrix} a & b/2 \\ b/2 & c \end{bmatrix} \\ B &= \begin{bmatrix} d/2 & e/2 \end{bmatrix}^\top \\ D &= f \end{aligned}$$

Let $v(cx, cx)^\top$ represent centre position of ellipse (ellipse translation), which is calculated using

$$B = Av.$$

Further, compute R (eigenvectors) and $Diag$ (eigenvalues) using Eigenstate decomposition,

$$A = R^\top DiagR.$$

R represents a rotation matrix which rotates ellipse about angle θ,

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Let

$$Diag = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \text{and }, S = v^\top Av - D.$$

Now, Major axis (maj_a) and Minor axis (min_b) of Ellipse are computed as,

$$maj_a = \sqrt{\frac{S}{\lambda_1}}$$
$$min_b = \sqrt{\frac{S}{\lambda_2}}$$

It should be noted that values of maj_a, min_b are swapped if $maj_a < min_b$. In such cases θ is also reduced by 90°.

APPENDIX
TWO

CONIC BACK PROJECTION

This method assumes that a conic matrix C (image conic), camera calibration K and circle radius r are known. First step is to normalise the conic matrix using K,

$$C_n = K^\top C K.$$

C_n represents a cone with vertex at projection centre of the camera. Equation of the cone is given by,

$$ax^2 + bxy + cy^2 + dxz + eyz + fz^2 = P^\top C_n P = 0$$

where $P = [X\ Y\ Z]^\top$ is a point on the cone.

$$C_n = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}$$

The objective is to find a rotation matrix R', which rotates the image plane such that it becomes parallel to the target plane (world circle plane). R' is combination of two rotations R_1 and R_2. R_1 is a 3D-rotation matrix which rotates the camera coordinate frame to the eigenvector frame. $R_1 = (e_1, e_2, e_3)$, where e_1, e_2, e_3 represent the eigenvectors obtained by diagonalising normalised conic matrix C_n. $\lambda_1, \lambda_2, \lambda_3$ represent the eigenvalues of C_n in ascending order ($\lambda_1 < \lambda_2 < \lambda_3$).

$$C_1 = R_1^\top C_n R_1 = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix},\ \bar{P}_1 = R_1^\top \bar{P}.$$

This rotation removes $xz,\ yz,\ fz$ coefficients from the cone equation C_1. Further, a second rotation R_2 is applied such that $x^2,\ y^2$ coefficients for the cone C_1 become equal. This rotation is carried along the Y-axis by angle θ. θ introduces the ambiguity in the solutions.

$$C_1 = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \theta = \pm \tan^{-1} \sqrt{\frac{\lambda_2 - \lambda_1}{\lambda_3 - \lambda_2}}.$$

C_2 is the rotated cone equation, with this rotation image plane becomes parallel to the target plane. Ideally the conic in the image plane should become a perfect circle after this rotation.

$$C_2 = R_2^\top C_1 R_2,\ \bar{P}_2 = R_2^\top \bar{P}_1$$

At this stage normal of the target plane can be determined from last column of the R' matrix.

$$\bar{n} = R' \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^\top, R' = R_1 R_2$$

Negation of the last column of R' allows for conversion of the solution to right-handed coordinate system.

$$C_2 = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & 0 \\ -x_0 & 0 & x_0^2 - r_0^2 \end{bmatrix}$$

$(x_0, 0, 1)$ is the centre of the image circle in rotated coordinate system, r_0 is the radius of the rotated image plane. For C_2, coefficients of x^2 and y^2 should be equal. This constraint can be used to solve x_0 and r_0. The distance of target plane ($\bar{T}$) can be computed as follows,

$$d = \frac{r}{r_0} = r\sqrt{\frac{-{\lambda_2}^2}{\lambda_1 \lambda_3}}$$
$$\delta = r\sqrt{\frac{-(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_2)}{\lambda_1 \lambda_3}}$$
$$\bar{T} = R' \begin{bmatrix} \delta & 0 & d \end{bmatrix}^\top$$

The rotation matrix R' is used to obtain target plane distance in the camera coordinate system.

