A 3D TOPOLOGICAL TRACKING SYSTEM FOR AUGMENTED REALITY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF DEPARTMENT OF COMPUTER ENGINEERING
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MÜNİR ERCAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
DEPARTMENT OF COMPUTER ENGINEERING

FEBRUARY 2010

Approval of the thesis:

A 3D TOPOLOGICAL TRACKING SYSTEM FOR AUGMENTED REALITY

submitted by MÜNİR ERCAN in partial fulfillment of the requirements for the degree of Master of Science in Department of Computer Engineering , Middle East Technical University by,

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

Prof. Dr. Müslim Bozyiğit
Head of Department, **Department of Computer Engineering** _______________

Assist. Prof. Dr. Tolga Can
Supervisor, **Department of Computer Engineering** _______________

Examining Committee Members:

Assoc. Prof. Dr. Veysi İşler
Dept. of Computer Engineering, METU _______________

Assist. Prof. Dr. Tolga Can
Dept. of Computer Engineering, METU _______________

Assoc. Prof. Dr. Kürşat Çağıltay
Dept. of Computer Education and Instructional Technology, METU _______________

Dr. Çağatay Ündeğer
Dept. of Computer Science, BILKENT University _______________

Dr. Sinan Kalkan

____

Date: _______________

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

Name, Last Name: Münir Ercan

Signature :

ABSTRACT

A 3D TOPOLOGICAL TRACKING SYSTEM FOR AUGMENTED REALITY

Ercan, Münir
M.Sc., Department of Computer Engineering
Supervisor : Assist. Prof. Dr. Tolga Can

February 2010, 46 pages

Augmented Reality (AR) has become a popular area in Computer Science where research studies and technological innovations are extensive. Research in AR first began in the early 1990s and thenceforth, a number of different tracking algorithms and methods have been developed. Tracking systems have a critical importance for AR and marker based vision tracking systems became the mostly used tracking systems due to their low cost and ease of use. Basically, marker systems consist of special patterns that are placed in the environment and detected by simple cameras. In this thesis, we propose a new marker system based on topological tracking where markers are detected and identified using their topology trees. In our proposed marker system, we create topology trees of markers as region adjacency graphs that are obtained from binary images of the markers. Similarly, camera frames are also converted into binary images and corresponding topology trees are created. We used left heavy depth sequences as the canonical convention for the topology trees. For marker tracking, we used a simplified version of subgraph isomorphism algorithm that searches marker topology trees in a frame topology tree. Finally, we tested our proposed system for performance in using spatially distinct marker parts; occlusion resolving; detection rate with respect to marker size, camera-marker angle, false positive marker detection; performance with respect marker library size. Our system achieved 90% marker detection success with 50 pixels marker size and an average of 1.1 false positive marker detection in ten different test videos. We made all tests in comparison with the widely used ARToolkit library. Our system surpasses the ARToolkit library for all tests performed. In addition, our system enables spatially distinct placement of marker parts and permits occlusion unless the topology of the marker is not corrupted, but the ARToolkit library does not have these features.

Keywords: augmented reality, 3D tracking, topological tracking

ÖZ

ARTIRILMIŞ GERÇEKLİK İÇİN 3 BOYUTLU TAKİP KÜTÜPHANESİ

Ercan, Münir
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Y. Doç. Dr. Tolga Can

Şubat 2010, 46 sayfa

Artırılmış Gerçeklik (AG), bilgisayar biliminde araştırma çalışmalarının ve teknolojik yeniliklerin çok çeşitli olduğu popüler bir alan haline gelmiştir. AG alanındaki araştırmalar ilk defa 90'lı yılların başında başlamış olup, ilerleyen yıllarda birçok takip algoritması ve yöntemi geliştirilmiştir. Takip sistemleri AG için kritik öneme sahiptir ve işaretçi tabanlı takip sistemleri kolay kullanımı ve düşük maliyeti ile en çok kullanılan takip sistemi olmuştur. Temel olarak, işaretçi sistemleri çevreye yerleştirilmiş ve kameralar tarafından algılanan özel şekillerden meydana gelmektedir. Bu tezde, işaretçilerin topolojik ağaçları kullanılarak algılandığı ve tanındığı topolojik takip tabanlı yeni bir işaretçi sistemi öneriyoruz. Önerdiğimiz sistemde, işaretçilerin topolojik ağaçlarını, işaretçilerin ikili resimlerinden elde edilen bölge bitişiklik çizgesi olarak oluşturuyoruz. Aynı şekilde, kameradan alınan kareleri de ikili resme çevirip ve karşılık gelen topolojik ağaçlarını oluşturuyoruz. Topolojik ağaçların gösterilmesinde standart yapı olarak sol ağırlıklı derinlik sırasını kullandık. İşaretçi takibi için, işaretçi topolojik ağacını, video karesi topolojik ağacı içinde arayan bir altçizge benzerlik algoritması kullandık. Son olarak, sistemimizi işaretçi parçalarının bağımsız yerlere yerleştirilmesi; kapatılma çözümü; işaretçi büyüklüğü, kamera-işaretçi açısı ve yanlış pozitif işaretçi takibine göre takip oranları; işaretçi kütüphanesi genişliği ve performansı başlıklarında test ettik. Sistemimiz 50 piksellik işaretçi büyüklüğünde %90 lık takip başarısına ve 10 değişik video kullanılarak yapılan testlerde 1.1 yanlış pozitif işaretçi takip ortalamasına ulaşmıştır. Bütün testlerimizi yaygın olarak kullanılan ARToolkit kütüphanesi ile karşılaştırmalı olarak gerçekleştirdik. Yapılan bütün testlerde sistemimiz ARToolkit'den daha iyi bir başarı sergiledi. Ayrıca sistemimiz işaretçi parçalarının bağımsız olarak yerleştirilmesine ve işaretçi topolojisi bozulmadığı sürece kapatılmaya izin vermektedir, ARToolkit de bu özellikler bulunmamaktadır.

Anahtar Kelimeler: artırılmış gerçeklik, 3 boyutlu takip, topolojik takip

To My Family

ACKNOWLEDGMENTS

First of all, I greatly thank Dr. Tolga Can for advising me in the master program and for his guidance, encouragement, and invaluable contributions throughout this research.

I wish to extend my sincere thanks and appreciation to examining committee members for their extensive and constructive comments on this study.

I also owe great thanks to my friends and Rotasoftians Mustafa Ayan, Furkan Kuru, and Gürkan Caner Birer for their help and contributions to this study.

My thanks are also due to my fellow Mehmet Kayım for his critical assistance when I got stuck during thesis writing period.

Finally, I would like to thank to Hilal and my family for their tolerance and support which were most valuable during this research.

This research was supported by The Scientific and Technological Research Council of Turkey with Project No: 3070204 and RotaSOFT Software Technologies.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Augmented Reality (AR) is based on adding virtual objects on real world images in real time. AR enables coexistence of virtual and real objects in a real environment. An AR system runs at real time and enables interaction between real and virtual objects. In order to create a simple AR system, two basic units are needed: a tracking framework and a display device. Tracking is the most important part of AR where objects from the real world are detected and 3D positions and orientations of the objects are extracted. Alignment of virtual objects with real objects is done by using this extracted 3D information. Many methods including optical, inertial, acoustic, mechanical, magnetic methods have been employed for tracking. Each method has advantages and disadvantages which are discussed in Section 2. After getting the 3D information from the environment, 3D virtual objects are placed on real environment images relative to the tracked object. Second part of AR is the visualization of the mixed world. The most popular display device for AR is the usual computer monitor which is cheap and easily accessible. The ad hoc device that is mostly used is Head Mounted Display (HMD). HMDs are typical head-worn displays that are used to visualize the mixed world in front of user's eye similar to using ordinary glasses. There have been many different display technologies including head-worn, handheld and spatial displays which are discussed in Section 2.

Azuma [3] states that a tracker should satisfy three main properties. First, a tracker should make pose estimation with a small error in both position and orientation. Second, latency between tracker and graphics engine should be very low. Finally, a tracker should also work at far distances. Maybe, we should add another requirement that a tracker should be easy to use by offering a comfortable usage. Also, it must be cheap if it is intended to be used by mass market. Vision based tracking is one of the most popular tracking method used in AR with their good pose estimation, low latency, easy usage and cheap price. Vision based methods employ cameras and computer vision techniques in order to track real objects. Special objects or markers are used for tracking in order to enhance the tracking ability. Fiducials are special pictures that are placed in the real environment. These pictures have specially chosen patterns that make them easy to recognize at changing environmental conditions. Different types of fiducials have been developed and utilized until now. Circle and square fiducials are used for different situations. ARToolkit is the most known AR library based on printable fiducials. Although ARToolkit provides good pose estimation of the camera, it has disadvantages. ARToolkit uses geometry of the fiducials for their detection. Using geometry for tracking has no tolerance to occlusion. Simple occlusion prevents the detection of the fiducial and results in a false positive. After detecting a fiducial in the frame, the ID of the fiducial is attained by using pattern matching algorithms. When there are many types of fiducials, pattern matching becomes exhaustive and ineffective resulting in too many false positives and the requirement of high computational power.

The design of fiducial systems has critical importance on the tracking success. Charles et al. [19] examine the fiducial design by means of shape, color, size and interior image. Best fiducial images are determined according to these criteria. They proposed that a fiducial should have at least four points that can approximate it to a square. Monochrome fiducials are chosen instead of color ones. A black border surrounding the interior image is used. Interior image of a fiducial should be chosen from a set where there is small correlation among the members. A fiducial system that employs these properties is considered to be the best fiducial. Although fiducial systems created until now uses most of these properties, tracking and pose estimation algorithms make a great difference on the performance of the system.

In this thesis, we propose a new fiducial system for Augmented Reality. Our system is based on topological tracking of fiducials. We created a library that takes the live stream from camera frame by frame and process each frame in order to detect the markers. Frames and fiducials are converted into topologic trees and fiducials in the frame are searched by using a subgraph isomorphism algorithm. Occlusion is allowed as long as the topology of the fiducial is not disturbed. Considering pattern matching, a lower computational power is required since tracking is done by searching a tree in another tree.

1.2 Related Work

From the beginning of the concept of Augmented Reality, vision based tracking methods have been very popular. The basic method that is used in vision based AR is using fiducials for tracking. Fiducials are printed papers that have special pictures. Main properties of a fiducial are shape and pattern. Until now, circular and square fiducials have been proposed.

Circular fiducials have the advantage that center of the circle is invariant to orientation of the marker. Cho et al. [9] proposed a circular marker system for AR systems. Their system uses multi-ring fiducials that have different colors at each ring. Each fiducial is recognized by its colors of homogenous rings. Camera pose is calculated by using three or more fiducials. Naimark and Foxlin [18] created a 2D barcode system for circular fiducials. Markers are created by using contrasting concentric circles (CCC) which have consecutive white and black regions. Their system was able to generate thousands of different fiducials and track them by using homomorphic image processing. Abad et al. [2] proposed a system that calculates the pose of the camera by using two concentric circles. They used this technique to present a simple calibration method for camera. Ababsa and Mallem [1] proposed a simple circular marker system that uses fiducial of a black circle surrounded by a white circle. Detection is done by using an ellipse fitting algorithm. Pose estimation is done by using moving camera and stable markers. The main advantage of the system is that it enables occlusion since occluded circles may also fit to an ellipse.

Square markers are the most popular models that enable more robust systems and better camera pose estimations. Kato and Billinghurst [17] proposed ARToolkit library which uses square markers that have black borders surrounding nonsymmetrical patterns. Fiducials are detected by searching square shapes in the frame. A square shape is recognized as a fiducial if its interior image matches with a stored pattern in the system. Any nonsymmetrical pattern placed in a black bordered square can be used as a marker. The ARToolkit library uses four corners of the fiducial to estimate the pose of the camera.

Rekimoto and Ayatsuka [20] proposed a marker system that is designed as 2D barcodes. 2 dimensional patterns have NxN squares where each square has black or white values. Arrangement of the pattern of the marker specifies the ID of the marker. Their system requires low computational power and supports thousands of markers since it does not use pattern matching.

Fiala [13] proposed a marker system that is based on the ARToolkit. Their system provides edge based marker finder that does not need thresholding. Also 2D barcode markers are employed; there are no stored patterns of markers. The proposed system can generate more than 2000 markers and requires lower computational power than the ARToolkit since pattern matching is not used for identification of markers.

Claus and Fitzgibbon [10] proposed a machine learning method that uses four points to track the fiducials. They trained a classifier with different fiducials under varying lighting and environmental conditions. For each frame, sub windows surrounding each pixel are classified as fiducials or not. Their system has 95 % overall performance at indoor and outdoor scenes which is superior to all known fiducial systems. The main drawback of their system is its high requirement of computational power.

Ababsa and Mallem [11] used Orthogonal Iteration (OI) algorithm in order to estimate pose. Their proposed system is used for square markers with four corners. OI algorithm is used to attain camera parameters by matching 2D image features with their 3D features. Results show that OI is a robust and effective algorithm for pose estimation for real time AR applications.

Jo et al. [16] proposed a new generation method that enables tracking of forces applied by user on the marker. This method uses square markers and tracks fingertip motions. Forces exerted by the user are calculated and virtual objects connected to the marker are manipulated according to these forces. This marker system enables natural interaction between real and virtual objects which is very important for AR applications.

