A STEREO HMD SYSTEM WITH VISUAL AND INERTIAL DATA CAPTURE FOR 3D AUGMENTED REALITY APPLICATIONS

A Thesis

by

Ahmet Kermen

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Electrical and Electronics Engineering

Özyeğin University
January 2016

A STEREO HMD SYSTEM WITH VISUAL AND INERTIAL DATA CAPTURE FOR 3D AUGMENTED REALITY APPLICATIONS

Approved by:

Prof. A. Tanju Erdem, Advisor
Department of Electrical and Electronics Engineering
Özyeğin University

Assist. Prof. Dr. Ali Özer Ercan
Department of Electrical and Electronics Engineering
Özyeğin University

Assoc. Prof. Dr. Selim Balcısoy
Faculty of Engineering and Natural Sciences
Sabancı University

Date Approved: 8 January 2016

To my mother

ABSTRACT

Augmented Reality (AR) technology has been considered to have a great potential in many different areas such as medical visualization, manufacturing, entertainment and cultural heritage for enhancing human-computer interaction (HCI) and improving user experience. It has been receiving growing interest by several major companies including Apple [1], Google [2], Microsoft [3] and Sony [4].

The main objective of augmented reality applications is to give realistic feeling of immersion with overlaying computer generated virtual scene images over user's view through specialized displays. Seamless blending of virtual scene elements on top of real world elements is one of the main requirements for immersive AR applications and it requires a high-accuracy 3D tracking.

This thesis deals with the design and development of an augmented reality (AR) system that is made up of a head-mounted display (HMD) unit, visual and inertial sensors. A camera pair is employed as the visual sensor, and the inertial sensor (IMU) unit houses an accelerometer and a gyroscope. The system tries to improve the accuracy of tracking by combining measurement data from visual sensors (cameras) with inertial sensors (accelerometer, and gyroscope). The main objective of this work is the development of a stereo HMD system with visual and inertial data capture for 3D augmented reality applications.

The work reported in this thesis was supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under the project numbered EEEAG-110E053, which is developed in Özyeğin University.

Keywords: Augmented Reality, Head-Mounted Display, Inertial Sensors, Calibration, Tracking, Visual and Inertial Sensor Fusion

ÖZETÇE

Eklenmiş gerçeklik teknolojisi (AR), sağlık ve endüstriyel eğitim, eğlence ve kültürel miras gibi farklı alanlarda kullanıcı deneyimi ve bilgisayar-insan etkileşimini artırmada büyük bir potansiyele sahip olarak görülmektedir ve Apple [1], Google [2], Microsoft [3] ve Sony [4] gibi büyük firmalarin ilgisini artarak çekmektedir.

Eklenmiş gerçeklik uygulamalarının ana amacı, kullanıcıya özel ekranlar aracılığı ile kendi gördüğü görüntü üzerine bilgisayar ile oluşturulmuş görüntüleri birleştirerek yüksek gerçeklik hissini vermektir. Sanal nesneler ile gerçek dünya nesnelerinin birbirlerine kusursuz birleşimi eklenmiş gerçeklik uygulamalarının ana gereksinimlerinden biridir ve yüksek hassasiyette 3 boyutlu takip sistemini ihtiyaç duyar.

Bu tez çalışması, eylemsizlik (IMU) ve görsel algılayıcılardan oluşan başa takılan bir ekran (HMD) tabanlı eklenmiş gerçeklik sisteminin tasarım ve geliştirmesini ele almaktadır. Görsel algılayıcı bir kamera çifti ve eylemsizlik algılayıcılar (IMU) ise sırasıyla ivmeölçer ve açısal hız ölçerden oluşmaktadır. Sistem, görsel ve eylemsizlik algılayıcılarından aldığı verileri birleştirerek 3 boyut takip hassasiyetini arttırmaya çalışmaktadır. Çalışmanın ana amacı, görsel ve eylemsizlik algılayıcılarından veri toplayacak, stereo başa takılan ekran tabanlı 3 boyutlu eklenmiş gerçeklik sistemi geliştirmektir.