APPENDIX
THREE

SIMULATION RESULTS

Table C.1.: Results of Monte Carlo simulations for circle back projection method

Noise	Circle Orientation		$\varnothing = 5mm$		$\varnothing = 12mm$	
σ_{noise}	ψ	r	CEE	NEE	CEE	NEE
0.1	0	500	1.608	5.521	0.672	3.658
		1000	6.74	7.84	2.74	5.187
		1500	16	9.722	6.319	6.274
		2000	30.18	11.42	11.49	7.397
	35	500	1.195	0.4894	0.5431	0.2066
		1000	4.767	0.994	2.179	0.4159
		1500	10.73	1.546	4.917	0.6299
		2000	19.13	2.168	8.772	0.850
	70	500	1.789	0.255	0.7853	0.0994
		1000	7.371	0.5255	3.164	0.2016
		1500	17.15	0.8287	7.185	0.3083
		2000	31.62	1.117	12.91	0.421
0.3	0	500	5.045	9.872	2.045	6.252
		1000	27	14.7	8.796	8.82
		1500	72.16	19.19	21.39	11.02
		2000	148.5	23.49	41.15	13.04
	35	500	3.711	1.562	1.504	0.5913
		1000	15	3.696	5.971	1.23
		1500	34.2	6.952	13.37	1.965
		2000	60.35	11.85	23.7	2.834
	70	500	5.908	0.8213	2.152	0.292
		1000	28.6	1.971	8.713	0.6255
		1500	80	3.489	20.3	1.031
		2000	173	5.287	38.23	1.516

BIBLIOGRAPHY

[AIC] *AICON 3D Systems*. http://www.aicon3d.de/start.html

[ART] *ART : Advance Realtime Tracking GmbH*. http://www.ar-tracking.com

[Azu97] AZUMA, Ronald T.: A survey of augmented reality. In: *Presence-Teleoperators and Virtual Environments* 6 (1997), Nr. 4, S. 355–385

[BAT11] BERGAMASCO, Filippo ; ALBARELLI, Andrea ; TORSELLO, Andrea: Image-Space Marker Detection and Recognition Using Projective Invariants. In: *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, 2011, S. 381–388

[Bim06] BIMBER, Oliver: Projector-based augmentation. In: *Book Chapter in Emerging Technologies of Augmented Reality: Interfaces and Design* (2006), S. 64–89

[BKEC] BOSTANCI, Erkan ; KANWAL, Nadia ; EHSAN, Shoaib ; CLARK, Adrian F.: Tracking methods for augmented reality. In: *The 3rd International Conference on Machine Vision*

[Bro99] BROOKS, Jr. F.P.: What's real about virtual reality? In: *Computer Graphics and Applications, IEEE* 19 (1999), Nr. 6, S. 16–27. – ISSN 0272–1716

[Cap] *Capture 3D, Inc.* http://www.capture3d.com

[CGCMC07] CHEKHLOV, D. ; GEE, A.P. ; CALWAY, A. ; MAYOL-CUEVAS, W.: Ninja on a Plane: Automatic Discovery of Physical Planes for Augmented Reality Using Visual SLAM. In: *6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007. ISMAR 2007*, 2007, S. 153–156

[CMGJ99] CURTIS, Dan ; MIZELL, David ; GRUENBAUM, Peter ; JANIN, Adam: Several devils in the details: Making an AR application work in the airplane factory. In: *Proc. Int. Workshop on Augmented Reality (IWAR 1998)*, 1999, 47–60

[DC02] DRUMMOND, Tom ; CIPOLLA, R.: Real-time visual tracking of complex structures. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24 (2002), Nr. 7, S. 932–946

[DD95] DEMENTHON, Daniel F. ; DAVIS, Larry S.: Model-based object pose in 25 lines of code. In: *International journal of computer vision* 15 (1995), Nr. 1-2, S. 123–141

[DP11] DUNN, Fletcher ; PARBERRY, Ian: *3D math primer for graphics and game development*. Boca Raton, FL : A K Peters/CRC Press, 2011. – ISBN 9781568817231 1568817231

[EXT] *EXTEND3D GmbH*. http://www.extend3d.de/en/

[FG11] FITE-GEORGEL, Pierre: Is there a reality in Industrial Augmented Reality? In: *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, 2011, 201–210