Wang et al. [22] proposed an infrared marker based AR system. IR markers that are invisible to user are projected on objects by using IR projectors. IR cameras are used to detect the markers and estimate pose. This method eliminates the disturbing appearance of the markers in a real environment.

1.3 Contributions

Our contribution in this thesis is a proposal of a new fiducial system for Augmented Reality. This system is based on topological tracking of fiducials. It promises spatially distinct usage of each part of fiducials, low computational power requirement, resolving occlusion and a better recognition method.

Each fiducial in our system consists of two parts. These two parts should be placed on a surface independently. Each part is recognized individually and pose estimation is done by using combination of two parts. Using one big marker is always a problem for AR applications since it occupies a big space on printed materials. Two independent parts make it easy to distribute the fiducial to different places on printed material. This provides efficient placement of text and visual materials on printed pages.

Our system is based on topological tracking of fiducials. A topologic tree which demonstrates the containership information of black and white regions is created from each frame grabbed from the camera. Topologic tree of fiducials are searched in the topologic tree of the frame. This is done by using a simplified subgraph isomorphism algorithm. Our system requires low computational power since geometry or interior image of the fiducial is not used for recognition.

Occlusion is one of the biggest problems in fiducial systems. The user may obscure some part of the fiducial while moving or playing with it. No tolerance to occlusion makes the system seem unreliable and inconsistent. Our system resolves the occlusion until topology of the fiducial is not disturbed. Any hand occlusion that is done on the border of the fiducial is resolved successfully. It is also possible to create more occlusion tolerant fiducials in our system by only expanding the borders of the fiducials.

Most of the systems created until now use pattern matching methods in order to track fiducials. These systems detect probable fiducials in the frame by considering fiducial shape. Then, detected probable fiducials are mapped to screen coordinates in order to make pattern matching. This process corrupts the probable fiducial image and causes false positives at tracking. Our system employs topologic tracking method that does not use pattern matching of images. Failures caused by pattern matching of images are avoided and a better tracking performance is achieved.

1.4 Thesis Outline

In Chapter 2, the background on Augmented Reality, tracking methods and display technologies are presented. Chapter 3 describes the overall structure of our system. Test results and comparison of our system to the highly popular ARToolkit system are given in Chapter 4. Conclusion and future work are summarized in Chapter 5.

CHAPTER 2

BACKGROUND

2.1 Introduction to Augmented Reality

Augmented Reality (AR) is a way of creating and using virtual objects on computer like Virtual Reality (VR). In VR, virtual objects are created in a virtual world and everything takes place at that synthetic world. However, AR provides usage of the real world where the real and virtual objects take place. Virtual objects are used as a part of the real world. In VR, user cannot see the real world while immersed in the system. AR provides vision of the real world with virtual objects augmented on it. All parts of VR are generated on computer as a supplant to reality whereas, in AR, only virtual objects are generated on computer as supplement to the real world. AR is a way of adding virtual objects on real environment providing coexistence of reality and virtuality. AR can be defined by its three important properties [4].

1. It combines real and virtual world in real time
2. It is interactive
3. It registers real and virtual objects

The aim of AR is to combine the real and the computer generated virtual world. Figure 2.1 shows where the reality, virtuality and AR intersect. AR stands between the real and the virtual world. AR starts from the real world and approaches to the virtual world by addition of virtual objects. When there is no vision from the real world, AR becomes a VR application.

AR provides a natural interaction between real and virtual objects. Virtual objects in AR can be manipulated by using hand, head or some part of the body. As it can be seen from Figure

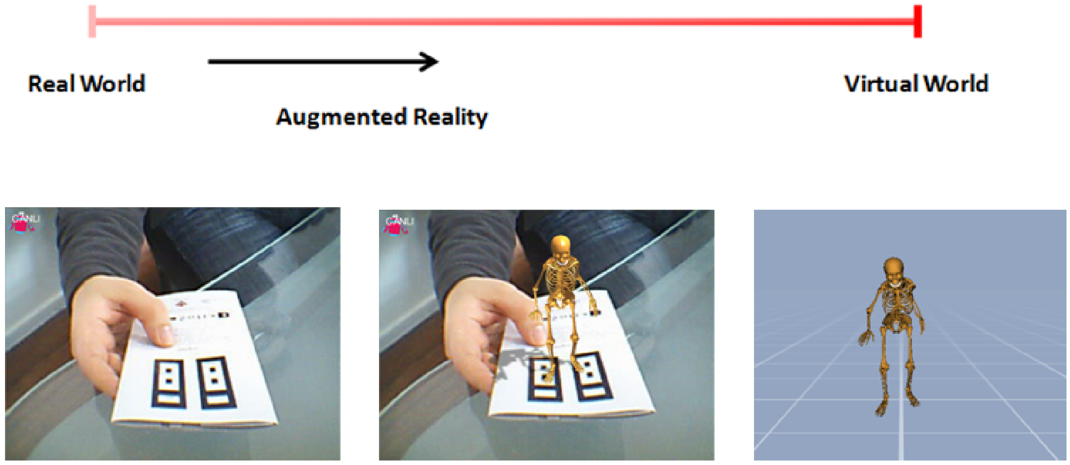


Figure 2.1: Real world, AR and virtual world.

2.1, a virtual skeleton can be moved by using the hand.

An AR system consists of two important parts: tracking device and display device. In order to perform a successful AR application, suitable technologies for both parts should be carefully analyzed.

2.2 Tracking Methods

Tracking is the base part of Augmented Reality. Tracking is the process of recognizing the reference object and estimating its 3D position and orientation. Registration of the virtual and real objects is done by using this 3D information. Any small errors on this information results wrong registration which decrease the effectiveness of the system. A good tracker system enables coexistence of real and virtual objects that demonstrates the virtual objects as a complementary part of real world. Azuma states that a good tracker should have three main properties. First, a tracker should have small errors on both position and orientation. Most of the time small errors at tracking results in large errors at alignment. For example, only 1.5 degrees of error at head tracking system causes more than 5 cm error at virtual object alignment. Position error should not be more than 1 to 2 mm and orientation error should be a small fraction of a degree. Second, latency of tracker must be very low. Latency is the time passed between the tracker measurement done and corresponding image appear in the display device. Latency of the system should not be more than 2 ms. Third, tracker should work at long distances. Many systems have been proposed that satisfy these properties. Different methods based on sound, light, magnetism etc. are employed on these systems. But satisfied properties created new drawbacks. Lines of sight problem, magnetic interference, low update rate, uncomfortable usage are the main examples. Each method has its own advantages and disadvantages. While one method allows wide area usage, other method enables high update rate. Another method should provide comfortable usage or easy calibration. Best method is chosen according to user preferences. Advantage and disadvantages of each method is described in this section.

A tracking system mainly consists of two components which are targets and sensors. Targets are the reference objects that are tracked in the environment. Sensors are the devices that track the sensors in the environment. Tracking systems are divided into two categories which are inside-out and outside-in according to placement of sensors and targets. Figure 2.2 [23] shows the design of each configuration. At outside-in configurations, sensors are fixed and targets (landmarks) are placed on moving objects. At inside-out configurations, targets are fixed and sensors are placed on moving objects. Both systems have advantages and disadvantages. Outside-in configuration is easier to setup and cheap. Targets are small and it is easier to place them on fixed surfaces in the environment or on moving objects. Sensors are big and heavy; most of the time it is not possible to place them on moving objects. Inside-out configuration should be more suitable for wide area tracking where sensors should move independently to very long distances.

2.2.1 Inertial Tracking

Inertial tracking uses accelerometers and gyroscopes in order to track the position and orientation. This system is based on Newton's second law of motion, F=ma. Accelerometers are used for calculating position. Accelerometers are created by using a small mass and a spring. One end of spring is fixed and other end is attached to the mass. When no force is applied the mass is at rest. If a force is applied to the mass, mass moves and gets a displacement. The measure of this displacement is used to measure the force exerted on the mass. Then, acceleration of the mass is calculated by using Newton's second law of motion.

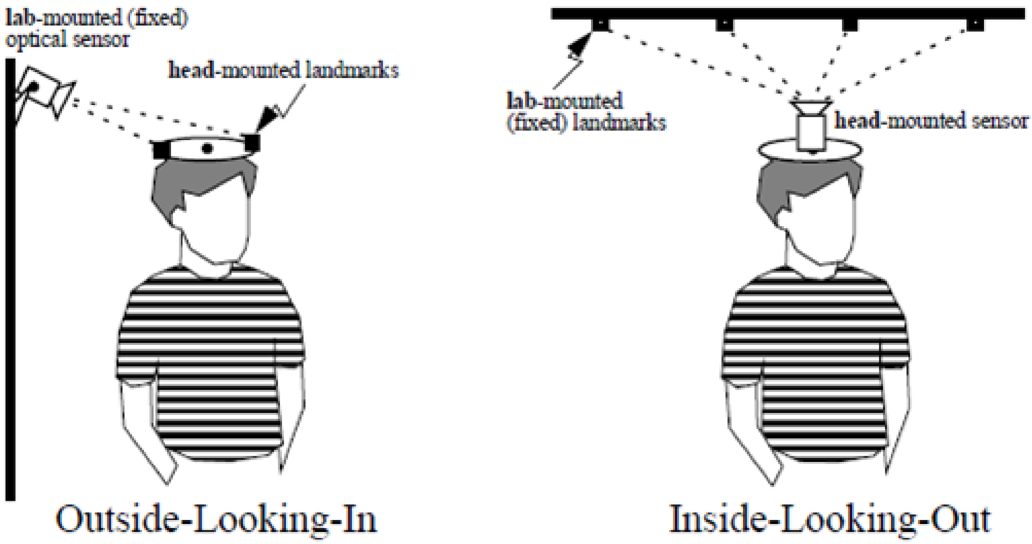


Figure 2.2: Outside-in and Inside-out Tracking [23]. (Courtesy of Greg Welch)

Gyroscopes calculate the angular acceleration by using the same method.They measure orientation based on the principle of conservation of angular momentum. Gyroscopes mainly consist of a mass called rotor which is spinning on one axis, spin axis, at a very high rate. High rate of spinning makes a large angular momentum which makes the system more irrelevant to the external forces. A rotor is mounted on the gimbals in order to measure the change in direction. The angle which gyroscope gets from its gimbals is the measure of its angular momentum.

An accelerometer or gyroscope tracks the orientation or position only for one axis. In order to get three dimensional information, three accelerometers and gyroscopes that are placed perpendicular to each other should be used. Inertial tracking systems have advantages and disadvantages. The basic advantage of the inertial systems is that there is no line of sight problem. Also there is no effect of magnetic interference. Inertial tracking systems present a small medium which are easy to use and mount. The disadvantage of the system is error drift. Error at each measurement is accumulated which results in a drift of error. Inertial systems should be recalibrated periodically in order to reset the error. Most of the time inertial trackers are used in hybrid systems where the other system is used for recalibration.

2.2.2 Optical Tracking

Optical tracking uses light in order to track the objects. Two types of targets and corresponding sensors are used in optical tracking. Passive targets should be markers and reflective materials which have no power. Markers are special pictures that are spread to the environment. Markers are the mostly used optical tracking method since it is very cheap and easy to use. More information about marker systems is given in Section 2.4. Video or CCD (Charged Coupled Device) cameras should be used as sensor for passive targets. Video and CCD cameras use image processing techniques for estimating 3D position and orientation. Active targets are LEDs (Light Emitting Diode) that are powered. Photodiodes should be used as sensor for active targets. Photodiodes use current that is proportional to the position of the light to estimate 3D information of the target.

Optical tracking systems have high update rate and accuracy. There is no effect of magnetic interference. Optical tracking systems mostly depend on the intensity of light sources. Quality of images that are input to the sensors establishes the success of the system. Also there must be a clear line-of-sight between the target and the sensor. Hi-Ball tracker [23] is an example optical tracker that uses photodiodes and LEDs for tracking. ARToolkit [17] is an example optical tracker that uses passive fiducials and video cameras.

2.2.3 Magnetic Tracking

Magnetic tracking uses magnetic fields in order to measure 3D position and orientation. Low frequency AC fields or pulsed DC fields should be used. Coils that are carrying current create the magnetic field. Three orthogonal coils are placed in both source and receiver. Coils at the source are activated serially. Activated coils create a magnetic field causing a voltage change in coils of the receiver. This voltage change in receiver coils is proportional to the strength of detected magnetic field. The strength of the detected magnetic field is related to the distance and angle between source and receiver. For each coil in the source, three measurements are estimated from the receiver. Totally nine measurements are estimated. These nine measurements are used to estimate the relative position and orientation of the source to the receiver.

Magnetic tracking provides tiny targets and sensors that are easy to mount and use. There is no line-of-sight problem at magnetic tracking. But magnetic interference causes a high distortion on the system. So, magnetic systems should not be used around metal objects. Also there is high latency for these systems.