Bu tezde raporlanan çalışma, Özyeğin Üniversitesi'nde geliştirilen ve Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) tarafından desteklenen EEEAG-110E053 numaralı proje dahilinda yapılmıştır.

Anahtar Kelimeler: Eklenmiş Gerçeklik, Başa Takılan Ekran, Eylemsizlik Algılayıcıları, Kalibrasyon, Hareket Takibi, Eylemsizlik Verileri ve Kamera Görüntülerinin Beraber Kullanımı

ACKNOWLEDGEMENTS

Foremost I wish to thank to my supervisor, Prof. Arif Tanju Erdem, who has supported me throughout my thesis work with his patience and knowledge. Without him this thesis, would not have been written. Besides my advisor, I would like to thank the rest of my thesis committee members for their time.

My sincere thanks goes to Dr. Mehmet Kemal Özkan for his continuous encouragement me to start writing this thesis report. I would also like to thank to all my team members, starting with Assist. Prof. Tarkan Aydın. I am thankful and indebted to him for his invaluable assistance. Cengiz Hüroğlu for his patience and continuous support.

I would like to give a big thanks to my family for supporting me through my entire life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

This chapter provides an introduction to the head-mounted display (HMD) based augmented reality (AR) systems. It also briefly describes the role of inertial and visual sensors, calibration, tracking, and sensor fusion in developing a system which gives a realistic feeling of immersion to users.

1.1 Motivation

Augmented reality (AR) blends real world and computer-generated digital data including but not limited to, graphics, sound and tactile. Today, majority of AR applications use live video data to augment (supplement) real-world environment elements. The definition of AR from Encyclopedia Britannica [35] is the following: "Augmented reality, in computer programming, a process of combining or 'augmenting' video or photographic displays by overlaying the images with useful computer-generated data."

The majority of today's augmented reality applications concentrate on visual augmented reality and the goal is to improve the user's perception of the real world by blending virtual graphics that is rendered by a computer over the real-world image the user views through specialized displays. AR is different from virtual reality (VR). VR gives complete immersion to a user in a virtual environment with no real world elements displayed.

The beginnings of AR date back to Ivan Sutherland's short article entitled "The Ultimate Display" [36] in 1965. In 1968, Sutherland created the first computer driven head-mounted display (HMD) to effectively display three-dimensional images based on the same principles as we know today [10]. His device displayed perspective vector graphics that changed as the user moved.

The term augmented reality (AR) is coined by Tom Caudell in 1992. He used the "augmented reality" term for a head-mounted display system that guided technicians in electrical wires assembly [11].

Azuma provides a commonly accepted definition of AR with his survey on augmented reality in 1997 [37] and due to the fast development in AR field with a new survey in 2001 [38], as a technology identified by three characteristics:

- blends virtual and real objects in a real environment
- interactive and runs in real-time
- aligns virtual objects with real ones

A basic augmented reality system consists of a display, a camera and a computing unit. The camera captures an image of the real environment, and then the AR system augments virtual objects on top of the real image and displays the result. The system captures an image of the real environment, finds markers and calculates the location and orientation of the system, and then augments virtual objects on top of the real world image and displays it on the screen [37]. The most common type of displays used in an augmented reality system is a head-mounted display (HMD) that let users see real world through optically or video-mixed.

When Sutherland wrote about the ultimate display [36], computer displays had limited technical capabilities and they were able to draw only primitive elements. Today display and rendering technology is moving toward photorealism, enabling seamless integration of virtual objects on top of real world objects. With photorealistic real-time display systems, many different domains benefit from the use of AR technology such as medical training and practice, education, manufacturing, arts and entertainment and military. Hence, there is renowned interest in developing high fidelity HMD systems.

1.2 The Objective of the Thesis

The goal of this thesis work is to design and develop a multi sensor stereo head-mounted display (HMD) system for augmented reality (AR) applications. The system contains a camera pair as both visual sensor and real video feed, accelerometer and gyroscope as inertial sensors (IMU), a head-mounted display (HMD) as viewing screen, a personal computer where, a rendering engine which generates virtual objects and main processing software which orchestrates all mentioned hardware and software modules, run. The main benefit is to provide a platform for researchers working on augmented reality (AR), 3D tracking, sensor fusion, occlusion, calibration, etc. topics.