[FH98] FARIN, Gerald E. ; HANSFORD, Dianne: *The geometry toolbox for graphics and modeling*. Natick, Mass. : A.K. Peters, 1998. – ISBN 1568810741 9781568810744

[Fia05] FIALA, M.: ARTag, a fiducial marker system using digital techniques. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005* Bd. 2, 2005, S. 590–596 vol. 2

[FMZ+91] FORSYTH, D. ; MUNDY, J.L. ; ZISSERMAN, A. ; COELHO, C. ; HELLER, A. ; ROTHWELL, C.: Invariant descriptors for 3D object recognition and pose. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 13 (1991), Nr. 10, S. 971–991

[FPF99] FITZGIBBON, Andrew ; PILU, Maurizio ; FISHER, Robert B.: Direct least square fitting of ellipses. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21 (1999), Nr. 5, S. 476–480

[Fri] FRIEDRICH, Wolfgang: ARVIKA : Augmented Reality for Development, Production, and Service. http://www.arvika.de/www/index.htm

[Git07] GITHENS, Greg: Product Lifecycle Management: Driving the Next Generation of Lean Thinking by Michael Grieves. In: *Journal of Product Innovation Management* 24 (2007), Nr. 3, S. 278–280. – ISSN 1540–5885

[GOM] *GOM GmbH: Optial Measuring Techniques*. http://www.gom.com/de/home.html

[GOO] *Google Glass Project*. http://www.google.com/glass

[HBN07] HINTERSTOISSER, Stefan ; BENHIMANE, Selim ; NAVAB, Nassir: N3m: Natural 3d markers for real-time object detection and pose estimation. In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007, S. 1–7

[HF04] HÖLLERER, T. ; FEINER, S.: Mobile augmented reality. In: *Telegeoinformatics: Location-Based Computing and Services. Taylor and Francis Books Ltd., London, UK* (2004)

[HLON91] HARALICK, R.M. ; LEE, D. ; OTTENBURG, K. ; NOLLE, M.: Analysis and solutions of the three point perspective pose estimation problem. In: *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, 1991, S. 592–598

[HZ03] HARTLEY, Richard ; ZISSERMAN, Andrew: *Multiple view geometry in computer vision*. Cambridge, UK; New York : Cambridge University Press, 2003

[IMH02] IPIÑA, Diego L. ; MENDONÇA, Paulo R. S. ; HOPPER, Andy: TRIP: A Low-Cost Vision-Based Location System for Ubiquitous Computing. In: *Personal and Ubiquitous Computing* 6 (2002), Nr. 3, S. 206–219

[KB99] KATO, H. ; BILLINGHURST, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: *2nd IEEE and ACM International Workshop on Augmented Reality, 1999. (IWAR '99) Proceedings*, 1999, S. 85–94

[KMK07] KIMURA, M. ; MOCHIMARU, M. ; KANADE, T.: Projector Calibration using Arbitrary Planes and Calibrated Camera. In: *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, S. 1–2

[KOB] KURZ, D. ; OLSZAMOWSKI, T. ; BENHIMANE, S.: Representative feature descriptor sets for robust handheld camera localization. In: *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, S. 65–70

[KPSL] KOLOMENKIN, M. ; POLLAK, S. ; SHIMSHONI, I. ; LINDENBAUM, M.: Geometric voting algorithm for star trackers. In: *IEEE Transactions on Aerospace and Electronic Systems* 44, apr, Nr. 2, S. 441–456

[LF05] LEPETIT, Vincent ; FUA, Pascal: *Monocular model-based 3d tracking of rigid objects: A survey*. Now Publishers Inc, 2005

[Low99] LOWE, David G.: Object recognition from local scale-invariant features. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* Bd. 2, 1999, 1150–1157

[LRH06] LUHMANN, Thomas ; ROBSON, Stuart ; HARLEY, Stephen Kyle: I.: *Close Range Photogrammetry: Principles, Techniques and Applications: Principles, Methods and Applications*. Revised edition. Whittles Publishing, 2006. – ISBN 1870325508