2.2.4 Acoustic Tracking

Acoustic tracking uses ultrasonic sound waves in order to track targets. Ultrasonic sound has a frequency level of 20 KHz which is higher than human audible sound. Sound waves emitted from the source create a sphere. In order to get 3D position of a target, there should be one source-three receivers or three sources-one receiver. In order to get 3D position and orientation, there should be three sources-three receivers. Position and orientation are estimated by using time-of-flight or phase coherence methods. Both methods use speed of sound which is nearly 340 m/s. Time of flight between receiver and source is calculated by using constant speed of sound. Calculated time is used to estimate the distance between the source and the receiver. Phase coherence uses the phase difference of sound waves between source and receiver in order to get the distance between them. Estimated distance in both methods is used for measuring 3D position and orientation.

The main drawback of the acoustic systems is that energy of sound waves diminishes as the distance between receiver and source increases. This makes the system not suitable for wide area tracking. Acoustic tracking uses constant speed of sound for calculating time of flight and phase coherence. However, speed of sound changes at different temperature and pressure. So, acoustic tracking can only be used at controlled environments. Also there must be clear line-of-sight for accurate tracking. Acoustic systems have high latency which causes a delay because of slow speed of sound.

2.2.5 Mechanical Tracking

Mechanical tracking uses joint angles and lengths in order to track 3D position and orientation. Mechanical tracking systems are created by fastening mechanical objects each other with joints. There must be only one known position in a system; 3D position and orientation of all other parts can be deduced from the known point by using joints angles and lengths. Mechanical systems may be ground based or body based. In ground based systems, known position is fixed to the ground. These systems restrict the area of motion. In body based mechanical systems, known position is fixed to a part of body. Exoskeletons are examples of body based systems. These systems may restrict the motion of the body because of their big size and heavy weight.

Bend sensors, gears and potentiometers are used in mechanical tracking for estimating 3D information. Mechanical tracking systems are robust and fast. They have good update rate. There is no interference with magnetic fields. However, the main drawback of the mechanical systems is that they mostly restrict movement area and motion of body. Long time usage is not supported because of their heavy weight.

2.3 Display Technologies

Display technologies in AR aim to present the virtual world overlaid onto the real world. Different methods have been employed for this reason until now. According to Eitoku et al.[12], an AR display should have four properties: (1) Virtual and real objects should be placed together. (2) Provide collaborative environment (3) Does not limit user by various parts (4) Displays 3D images. According to the area of usage, display technologies should be divided into three categories which are head-worn displays, handheld displays and projective displays [4]. Head-worn displays are placed on the head like glasses. Handheld displays are mobile devices that are placed on the hand. Projective displays are used for projecting virtual image directly onto the real objects and placed independent from the user. There are different possible displays that should be suitable for different situations. Figure 2.3 [8] shows the different combinations of placement of displays according to the user and real world. Whereas retinal displays directly augment virtual world on retina of the user, spatial displays augment virtual world on real object. Two types of images are created by AR displays which are planar and curved images. Curved images are created by projectors and retinal displays. Planar images are created by the displays that create images on planar surfaces.

2.3.1 Head-worn Displays

Head-worn displays (HWDs) are placed on head of the user. They create the augmented image in front of the users eyes. There are two types of HWDs which are video-see-through

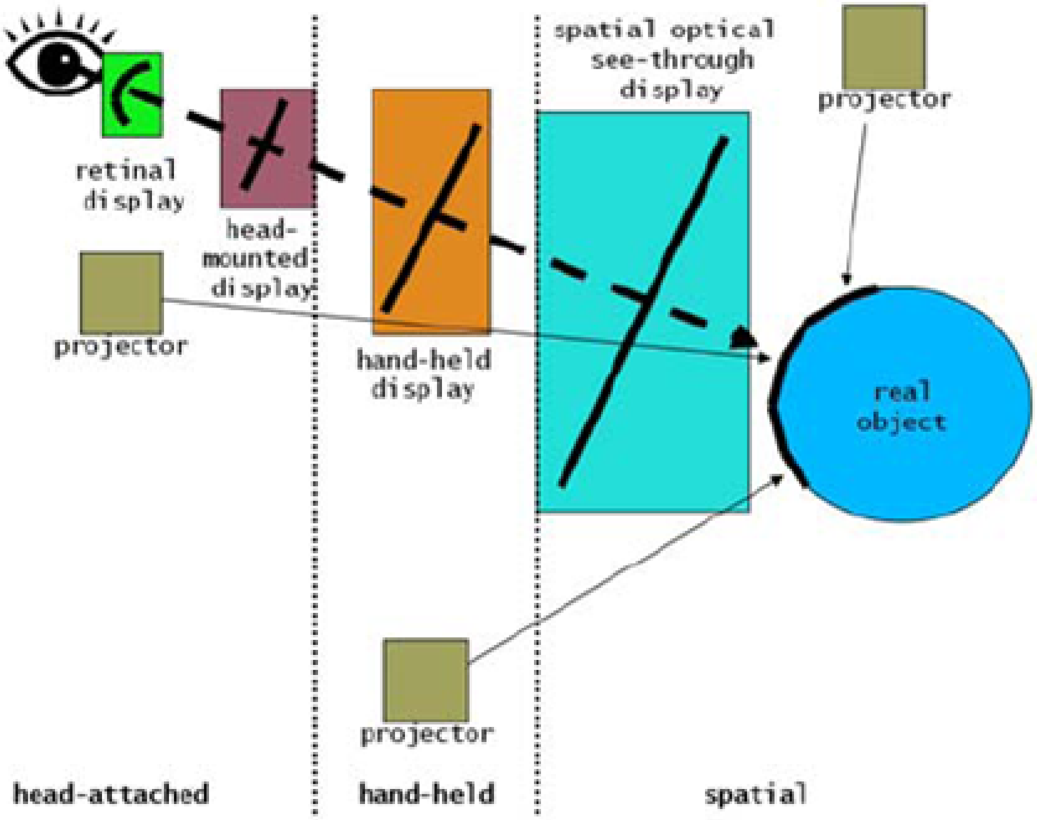


Figure 2.3: Categorization of Display Technologies [8]. (Courtesy of Oliver Bimber)

and optical-see-through. Video-see-through HWDs use video mixing where real time video is mixed with virtual images. Optical-see-through HWDs create the augmented image by adding virtual images directly on glasses or on user eyes. Transparent lenses are used for optical-see-through HWDs in order to enable real scene while displaying virtual image. Head-mounted displays, retinal displays and head mounted projectors are typical HWDs. Head-mounted displays use screens in front of users eyes in order to demonstrate the augmented image. Real scene is taken from video camera as video stream. Video stream is augmented with virtual image and sent to the LCD screen that is placed in front of users eyes. HMDs are the most direct way of representing the augmented image. But long time usage disturbs users because of their heavy weight. Also HMDs suffer from parallax error which is caused by the distance between eyes and tracking camera. Retinal displays project virtual images directly on to the users eye without need of LCD screens. Low power lasers are used to form image on retina. Retinal displays provide high brightness and contrast with wide field of view. Main drawback of retinal displays is that only red images are displayed since low power blue and green lasers are too expensive.

2.3.2 Handheld Displays

Typical handheld devices like cell phones, Personal Digital Assistants (PDAs), tablet PCs and Mobile Internet Devices (MIDs) can be used as Handheld Displays (HHDs). HHDs are very popular for their small size, easy availability, wide usage and high mobility. Cell phones with cameras tend to become one of the most popular HHDs for Augmented Reality being the most probable device that can carry AR to mass market. HHDs mostly use built-in cameras as trackers and monitors as displays. However, there are also optical see-through HHDs and hand held projectors. For example, Stetton et al. propose an optical see-through HHD for demonstrating tomography data on the patient [14]. Raskar et al. propose a hand held projector for demonstrating shadows of virtual objects. Main drawbacks of HHDs are low computational power and low resolution cameras. Low computational power limits the methods that can be used for AR on HHDs and low resolution cameras cause poor quality images. Also, small screen size with narrow field of view brings along uncomfortable usage. HHDs are alternative displays to HWDs for mobile applications. However, HHDs do not provide hands-free usage like HWDs.

2.3.3 Spatial Displays

Spatial displays are based on augmenting the real world without need of user attached devices. Real world is augmented directly by using projectors or augmented scene is shown by using detached displays like screens. A usual computer monitor is the most popular screen based display. Monitors use video mixing where real world and virtual world are mixed and shown on monitors. Monitors have shortcomings of video mixing methods which are low resolution vision of the real world and narrow field of view. There are also optical see through spatial displays. Transparent screens [7] are used for adding augmented image on real world by using detached optical see through displays. Transparent screens have shortcomings of optical see through HHDs plus immobility. Projectors are used for projecting the augmented image directly onto the real objects. Projector based spatial displays supports multiple users and wide area usage. They have no restriction on body and no disturbing devices. Main drawback of projector displays is that they rely on surfaces of the real objects. In order to fully augment an object, all visible parts of the objects should be projected by a projector. Also user and objects on real environment create shadows that prevent the augmentation of shadowed parts.

2.4 Marker Based Vision Systems

Augmenting the real world starts with getting the 3D position and orientation information from the environment. The best, also the most desirable way of getting 3D information is to process natural environment images and locate the virtual objects in the environment. However, processing natural environment images does not provide reliable results to track real objects and consumes high computational power. Because of these reasons, new generation vision methods use special pictures located in the environment to estimate 3D information. These special pictures are called fiducials or markers. Markers have special patterns that are easy to recognize by simple computer vision techniques.

Figure 2.4 shows the overall structure of marker based vision systems. Frames coming from live video stream is processed at first step. Processing the frame makes it ready for marker search. Simple methods just reveal the probable marker regions while advanced methods also process the probable regions. Marker search is done by matching probable regions with known marker designs. 3D position and orientation of the detected markers are estimated. Virtual objects are overlaid onto the real environment by using 3D position and orientation information.

Designing good marker systems has critical importance on tracking success. Shape, size, color, interior image design are the basic properties of a marker system that should be carefully handled. Owen et al.[19] examined the fiducial design by means of these properties. They state that a successful marker design should possess markers that have wide black borders. Wide black borders define the marker area and make it easy to be recognized. Monochrome markers are more reliable than colored markers. Because, changing environmental and lighting conditions disturb the color of the marker. Interior images of the markers should be chosen from a set that have small correlation among members. A successful marker should have four corner points that approximate it to a square. At least four corners are needed in order to make a good estimation of 3D position and orientation.

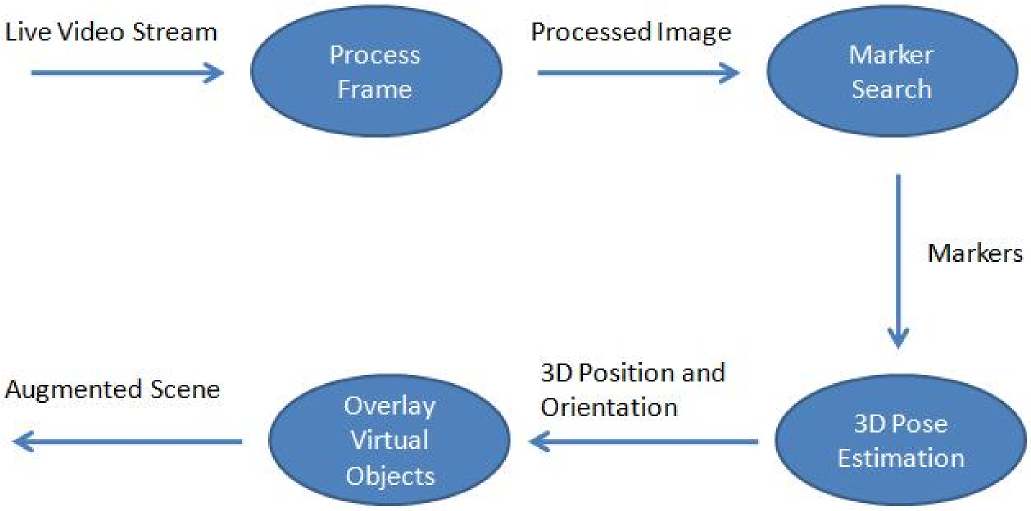


Figure 2.4: Overall Structure of Marker Based Vision Systems.

Tracking algorithms and pose estimation methods are also very important in creating successful designs. In addition to these properties, markers should be easy to use in order to be used by the mass market. For example, a marker with size 30 cm height to 30 cm width will not be a good design for consumers. Detection of a marker in a frame mostly depends on its shape. Detected markers are identified by using shape matching or barcode reading methods. Current marker systems should be categorized according to their shapes. Two highly known and used shapes are circular and square.

2.4.1 Circular Markers

Circular markers have the advantage that, the marker centroid is invariant to orientation. Viewing direction has no effect on 3D position estimation. An oriented circle fits to an ellipse and an ellipse contains all the information to estimate the pose of the circle. Circular marker tracking starts with determining probable marker regions in the frame. Contours of the detected probable regions are tried to fit to ellipses using ellipse fitting algorithms. Then, 3D position and orientation of the marker is estimated by using 3D transformation algorithms. Ellipse fitting algorithm establishes the quality of the system since 3D information is estimated from ellipse region. Least square algorithm applies a least square ellipse fit to the detected region. Least square algorithm has a good approximation of the 3D information but requires high computational power. Simple fit method uses center point of the detected region. Then, longest and shortest distance from the center point is used to determine the ellipse fit. 3D transformation algorithms use ellipse back projection methods in order to estimate the 3D position and orientation. 3D virtual objects are placed on the environment by using this 3D information as a reference point.