The type of the display used in augmented reality systems is generally an optical see-through or video coupling HMD. In optical see-through HMDs, there are optical instrument such as transparent mirrors functioning as displays and lens systems in front of the user's eyes. These partially transparent optical mechanisms allow users to see the real world and also the image of the virtual scene projected onto the transparent display by looking directly through them.

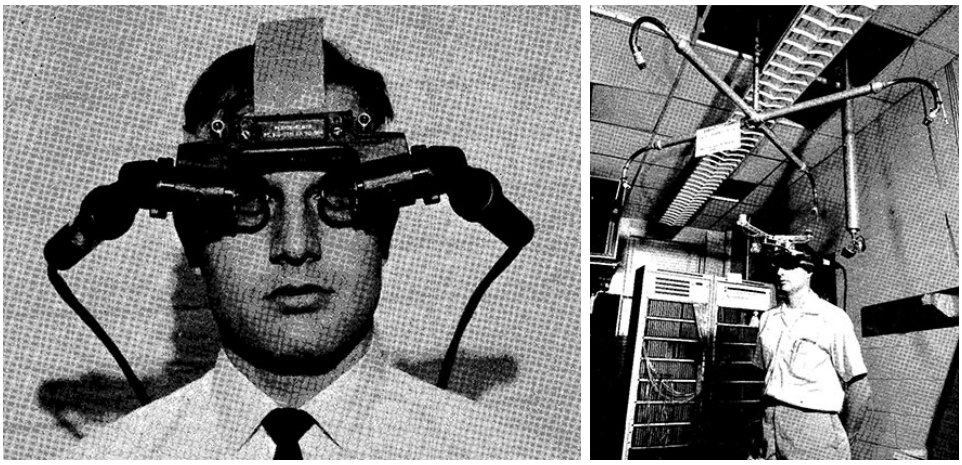
However, video see-through HMDs fuse virtual graphics with images of real-world scene captured by single or pair of cameras mounted to the user's head. An opaque display is fitted in front of the user's eyes, where the user experiences the augmented scene.

Optical see-through HMDs allow users to see the real world directly. This feature can be considered as an advantage, but also a drawback, because it means there might be a considerable time lag between the real-world view and the virtual image [37]. Additionally, the optical instruments of an optical see-through HMDs usually reduce the amount of light coming from the real world. This has a negative effect on what users see from the real world, limiting the application of this type of HMD to different scenarios [39].

The lag between the real-world and the virtual scene for a video see-trough type HMD can easily be eliminated by applying a predefined delay during the presentation of the augmented image. In addition, brightness matching techniques can be used to eliminate any illumination difference between the virtual and real scene images [40]. These are the advantages why a video see-through type HMD was used in this thesis work.

1.3 Literature review

In 1960s Ivan Sutherland created the first model of the head-mounted display (HMD) and called "The Sword of Damocles" because the weight of the mechanical tracker required that it be mounted to the laboratory ceiling. Its graphical capabilities were limited and provided only primitive wire-frame graphics that animated as the user moved [10]. His display and sensor setups can be seen in Figure 1. Since Sutherland's pioneering work, the main components to create an AR system have remained the same: display, tracking module, and computing unit with rendering and controlling software. The basic overview of an AR system can be seen in Figure 2.



(a) HMD with miniature CRTs (b) Mechanical position sensor

Figure 1: Sutherland's HMD setup showing displays and mechanical sensors [10].

There are two popular display categories for an AR system, head-mounted displays (HMDs) and monitor-based displays. Under the first category, optical see-through

and video see-through type HMDs are the ones that can be seen as directly related to augmented reality applications.