[MET] *METAIO GmbH*. http://www.metaio.com/

[ML05] MACINTYRE, Blair ; LIVINGSTON, M.A.: Guest Editors' Introduction: Moving Mixed Reality into the Real World. In: *Computer Graphics and Applications, IEEE* 25 (2005), Nr. 6, S. 22–23. – ISSN 0272–1716

[MYNW09] MOOSER, J. ; YOU, S. ; NEUMANN, U. ; WANG, Quan: Applying robust structure from motion to markerless augmented reality. In: *2009 Workshop on Applications of Computer Vision (WACV)*, 2009, S. 1–8

[Nav04] NAVAB, Nassir: Developing killer apps for industrial augmented reality. In: *Computer Graphics and Applications, IEEE* 24 (2004), Nr. 3, S. 16–20

[NF02] NAIMARK, Leonid ; FOXLIN, Eric: Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In: *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, 2002, 27

[Ope] *OpenCV*. http://opencv.org/

[PC02] PUPILLI, Mark ; CALWAY, Andrew: Real-time Structure from Motion for Augmented Reality. 2002. – Forschungsbericht

[PK07] PINTARIC, Thomas ; KAUFMANN, Hannes: Affordable infrared-optical pose-tracking for virtual and augmented reality. In: *Proceedings of Trends and Issues in Tracking for Virtual Environments Workshop, IEEE VR*, 2007, 44–51

[RBW05] REGENBRECHT, Holger ; BARATOFF, Gregory ; WILKE, Wilhelm: Augmented reality projects in the automotive and aerospace industries. In: *Computer Graphics and Applications, IEEE* 25 (2005), Nr. 6, S. 48–56

[RM04] RHIJN, Arjen van ; MULDER, Jurriaan D.: Optical tracking using line pencil fiducials. In: *Proceedings of the Tenth Eurographics conference on Virtual Environments*, 2004, S. 35–44. – Cited by 0008

[SA85] SUZUKI, Satoshi ; ABE, Keiichi: Topological structural analysis of digitized binary images by border following. In: *Computer Vision, Graphics, and Image Processing* 30 (1985), Nr. 1, S. 32–46

[SK06] SCHWERDTFEGER, Björn ; ; KLINKER, Gudrun: Hybrid Information Presentation: Combining a Portable Augmented Reality Laser Projector and a Conventional Computer Displa. In: *Proc. Shortpapers and Posters of 13th Eurographics Symposium on Virtual Environments, 10th Immersive Projection Technology Workshop(IPT - EGVE 2007)*, 2006

[Sut65] SUTHERLAND, Ivan E.: The ultimate display. In: *Multimedia: From Wagner to virtual reality* (1965)

[Sut68] SUTHERLAND, Ivan E.: A head-mounted three dimensional display, 1968, S. 757–764

[Tsa87] TSAI, R.Y.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. In: *Robotics and Automation, IEEE Journal of* 3 (1987), Nr. 4, S. 323–344. – ISSN 0882–4967

[UBI] *Ubitrack*. http://campar.in.tum.de/UbiTrack/WebHome/

[WAC+90] WANG, Jih fang ; AZUMA, Ronald ; CHI, Gary Bishop V. ; BISHOP, Gary ; CHI, Vernon ; EYLES, John ; FUCHS, Henry: *Tracking a Head-Mounted Display in a Room-Sized Environment With Head-Mounted Cameras*. 1990

[WCWL12] WENG, Dongdong ; CHENG, Dewen ; WANG, Yongtian ; LIU, Yue: Display systems and registration methods for augmented reality applications. In: *Optik - International Journal for Light and Electron Optics* 123 (2012), Mai, Nr. 9, S. 769–774. – ISSN 00304026

[Wel93] WELLNER, Pierre: Interacting with paper on the DigitalDesk. In: *Commun. ACM* 36 (1993), Juli, Nr. 7, S. 87–96. – ISSN 0001–0782

[Zha00] ZHANG, Zhengyou: A flexible new technique for camera calibration. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22 (2000), Nr. 11, S. 1330–1334

[Zha01] ZHANG, Luzi: *Camera calibration*. Aalborg University. Department of Communication Technology, 2001