Three different circular marker systems that are proposed until now is shown in Figure 2.5. Simple circular marker system [1] consists of black circles surrounded by circular white border. This is the simplest form of a circular marker. Multi-ring color circular markers [9] utilize colors to identify the different types of the markers. Ellipse fitting algorithms are used to get 3D pose estimation and color sequence is used to identify the marker. Using 6 colors, more than 300 unique markers can be created. Lastly, the circular barcode markers [18] are used as 2D barcode in order to identify the markers. Circular barcode systems allow identification of thousands of unique markers.

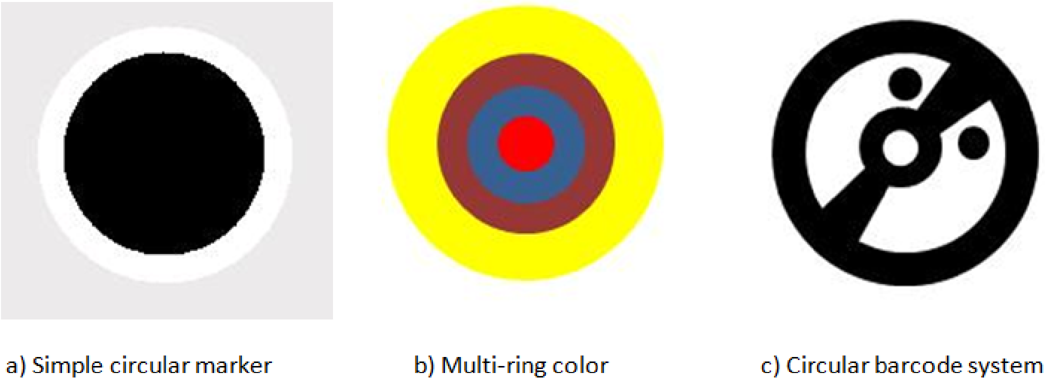


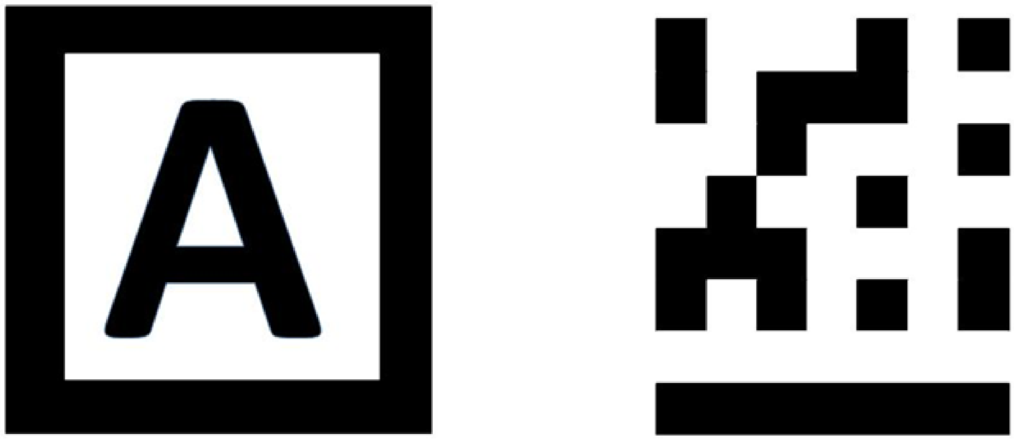
Figure 2.5: Circular markers used in AR.

2.4.2 Square Markers

Square markers have the advantage that they have four corners. At least four corners are needed in order to make good quality pose estimation. Three points should be used for pose estimation but this may lead to confusion which should be resolved by making further estimations. Tracking of square markers starts with determining probable regions for markers.

Probable regions are the ones that have square shapes. Contours of the probable regions are fit to four lines and four corners are extracted from four intersecting lines. Projective transformation algorithms are used in order to get 3D information from four points. Projective transformation, also called homography, is a combination of consecutive perspective projections. Homography describes the relationship between two observed positions of one object in two different scenes. These two positions are related to each other by a homography function. This homography function is used for getting rotation and translation of camera between two scenes. For the case of augmented reality, square markers are used as reference objects that define the homography function. Pose of the camera according to markers is estimated.

The estimated pose is defined as a transformation matrix which includes rotation, scale and translation. Examples of two well-known marker systems are shown in Figure 2.6. ARTookit markers [17] employ a black square border. Interior image surrounded by the black border could be any nonsymmetrical pattern. Markers are identified by using a pattern matching method. After getting the 3D transformation matrix, interior image is normalized to screen coordinates. Stored shape of the pattern is compared with the normalized detected pattern. CyberCode markers [20] are defined as 2D barcodes. Pattern of the marker consists of zero-one pixels which are recognized by CCD or CMOS cameras.



a) An example ARToolkit marker.

b) An example CyberCode marker.

Figure 2.6: Square markers used in AR.

CHAPTER 3

MATERIALS AND METHODS

3.1 Overall System Architecture

We propose a marker based augmented reality system that utilizes topologic properties of markers for tracking. In our system, markers are detected and distinguished by their topology. Topologic properties of the markers are stored as a tree. Frames grabbed from the camera are converted into topology trees. Search for markers in the frame is carried out by searching for marker topology trees in the frame topology tree.

Tracking starts with taking an image from the camera. The image is converted into a binary image (black and white image) by using dynamic thresholding. Connected components of the binary image are extracted by using segmentation. By using connected components as a node, topology tree of the binary image is created. Topology tree is an unordered rooted tree that holds containership information of the image. Nodes correspond to the connected components of the image. Children of a node are the regions that are contained by that node. Root is the background region of the image. Markers have special topology trees. Tracking is done by searching the tree of a marker in the topology tree of the image. This is done by using a simplified subgraph isomorphism algorithm. After tracking the markers, our system uses geometry of the marker for 3D pose estimation. We use two markers in order get better estimation of 3D transformation matrix. We decided to use corners of the two markers since corners will give the biggest geometry that will end up with the highest accuracy. We used homography estimation methods of OpenCV library in order to get 3D transformation matrix of 4 corner points. Then, this transformation matrix is used as the transformation of 3D objects to be placed with reference to the marker. Figure 3.1 shows the overall architecture of our system. In the following subsections we describe each stage of our framework in detail.

3.2 Thresholding

Stream from the camera is processed frame by frame. In thresholding, a frame is converted into a binary image (black and white image) by using a threshold value. Determining the threshold value is the key point for successful conversion. In global thresholding a constant threshold value is set for all the frames. Global thresholding has some drawbacks. For example if lighting is poor, then all frames are converted into a black image. If lighting is high, then frames may be converted into a white image. In order to overcome this problem, threshold values should be determined dynamically. A threshold value of each frame should be determined by taking the average of pixels at that frame. But most of the time, the threshold value that is suitable for half of the frame is not suitable for the other half. Even most of the time, a threshold value for one pixel is not suitable for other pixels.

In our system, any frame is converted into a binary image by using adaptive thresholding [6]. In adaptive thresholding, threshold values are determined by using a window (32 X 32) for each pixel. The threshold value is calculated by using the average of the pixels in the 32x32 window. After calculating the threshold values, pixel value is set to white if it is greater than the threshold value or set to black otherwise. By using only black and white values, a binary image is created.

3.3 Segmentation

After thresholding, the next step is segmentation. Segmentation is done in order to group white and black regions in the frame. A region growing method is used for segmentation. Segments are grown by comparing the intensity values neighbouring pixels with the mean of the intensity values of the pixels of the segment. The segmented image is converted into a region adjacency graph. The region adjacency graph of a binary image is a tree that holds the containership information of segmented regions and the adjacency information for these type of images always reduces to ”contains” information [21]. Root of the tree is the background of the image. Each node of the tree is a segmented region. Children of each node are the segments inside the segment corresponding to that node. Leaves are the segments that contain

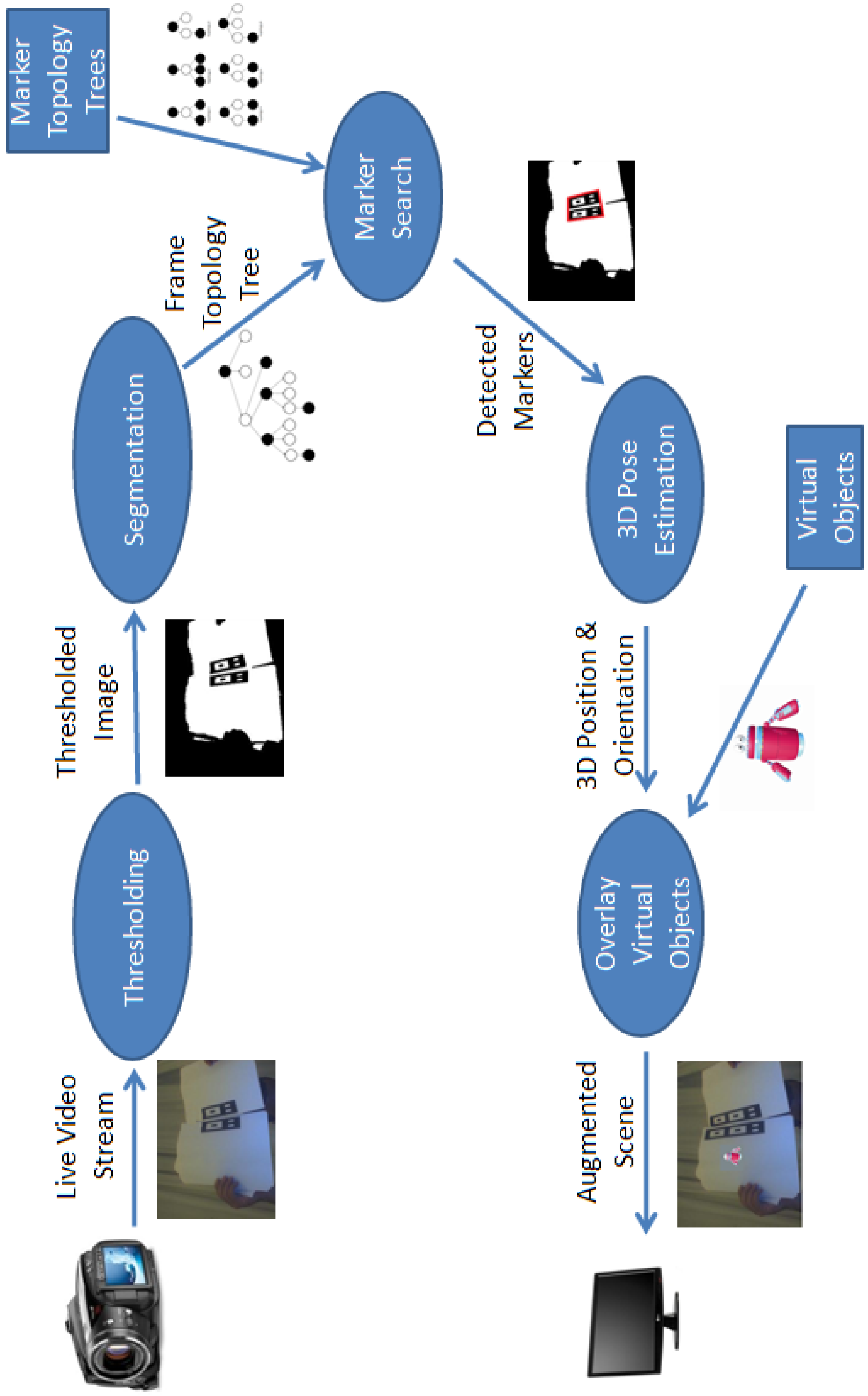


Figure 3.1: Overall System Architecture

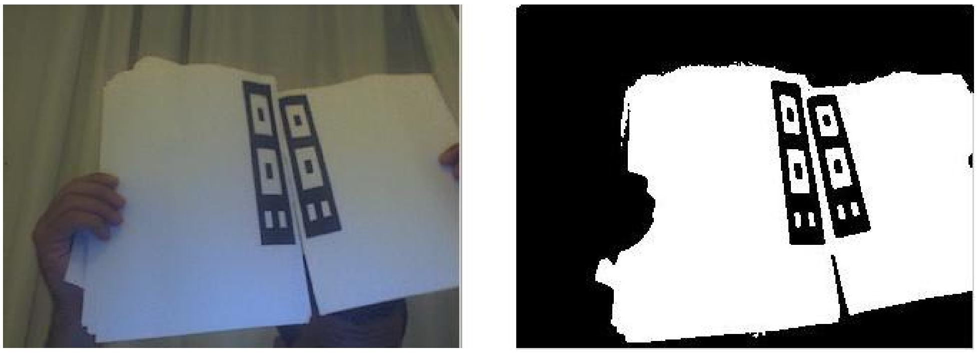


Figure 3.2: Original frame and thresholded frame

no segments inside them.

The created region adjacency graph is a topology tree and it is irrelevant to size and geometry. Topology tree just keeps the region information of segments containing other segments. Since size and geometry are not important, same topology may lead to many different shapes (Figure 3.3).