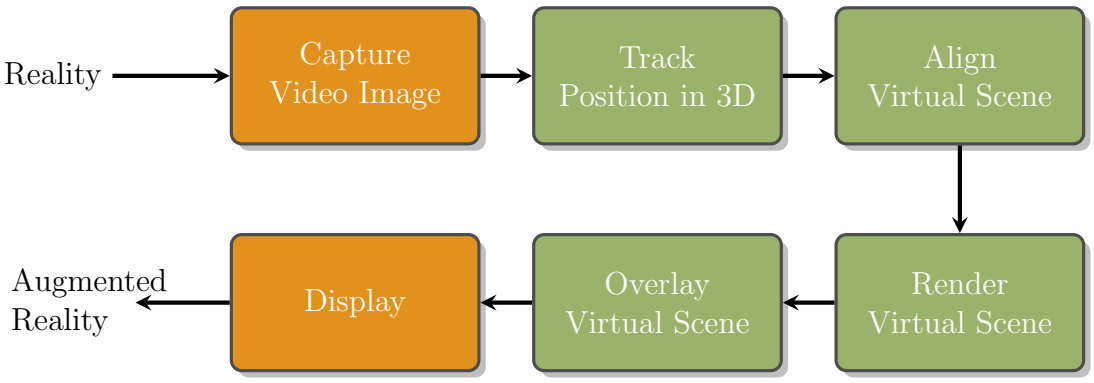


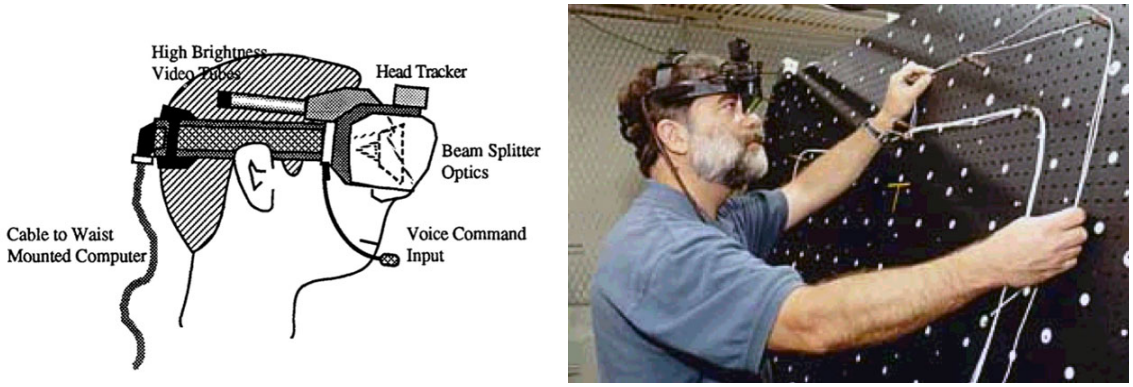
Figure 2: Overview of a simple AR system.

In optical see-through setup, virtual scene is projected on a semitransparent display in front of user's eye. Since this display is partially transmissive user is able to view the real world and the virtual scene blended into the real one. Head-Up Display or "HUD" name is sometimes used to refer to optical see-through HMDs, because HUD systems used in military aircrafts take very similar approach [37].

The optical see-through HMDs have an advantage that users can see the real world directly without requiring any display equipment. Their performance highly depends on the synchronization of real and virtual world which should have a time lag not noticeable by the user. This type of displays also usually limit the amount of light user sees from the real world which may make it impractical for some applications. AR is often associated with adding objects to real environments but it can also be used to remove objects from real environments. Since optical see-through display setups have no control on what user sees from the real world, this feature does seem to be out of scope for this type of HMD.

Tom Caudell's head-mounted system at aircraft manufacturer Boeing is one of the early optical see-through head-mounted (HUD) systems for augmented reality, which displays plane specific electrical wiring schematics in order to guide workers

in assembling large bundles of electrical wires. His HMD setup diagram and it's application can be seen in Figure 3.



(a) Components of HMD (b) HMD in use

Figure 3: Caudell's HMD setup diagram and while it's in use [11].

Google Glass is an example of optical see-through type HMD which brings smartphone like functionality into miniature mobile (cordless and battery powered) form. It can also be mounted to prescription lenses as seen in Figure 4.

Figure 4: A Google Glass for prescription lens [12].

Video see-through HMD systems have one or two head-mounted cameras to capture real world image. This type of HMD systems block user's view, allowing users to see only computer generated graphics through opaque display fitted in front of their eyes. Users do not see the real world, but instead see augmented real-world to increase the sense of immersion. A video see-through HMD system equipped with a stereo camera pair lets users directly perceive the depth of the scene. A full stereoscopic video see-through HMD can be seen in Figure 5.

(a) Front view (b) Side view

Figure 5: A video see-through HMD by Trivisio Prototyping GmbH. [13].

The Rift from Oculus VR[1] is a virtual reality head-mounted display, and it is one of the first consumer-targeted virtual reality headsets [41]. Although its main goal is virtual reality applications and does not support augmented reality out of the box, it can easily be extended by integrating video cameras to be used in augmented reality applications [42, 15, 43]. Such a modification can be seen in Figure 6.

(a) Original Rift (b) Modified Rift for AR applications

Figure 6: Oculus Rift by Oculus VR, LLC [14] and Ovrvision [15].

The HMZ series [4] and PlayStation VR (Project Morpheus)[2] [30] from Sony, Cinemizer [28] from Carl Zeiss, HTC Vive (SteamVR)[3] from HTC partnership with Valve [31] and other commercially available or to be available head-mounted display products can be used for different AR applications. The Google Cardboard [33] virtual

[1]At the time of writing, only prototype version available for purchase.
[2]At the time of writing, it is currently a prototype and has no confirmed release date.
[3]At the time of writing, only available to the selected developers.

reality (VR) platform which is a fold-out cardboard mount for utilizing smartphones, Oculus Gear VR [44] to be used with Samsung GALAXY smartphones and the ZEISS VR ONE GX [32] which is fully compatible with the Google Cardboard platform, can be listed as affordable and simple consumer-level alternatives to all-in-one HMD systems.

There are two display types for the monitor-based display category for an AR system: hand-held and spatial displays. Both offer a cost-effective way to introduce AR applications. Spatial displays are mostly positioned fixed in an environment. They include screen-based video see-through, and projective displays. These interfaces are usually traditional desktop computer screens and televisions. LEGO Digital Box [16] is one example AR application which uses monitor-based display. It gives the opportunity to see many LEGO sets in 3D on the screen. Figure 7 shows how it displays an animation of the completed kit over the live video when the user holds a LEGO box up to the camera.

(a) Showing to camera (b) Viewing AR on screen

Figure 7: LEGO digital box by LEGO group and Metaio GmbH. [16].

The increase in processing power and availability of built-in high resolution cameras has transformed handheld devices, specifically smartphones and tablets, to handheld multimedia gadgets ideal for AR applications. In the recent years, the ease of access to smartphones and tablets has made different AR applications popular in many areas. One of the popular examples is a mobile application from Wikitude [17] which displays geographically-relevant information on the smartphone or tablets

screen using the built-in camera in a given area. Useful information is often presented in the form of online articles about a landmark, or directions to a point of interest. This and marker based marketing AR sample applications are shown in Figure 8.

(a) Location based AR (b) Marker based AR

Figure 8: Wikitude's two AR applications by Wikitude GmbH. [17].

Real-time tracking is one of the indispensable parts of an AR system. Tracking is followed by registration where, the data from tracking is used to blend the virtual scene with real world. Tracking techniques for AR can be categorized into three types, which are sensor-based, vision-based and hybrid [45]. The application domain and requirements determine the tracking technique to be used.

Physical sensors such as magnetic, acoustic, inertial, optical, GPS and mechanical sensors are used in sensor-based tracking techniques. Some type of sensors detect signals from their counterpart transmitters fixed globally such as magnetic and acoustic. Inertial sensors are placed on the user controlled AR component such as HMD to calculate the user's viewing position in 3D. They all have their trade offs. For example, magnetic and inertial sensors have a high update rate and they can be very small and light, but magnetic sensors are affected by nearby electromagnetic fields or materials and inertial sensors suffer from noise for slow motion which results in error buildup. PlayStation VR from Sony tracks user's head movement with accelerometer and gyroscope sensors run at a frequency of 1000Hz [30]. Newman et al. [46] developed an AR system that employs ultrasonic sensors for tracking in large indoor areas.