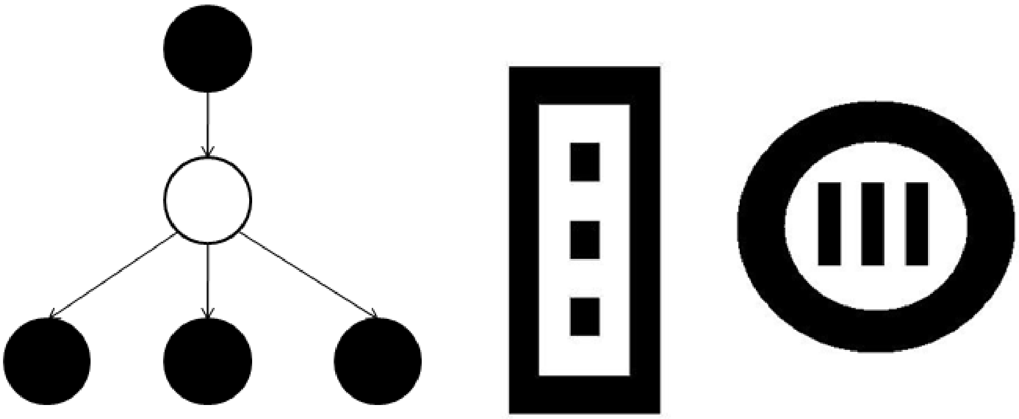


Figure 3.3: Topology tree with two different geometries.

Also there is no order among nodes. A shape may lead to different trees. Our system does not care about the order of nodes in the tree but sorts the nodes according to their depth in order to keep recognition simple. As it can be seen from Figure 3.4, one geometry can have more than one topology tree. Since the order is not important, nodes are sorted according to their depth and resultant topology tree is used for recognition. So, the created topology tree is an unordered rooted tree.

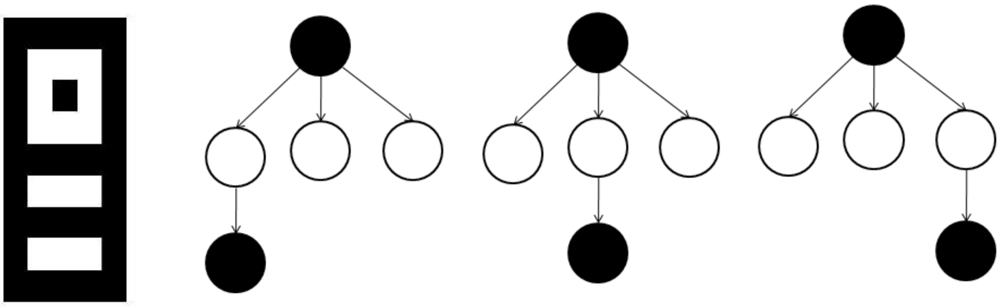


Figure 3.4: A geometry with three possible topology trees.

3.4 Recognition

Segmentation creates the topology tree of the frame. On the other side, we have the topology tree of markers stored in the application. Recognition process is done as a search of a tree in another tree which can also be regarded as the subgraph isomorphism problem. Although the subgrahp isomorphism is an NP complete problem, we used canonical conventions and rules to reduce the complexity to linear time. Recognition starts with detecting candidate nodes. Candidate nodes are the nodes that have a chance to be a marker. Candidate nodes are detected according to two properties which are the number of leaves and tree depth. For example, when a set of markers are created by topology trees of height 3, then any node at topology tree of frame with height greater or less than 3 cannot be candidate node for that marker set. If this set of markers also requires that a marker should have at most 4 leaves, then nodes with more than 4 leaves cannot be a candidate node.

Topology of the candidate nodes are compared with topology of the markers. In order to make the recognition part easier, a canonical convention for topology trees is needed. We used left heavy depth sequences [15] for this purpose.

Depth sequence of a tree is described as the sequence of depth values that came across at preorder traversal of the tree. Depth value of root is 0 and depth value of any other node is number of edges between root and that node. Each node has the value of its depth. While preorder traversal of the tree, these node values create a sequence of numbers which is used to identify that tree. This makes a useful convention for ordered trees where each node of the tree has a specific order. Since the order of visiting children makes different sequences, this convention is not suitable for unordered trees. For the geometry at Figure 3.4, the sequences (0,1,2,1,1) , (0,1,1,2,1) and (0,1,1,1,2) are possible. In order to make depth sequence notation suitable for unordered trees, left heavy depth sequences are used. Depth sequences get the left heavy property by moving the deepest nodes to the left. To create left heavy depth sequences, nodes with bigger depth values are visited first during the traversal of the tree. Figure 3.5 shows a topology tree and its corresponding left heavy depth sequence tree. Left heavy depth sequence of the tree is written in the figure caption.

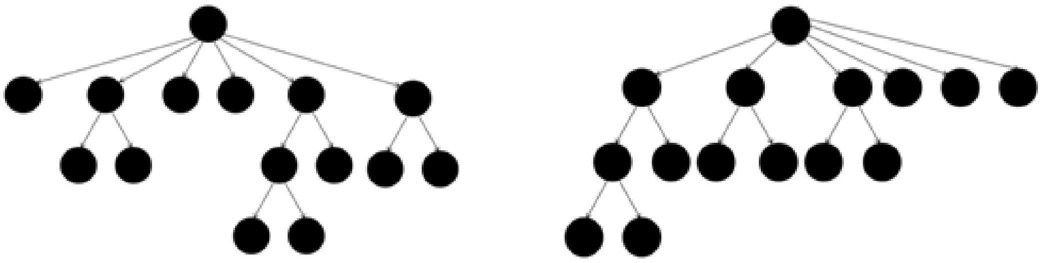


Figure 3.5: Topology tree and its ordered left heavy depth sequence tree. Left heavy depth sequence of the tree is (0,1,2,3,3,2,1,2,2,1,2,2,1,1,1).

3.5 2D Location and Orientation

After recognition of the marker, the main part is finding the 2D location and orientation of the marker. While creating the topology tree of a marker, oriented bounding boxes for each segment is stored in nodes. The center of the bounding box is a good approximation of the location of the segment. Using the center of the bounding box may lead to wrong results when the area of the bounding box is too large or shape of the bounding box is not circle or square. In order to increase the accuracy of the approximation, leaf areas are used. Since leaf areas are the smallest regions in the markers, they would give the most accurate information. Firstly, the centroid of each leaf of marker is calculated. Then, averages of centroids of black and white leaves are calculated separately. Average of centroids of black leaves of a marker is called b-centroid and average of centroids of all white leaves of a marker is called w-centroid. Orientation of the marker is calculated as a vector from b-centroid to w-centroid. There are also symmetric markers that have no 2D orientation value. If there is no black leaf or there is no white leaf or the b-centroid of the marker equal is to the w-centroid of the marker then marker is assumed to be symmetric and has no orientation value. Our system enables usage of symmetric markers to some extend. Our proposed tracking system requires two markers to obtain 3D transformation matrix and one of these two markers may be symmetric. Both of the markers cannot be symmetric since orientation value cannot be extracted from two symmetric geometries.

The location of the marker is extracted by using the bounding box of the whole marker. 4 corners of the marker give the 2D location of the marker. So, all of the proposed system works with rectangle markers. B-centroid and w-centroid values should also be used as location values. But using corners of the marker creates a bigger geometry that will be used for estimating the transformation matrix. Bigger geometry brings more accurate estimations and results. While geometry of the marker is not important for tracking, it is important for obtaining location and orientation. This property makes our system a hybrid system which uses topology for tracking and geometry for location and orientation estimation.

3.6 Design of The Marker Set

3.6.1 Marker Design

Markers are the images that will be recognized by the system. Since our system works based on topology of markers, markers are created by using different topology trees. In order make a border for markers; all topology trees of markers have a black area as the root. Creating simple but effective markers is extremely important in marker tracking since recognition is done according to topologies of markers. Effective marker is the one that is small in size but complex enough to be recognized by the software. Small markers are easy to use and they could be placed into printed documents easily but complex markers tend to become large in size. There are two important factors in creating effective markers: tree depth and number of leaves.

Tree depth is the most important part for creating markers. When tree depth is small, markers become so simple and small. For example, tree depth of 1 provides topology trees that consist of only root and white leaves. Figure 3.6 shows the three possible topology trees and their corresponding geometries. These topologies are not complex enough to make effective recognitions since there will be lots of similar topologies in the image. An ordinary image may contain black nodes with many white leaves. All these nodes will be recognized as a marker. This increases the number of false positives and makes the system unreliable. So small tree depths creates simple markers but not effective ones.

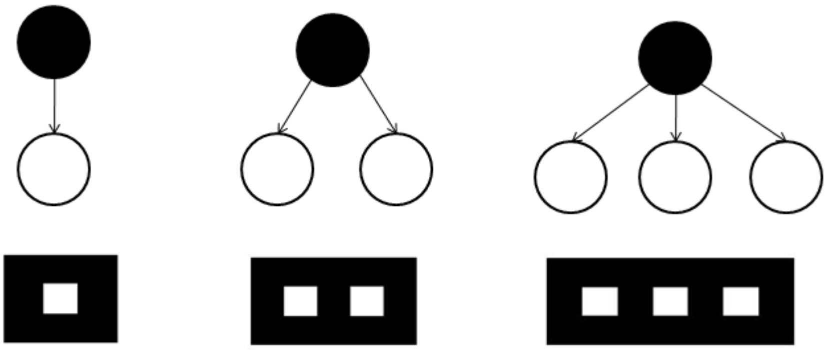


Figure 3.6: Three possible topology trees for tree depth of 1 are shown above. For each topology tree, corresponding geometry is shown below.

When tree depth is too large, markers become very large. Figure 3.7 shows the simplest topology tree and geometry that can be created by using tree depth of 3 and the simplest topology tree and geometry that should be created by tree depth of 4. If we sketch the markers pixel by pixel, tree depth of 3 will have at least 7 pixels at width and tree depth of 4 will have at least 9 pixels at width. Both geometries are very large in width for our marker system since we use two markers side-by-side to obtain the 3D transformation matrix which is discussed in Section 3.6.2.

Also the maximum number of leaves is very important for creating simple but effective markers. If this number increases, then the size of the markers would increase dramatically. Also, setting the maximum number of leaves to a large number would increase the number of candidate nodes at recognition part and end up with a decrease on performance. Large number of leaves will make very complex markers. Each added leaf will add a new area to its geometry. Complex markers are easier to recognize but also open to confusion. A simple mistake may lead to failure at tracking. In order to overcome this problem, most of the time, tolerance values are used. Tolerance values give the chance to accept recognition even though 100 % recognition is not achieved. For example, a marker with 6 leaves is accepted as recognized

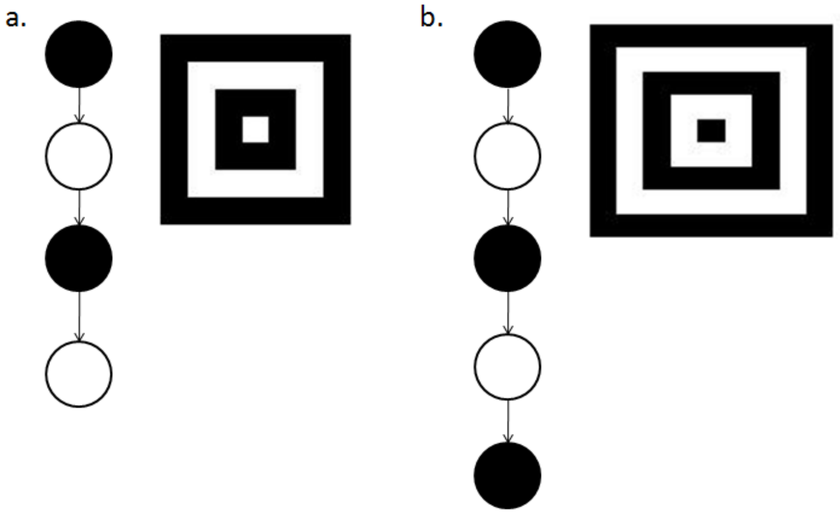


Figure 3.7: a. The simplest topology tree and the simplest geometry for tree depth of 3. b. Simplest topology tree and simplest geometry for tree depth of 4.

when 4 of its leaves are recognized with tolerance value of 2. In order to increase reliability, we did not choose this method. While very simple markers increases the number of false positives, very complex markers increases the number of false negatives.

Since tracking by topology uses segments from the image, the thinnest area of the geometry will be weakest area which will cause the failure for tracking. In order to prevent making weak regions, all segments of a marker are created proportional to each other. Each region is created pixel by pixel and width and height of the regions are created by using the same minimum width and height values.

We set tree depth to 2 in order to generate simple but effective markers. The outer region is always set to black in order to clarify the borders of the marker. Maximum number of leaves is set to 3. Width of the overall marker is determined as 5 pixels. Because, tree depth of 3 does not provide smaller width markers. Height of the overall marker is set to 9 pixels since it gives us the possibility to create 6 markers that is enough for our system. Corresponding topology trees are shown in Figure 3.8.

Figure 3.9 shows the eliminated topologies that have tree depth of 2 and maximum number of leaves is 3. First topology in this figure is not complex enough to be an effective marker. Other topology trees in this figure require more than 9 pixels in height. If we need more

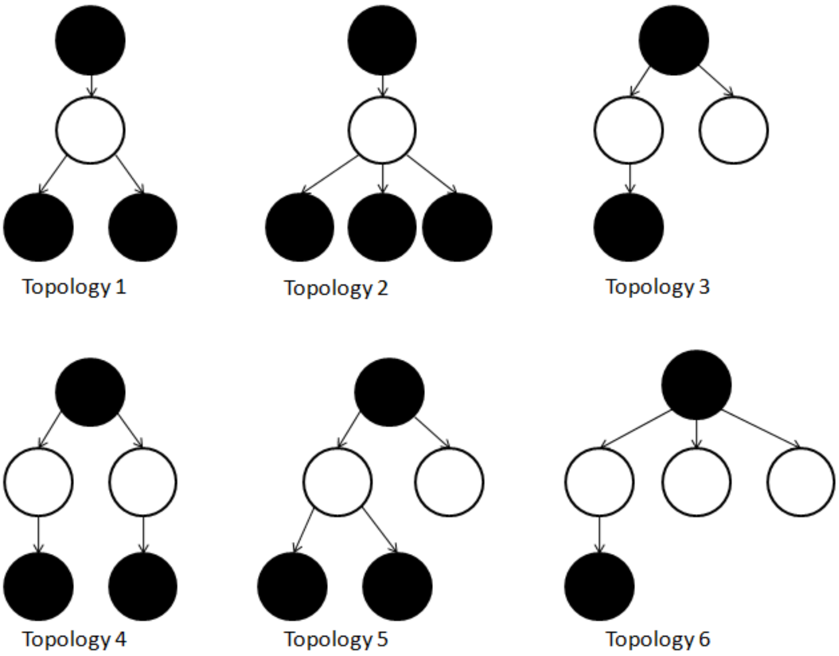


Figure 3.8: Topology trees when tree depth is 2 and maximum number of leaves is 3.

markers we may also use these topologies by increasing the height of the markers.

All trees are created as left heavy depth sequence trees and their corresponding left heavy depth sequences are written. For each topology in Figure 3.8, a geometry is created (Figure 3.10. Creating effective geometries is also important. As we demonstrated before, a topology tree may lead to very different geometries. Choosing the best geometry increases the usability and performance of the system very much. We sketched our geometries as rectangles since we use their corners for 3D pose estimation. We should use symmetric regions for leaf segments since we use their center points to extract orientation. Squares, rectangles or circles are all possible geometries for leaf segments. In this thesis, we used rectangles.

Topology is the structure of the marker. A structure turns into different geometries. So, different markers can be created by using the same topology. Even by using the rules mentioned above, topology trees may lead to different geometries. From topology 3 of Figure 3.8, three possible geometries can be created (Figure 3.11). One of these topologies is symmetric which we allow in our system but do not prefer if we have a chance to construct an asymmetric one. The remaining two geometries are both possible and effective markers. Topology tree, 2D location and orientation, corner points and also 3D transformation matrix of both markers are

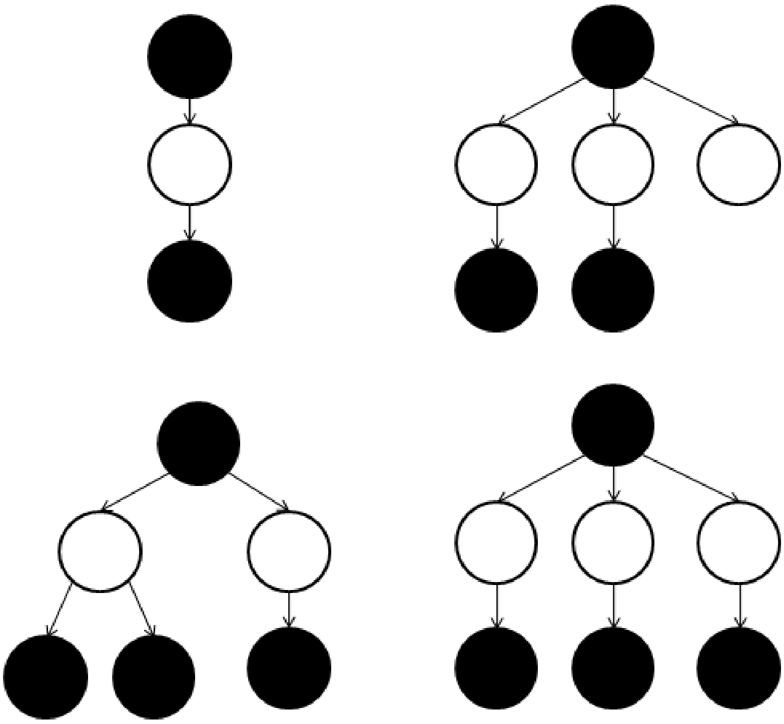


Figure 3.9: First topology is eliminated since it is not complex enough. Other topologies are eliminated since they require more than 9 pixels in height.

the same. So they can be used instead of each other.

3.6.2 Bi-marker Design

After recognition of the marker, 2D location and orientation of the marker is calculated. In order to use this information for Augmented Reality, we have to find 3D location and orientation of the marker. Each marker has 2D location as its four corners and 2D orientation. These four corners should be used for extracting 3D information but marker size should be large enough to get reliable results. Tracking errors has big effects on small markers. Markers that are created in previous section are small in width whereas large in height. In order to make balanced markers, we designed a new marker system that consists of pairs of markers that are placed side-by-side.

After generation of topology trees of markers, we end up with six different markers. We use binary combinations of these 6 markers, we will call bi-marker from now on, to create a geometry for augmented reality. A bi-marker consists of two markers located side-by-side.

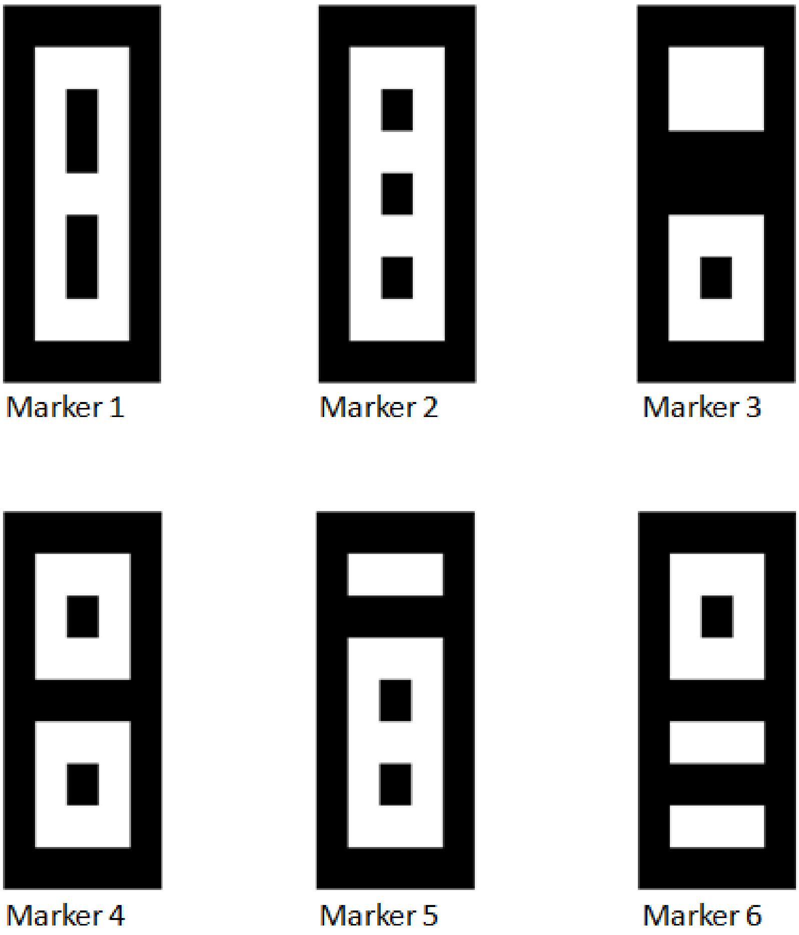


Figure 3.10: Markers when tree depth is 2 and maximum number of leaves is 3.

A bi-marker is identified according to the markers it includes. 2D orientation of marker is also important. Same markers with different 2D orientations according to each other create a different bi-marker. Markers in Figure 3.12 are different markers since they have different 2D orientations. One asymmetric marker is enough to get the orientation of the bi-marker. But two symmetric markers should not be used since orientation of the bi-marker cannot be found. So symmetric markers are allowed in our system but both markers of a bi-marker cannot be symmetric. A number of different bi-markers from our marker set are presented in Appendix A.

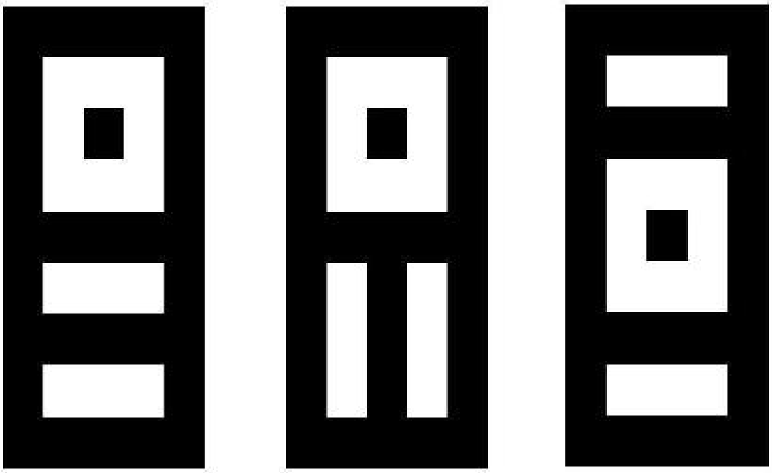


Figure 3.11: Possible markers that should be created by using topology 2 of Figure3.8 .

3.7 Obtaining Transformation Matrix

4 corners of a bi-marker are extracted from outer corners of two markers (Figure 3.13). These four corners are used to get the 3D transformation matrix. We have used the homography function of OpenCV, in order to get the 3D transformation matrix of a bi-marker detected from the camera. Homography function of OpenCV gets source points and destination points and outputs homography matrix that includes transformation between two points. We used the ideal screen coordinates as source points and outer corners of the detected marker as destination points.

Homography is used to get the 3D transformation of the points from one 2D frame to the other. Obtained transformation matrix includes rotation, translation and scale which shows the transformation of bi-marker according to the camera. In augmented reality, bi-markers are used as reference points. All 3D contents are placed by using these reference points. In order to put a 3D object on the bi-marker, firstly, objects should be transformed by using the transformation matrix of the bi-marker. Then the 3D object should be rendered at that place.

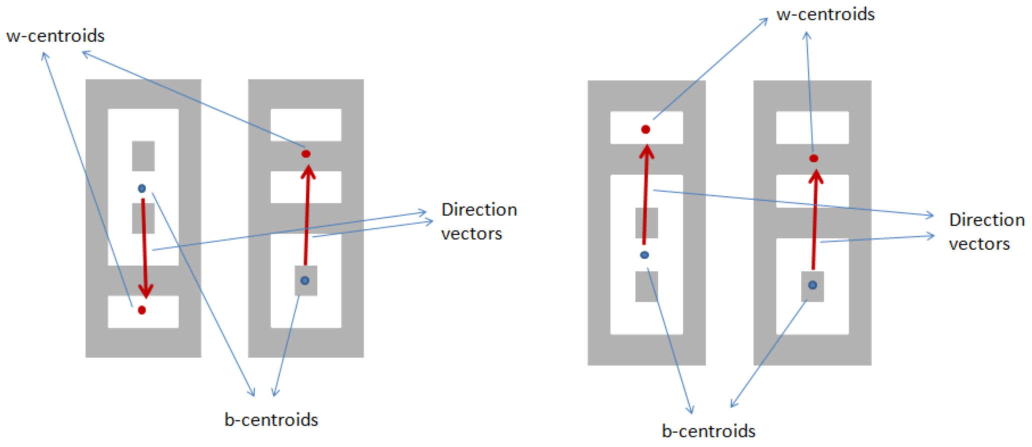


Figure 3.12: Two bi-markers which consist of same markers with different 2D orientations.

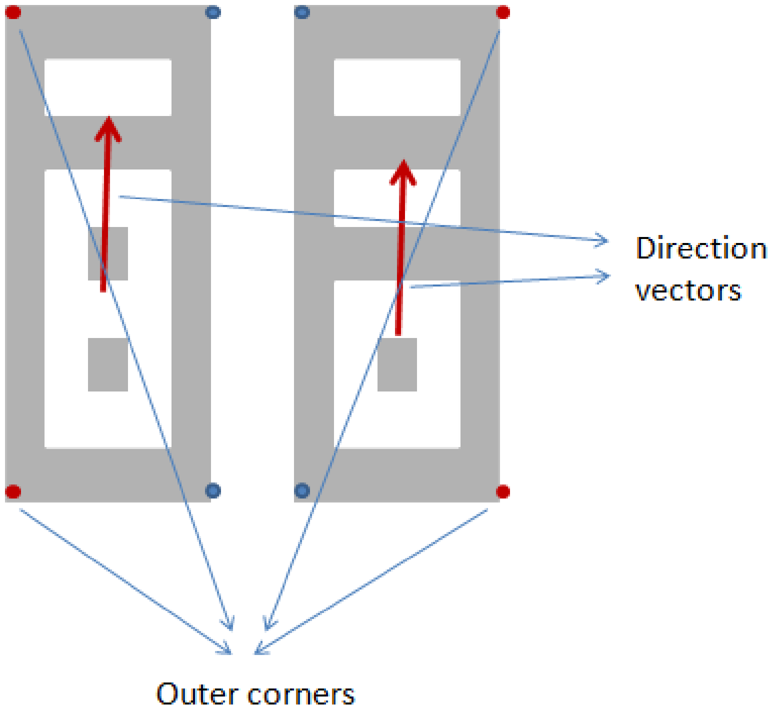


Figure 3.13: 2D direction vectors and four outer corners that are used for obtaining the transformation matrix.

CHAPTER 4

RESULTS

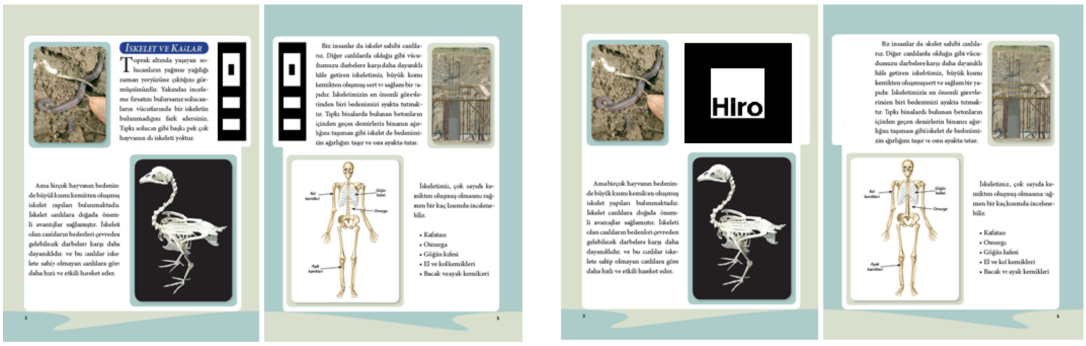
We compared our system with the ARToolkit library according to: (i) spatially distinct usage of marker parts; (ii) detection rates; (iii) marker library size and performance; (iv) occlusion handling. All tests were done on a system with AMD Athlon 64 X2 Dual Core 4000+ 2.11 GHz processor and 1 GB RAM. Microsoft LifeCam VX-7000 was used for marker tracking by setting the resolution to 640x480.

4.1 Spatially Distinct Usage of Marker Parts

Our system provides a better usage of markers on printed materials compared to other systems such as ARToolkit. For most of the time, developers used markers consisting of a single piece. However, a single piece marker with 12-20 cm of edge length generally prevents an efficient placement of text and visual materials on the same page. Our system is based on spatially distinct usage of marker parts which enables efficient collocation of other materials with the marker (Figure 4.1). Each bi-marker consists of two parts. Size of each of the parts is half of a square. These two parts should be placed on different parts of the printed material. Figure 4.1 shows the placement of markers on two pages of a book for both systems.

4.2 Detection Rate

Detection rate is the most important factor that determines the success of a tracking system. We analyzed the performance of our system and ARToolkit in detecting markers according to: (i) marker size, (ii) camera-marker angle, (iii) false positive marker detection, and (iv)



a) Our System

b) ARToolkit

Figure 4.1: Placement of markers for both systems

illumination of the environment. Marker size is defined as the number of pixels occupied by the marker in the screen. Camera-marker angle is the angle that between the marker plane and camera axis (Figure 4.2). False positive marker detection is incorrect detection of a marker when actually it does not exist. Illumination of the environment is specified into five different cathegories:very low, low, normal, high, and very high.

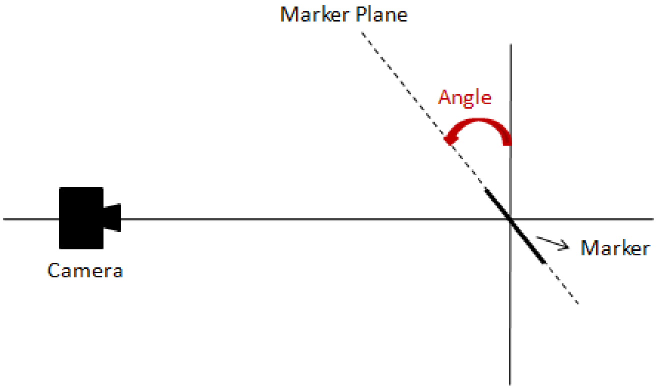


Figure 4.2: Camera-marker angle

For the analysis made for the effect of marker size, confidence factor at ARToolkit library was set to 0.8 and marker sizes from 10 to 100 pixels were tested. The marker size was increased by 10 pixels at each step. We gathered 30 results for each system at each step and their overall averages were considered. The summary of the experimental results is graphed in Figure 4.3. Our system has a better performance nearly for all marker sizes tested. According to the results, a marker size of 50 pixels or greater achieves 90 % success rate in our system. Whereas in the ARToolkit system, marker size of 65 pixels achieves 90 % success rate.

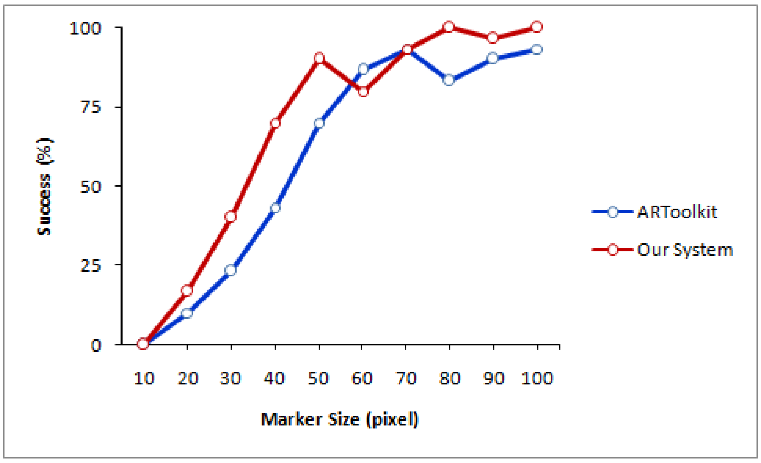


Figure 4.3: Marker size vs Success Rate

We have also analyzed the performance of the two systems for different camera- marker angles ranging from 0 to 90 degrees. Marker-camera angle is increased 15 degrees at each step. We gathered 30 results for each system at each step and their overall averages were considered. The summary of the experimental results is graphed in Figure 4.4. Our system has a better performance for all camera-marker angles tested.

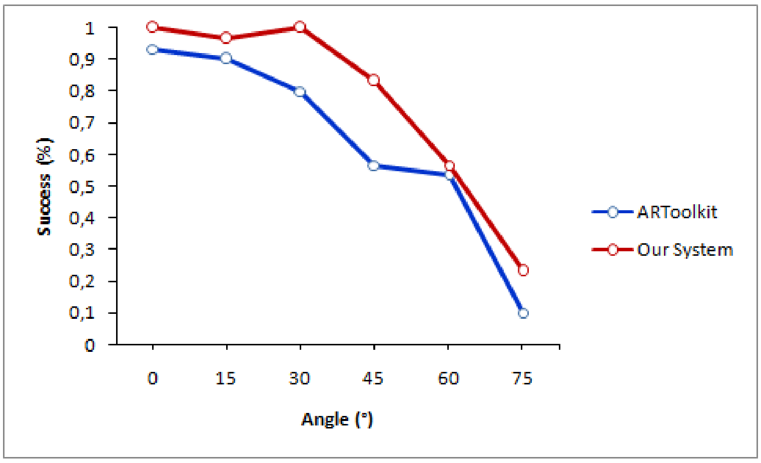


Figure 4.4: Camera-Marker angle vs Success Rate

False positive detection is one of the main problems in marker tracking systems. False positive detection is the incorrect detecting of a marker in a frame where the marker actually does not exist. In order to evaluate both systems for false positive detection, 10 different indoor and outdoor videos were recorded. The videos do not include any marker and both systems were tested for all videos. Results are shown in Table 4.1. For all videos, our system showed a better performance.

Table 4.1: Number of false positive detections for each system.

Video No.	Number of Frames	ARToolkit	Our System
1	227	4	1
2	249	2	0
3	236	9	2
4	188	3	1
5	162	4	0
6	258	1	1
7	216	7	2
8	182	5	2
9	191	2	1
10	214	6	1
Average	212.3	4.3	1.1

Illumination is one of the main problems in image processing methods using streams from the camera. Lighting in the environment has a significant effect on tracking. In order to compare the success of both systems in changing lighting conditions, we recorded a video in daylight illumination. Then, we changed the brightness of the video in order to increase and decrease the light in the images. Five steps are organized: very low, low, normal, high, and ver high. 30 frames are grabbed for each step. The summary of the experimental results is graphed in Figure 4.5. According to the results, changing lighting conditions has significant effect on the ARToolkit library. Our system was slightly effected in low and high lighting conditions and even achieved a good performance at very low and very high lighting conditions.

The difference in the performance of our system and ARToolkit can be attributable to three reasons: (i) different thresholding methods; (ii) different tracking methods; and (iii) different marker identification methods utilized by the two systems.

ARToolkit employs global thresholding in order to binarize the input frame. Since, threshold values are not determined dynamically by the ARToolkit system, the intensity of light in the environment becomes a critical factor for the tracking success. However, in our system, threshold values are determined dynamically and changes in the lighting of the environment

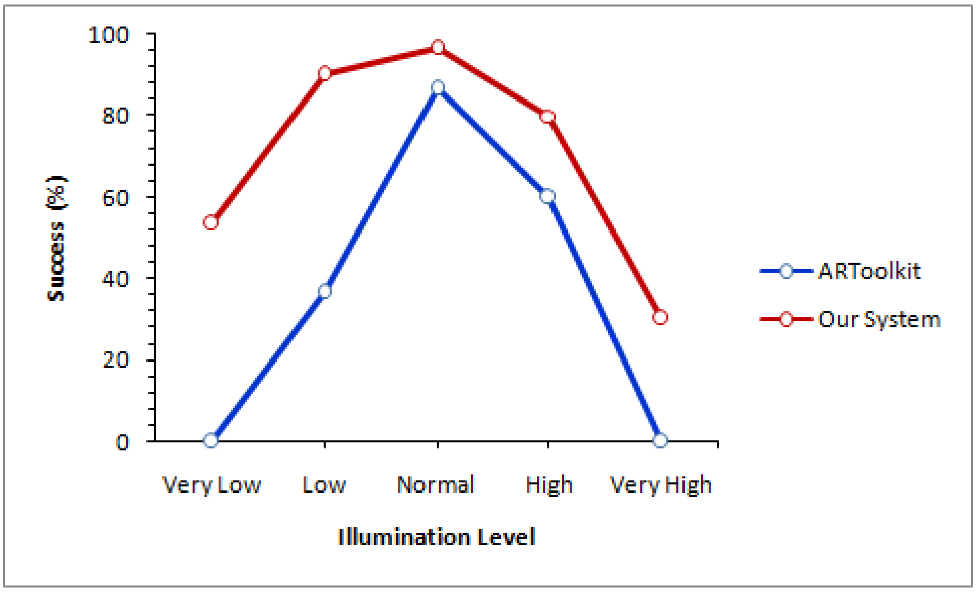


Figure 4.5: Illumination vs Succes Rate

has a lower effect on tracking success.

Secondly, unlike our system which uses the topology of the markers, ARToolkit uses the geometry instead and searches for quadrilateral shapes in the frame to detect a marker. However, tracking by using the geometry of the markers can easily lead to some short-comings, because even minor deformations in the geometry of the markers can prevent their detection. For example, when a marker slightly cambers, the shape of the marker is no longer quadrilateral and hence, it becomes undetectable by the ARToolkit system.

Finally, even though the quadrilateral shape is successfully detected, the patterns in the quadrilateral shape should match with the stored marker patterns in order to detect it as a marker. However, it may not be assigned as a marker due to some disadvantages of the normalization process. Normalization process transforms the patterns located on the detected quadrilateral shape into screen coordinates. Patterns may be corrupted and matching with stored patterns may fail. However, our system uses subgraph isomorphism algorithm in order to detect the markers. The subgraph isomorphism algorithm does not have such drawbacks of the pattern matching methods.

4.3 Library Size and Performance

Most of the time, more than a single marker is needed for AR applications. In many situations, the number of markers needed may be very high and this necessitates a large marker library size to meet the requirements. Theoretically, infinite number of markers should be created for the ARToolkit library since any rotationally asymmetric pattern inside a black square border can be used as a marker. Similarly, thousands of markers should be created for our system by increasing the variety of the topology trees. Although, library size of both systems is large enough to create complex applications, a large number of markers leads to a decrease in the performances. We analyzed the performance of both systems under varying number of visible markers. We created 100 patterns for the ARToolkit library using ARTOOLKIT PATTERN MAKER program[1]. We stored 10 different topology trees in our system which enable our system to detect more than 100 bi-markers. Figure 4.6 shows the results. Whereas the ARToolkit library is highly affected by the increase in the number of visible markers, our system is only slightly affected.

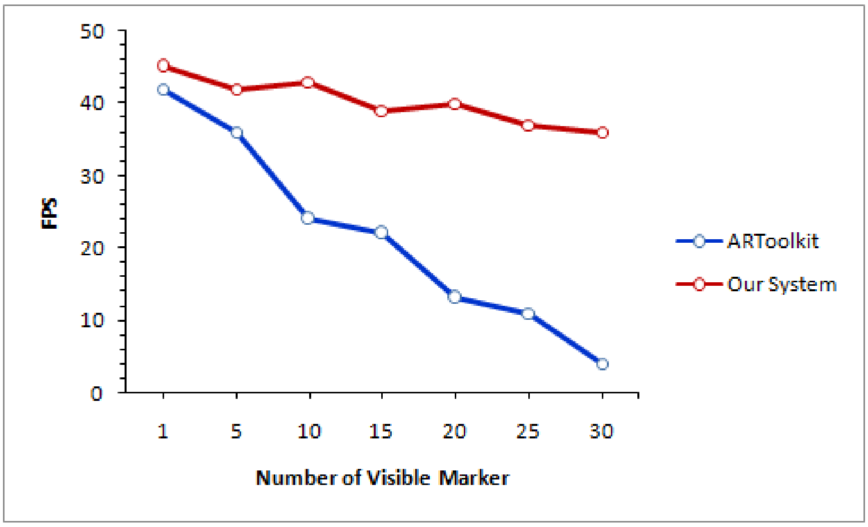


Figure 4.6: Number of visible marker vs FPS

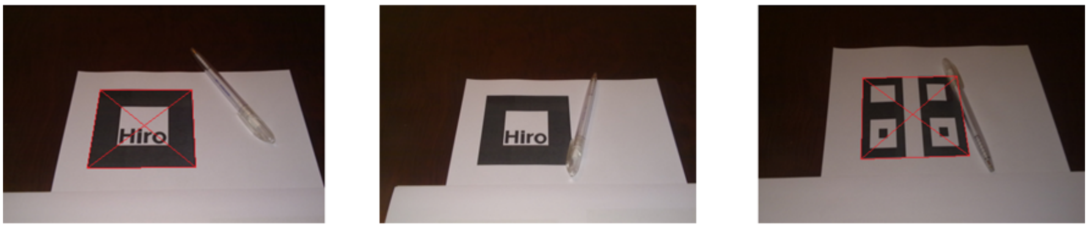
The sharp decrease in the performance of the ARToolkit library with increasing number of visible markers is due to the pattern matching method used in marker detection. In our system, we are using a simplified subgraph isomorphism algorithm with linear time complexity which

[1] www.cs.utah.edu/gdc/projects/augmentedreality/

outperforms the pattern matching method.

4.4 Occlusion

Users may occlude some parts of the markers while using them and obscure the visibility of the patterns needed for the detection of the markers. This generally leads to unsuccessful tracking of the markers. Marker systems should realize these occlusions and permit them to some extend. The marker tracking method used by the ARToolkit library is not capable of detecting markers occluded at any extent by the users. Our system tolerates occlusion unless the topology of the marker is corrupted. Figure 4.7 shows an illustration of the effect of occlusion on both systems. As shown in the figure, ARToolkit system generally fails even if a tiny area on the marker is occluded. However, occlusions to the border of the markers are tolerated in our system.



a) ARToolkit marker detected

b) ARToolkit marker occluded

c) Our marker occluded and detected

Figure 4.7: Occlusion handling on both systems

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

Augmented Reality is a developing and popular research area in Computer Science. Since the early 1990s when research in Augmented Reality was first began, a number of different algorithms and methods have been proposed for tracking and visualizing. Also, many applications in various areas such as advertising, architecture, entertainment, education and industry based on Augmented Reality were developed. Tracking systems detect objects in the real world and extract their 3D position and orientation. Hence, tracking systems constitute the core part of Augmented Reality. Among many different methods developed to track real objects, marker based tracking systems became prominent for their ease of use and low cost.

Marker based vision tracking systems remain an interesting area in Augmented Reality. In this thesis, we developed a marker based tracking system using topological properties of markers. Markers are detected and identified using their topology trees. Topology trees of markers are region adjacency graphs that are obtained from binary images of markers. Similarly, camera frames obtained are also converted into binary images and corresponding topology trees are created. We used left heavy depth sequences as the canonical convention for topology trees. We track markers using a simplified version of subgraph isomorphism algorithm that searches marker topology trees in a frame topology tree.

Our proposed marker system utilizes bi-markers that consist of two markers. Markers are relatively small shapes that are easy to recognize. Our detection algorithm tracks the markers and pairs them according to their 2D positions and orientations. This feature enables the usage of spatially distinct marker pairs. Hence, the two parts of a bi-marker can be placed at different places on a surface.

We compared our proposed system with the ARToolkit library. Spatially distinct usage of marker parts, occlusion handling, detection rate, marker library size and performance were used as testing parameters. Our system surpasses the ARToolkit library by enabling placement of marker parts on different surfaces and permitting occlusion as long as the topology of the marker is not corrupted. In addition, our system has a better detection rate than the ARToolkit library according to marker size, camera-marker angle and false positive marker detection. Marker library size of both systems is large enough to create complex applications, but the performance of the ARToolkit library decreases rapidly with increasing number of visible markers in the scene. In our system, increase in the number of visible markers slightly decreases the performance.

5.2 Future Work

In our system, the detection of bi-markers is not dependent on their geometry. Hence, bi-markers with any geometry can be used during the detection process. However, for the extraction of 3D transformation matrices, we are using specifically the corners of the bi-markers and this necessitates detected markers to be in quadrilateral shapes. To be able to use bi-markers that are not only quadrilateral but also in other geometrical shapes, another method for the extraction of 3D transformation matrices that does not use the corners of the bi-markers can be developed. By this way, detection algorithm and extraction of 3D transformation matrix processes will both not require markers with certain geometry. For example, bi-markers can be embedded into a picture or text which leads to the creation of natural markers and prevents the loss of visual integrity.

Creating different markers is also an important problem in marker systems. A marker generation tool that can create different effective markers can be created.

In addition, the interaction of virtual and real objects is a key point in Augmented Reality. Our system provides 3D movement of virtual objects with respect to the movement of the markers. Additional interaction methods can be added to our system by improving the existing marker system on the marker design and tracking. Simple hand interactions such as selective touching among multiple items can be embedded into the system. In order to achieve this, markers that consist of more than two parts can be designed. Tracking algorithm should track the whole marker and detect those parts that are selected by occluding. Each selected part or combination of parts can correspond to a different option and by this way, a single marker can possess multiple options.

Finally, our system enables occlusion only on the border of the markers and occlusions that corrupt the topology of the markers are not handled. An important extension to our system can be adding a better occlusion handling method that tolerates a larger occluded area.

REFERENCES

[1] F. Ababsa and M. Mallem. A robust circular fiducial detection technique and real-time 3d camera tracking. *Journal Of Multimedia*, 3, No:4:34–41, 2008.

[2] F. Abad, E. Camahort, and R. Vivo. Camera calibration using two concentric circles. *Proc. of Inter. Conf. on Image Analysis and Recognition (ICIAR 2004)*, pages 688–696, 2004.

[3] R. Azuma. Tracking requirements for augmented reality. *Communications of the ACM*, 36, Issue 7:50–51, 1993.

[4] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, Vol. 21, No. 6. (2001):34 – 47, 2001.

[5] R. Bencina and M. Kaltenbrunner. The design and evolution of fiducials for the reactivision system. *In Proceedings of the 3rd International Conference on Generative Systems in the Electronic Art*, 3rd iteration 2005, 2005.

[6] J. Bernsen. Dynamic thresholding of grey-level images. *Proc. of the 8th Int. Conf. on Pattern Recognition (ICPR)*, pages 1251–1255, 1986.

[7] O. Bimber, F. Coriand, A. Kleppe, E. Bruns, S. Zollmann, and T. Langlotz. Superimposing pictorial artwork with projected imagery. *IEEE MultiMedia*, Volume 12 , Issue 1 (January 2005):16 – 26, 2005.

[8] O. Bimber and R. Raskar. Modern approaches to augmented reality. *International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2005 Courses*, Session: Spatial augmented reality: a modern approach to augmented reality, 2005.

[9] Y. Cho, J. Lee, and U. Neumann. Multi-ring color fiducial systems for scalable fiducial tracking augmented reality. *Proc. of IEEE VRAIS*, page 212, 1998.

[10] D. Claus and A. W. Fitzgibbon. Reliable fiducial detection in natural scenes. *Proceedings of the 8th European Conference on Computer Vision (ECCV '04)*, 4:469–480, 2004.

[11] F. Ababsa and M. Mallem. Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems. *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, pages 431 – 435, 2004.

[12] S. Eitoku, T. Tanikawa, and Y. Suzuki. Display composed of water drops for filling space with materialized virtual three-dimensional objects. *Proceedings of the IEEE conference on Virtual Reality*, pages 159 – 166, 2006.

[13] M. Fiala. Artag, a fiducial marker system using digital techniques. *Proc. of IEEE Int. Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 2:590–597, 2005.

[14] D. Hildebrand. Real time tomographic reflection: Phantoms for calibration and biopsy. *In proceedings of IEEE/ACM International Symposium on Augmented Reality (ISMAR'01)*, pages 11–19, 2001.

[15] S. Nakano and T. Uno. Efficient generation of rooted trees. *National Institute of Informatics*, 2003-005E:3025–3030, 2003.

[16] K. Jo, Y. Kakehi, K. Minamizawa, H. Nii, N. Kawakami, and S. Tachi. Arforce: A marker-based augmented reality system for force distribution input. *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, 2008.

[17] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. *Proc. of ACM/IEEE Int. Workshop on Augmented Reality (IWAR 1999)*, pages 85–92, 1999.

[18] L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR '02)*, pages 27–36, 2002.

[19] C. B. Owen, F. Xiao, and P. Middlin. What is the best fiducial? *The First IEEE International Augmented Reality Toolkit Workshop*, pages 98–105, 2002.

[20] J. Rekimoto and Y. Ayatsuka. Cybercode: designing augmented reality environments with visual tags. *Proc. of Inter. Conf. on Designing augmented reality environments, (Dare 2000)*, pages 1–10, 2000.

[21] A. Rosenfeld. Adjacency in digital pictures. *Information and Control*, No.26:24–33, 1974.

[22] T. Wang, Y. Liu, and Y. Wang. Infrared marker based augmented reality system for equipment maintenance. *2008 International Conference on Computer Science and Software Engineering*, 5:816 – 819, 2008.

[23] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, and D. Colucci. High-performance wide-area optical tracking: The hiball tracking system. *Presence: Teleoperators and Virtual Environments*, Volume 10 , Issue 1 (February 2001):1–21, 2001.

APPENDIX A

BI-MARKERS

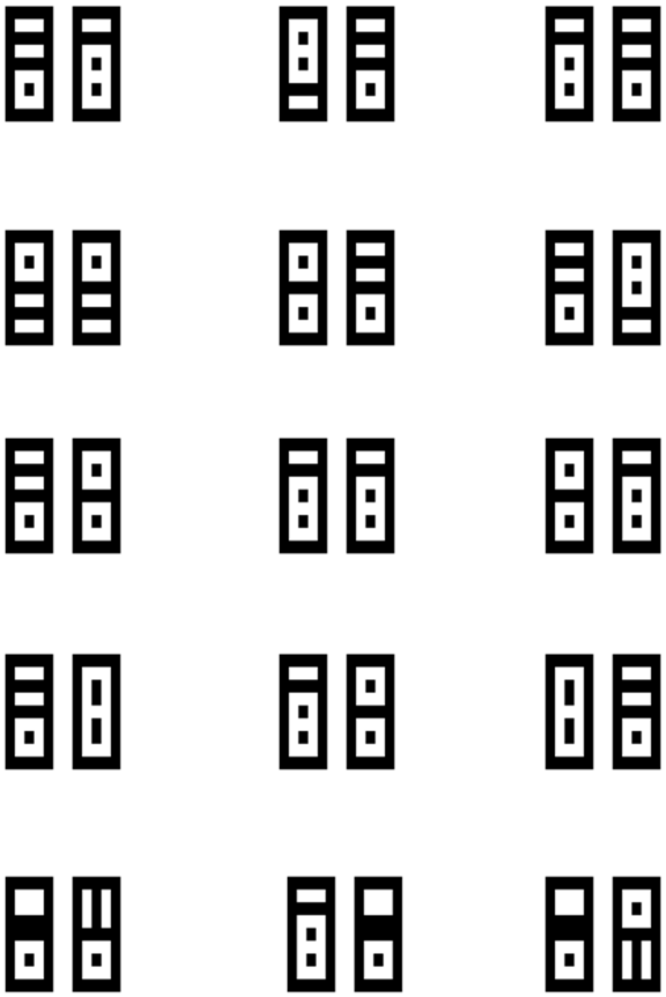


Figure A.1: Bi-markers