Vision-based tracking techniques use computer vision and image processing algorithms to estimate the camera position relative to real world objects by finding correspondence between 2D image features and their 3D world coordinates. This type of tracking techniques are convenient for an AR application that already employs a camera as part of the system, using the camera as both image capture and sensor device. The goal is to find artificially placed features (markers) or naturally occurring features that are part of the actual environment, and use them to align virtual objects. Natural features are used for tracking depending on application requirements or cases where placing artificial cues on the scene is not acceptable. These artificial markers can be unique patterns which image processing software is able to recognize or infrared LEDs tracked by infrared sensitive video cameras. Vision-based tracking performance may degrade due to low frame-rate of the camera which outputs blurred image under fast motion for cases where there's not enough natural features that can be detected by the AR system. LEGO Digital Box [16] AR application is an example for vision-based tracking technique.

For some AR applications vision-based or sensor-based methods alone cannot provide robust tracking accuracy. For these cases, hybrid methods have been developed which combine information from different type of sensors [47]. For example The Rift from Oculus VR combines information from inertial sensors (gyroscope and accelerometers) and external infrared tracking sensor which track precisely positioned infrared LEDs on the device to improve tracking accuracy [41].

Augmented reality technology has several application areas including but not limited to, medical, manufacturing, entertainment, visualization, education, advertising and cultural heritage.

In the field of medicine, AR is used for medical surgical aid [48], training [49] purposes. It has been even considered as a treatment tool to help people overcome phobias [18] as one sample research can be seen in Figure 9.

(a) Only the therapist's hands. (b) With the participant

Figure 9: AR is used for the treatment of cockroach phobia [18].

Assembly tasks in manufacturing can be recognized as one of the most promising application areas of AR which started with the work by Caudell [11]. AR applications can show the step-by-step instructions at each stage on a head-mounted display or on a desktop screen. The interface of such as system can be seen in Figure 10. The work of Reitmayr et al. [22] tries to provide evidence to support the claim that AR systems improve assembly task performance.

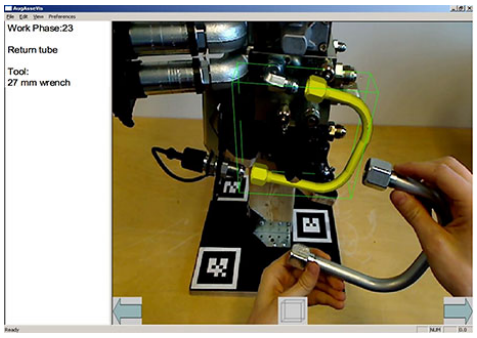


Figure 10: Augmented reality assembly instructions [19].

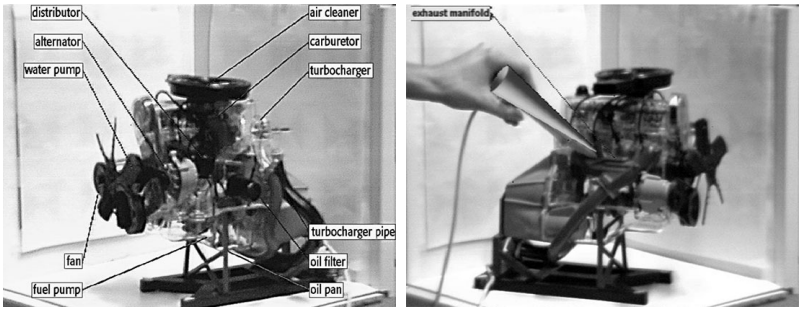
Augmented reality enables interactive games. The first outdoor mobile augmented reality video game ARQuake [20] was demonstrated by Bruce Thomas and his team at the University of South Australia. All the game characters are placed on top of a real environment that user could actually walk. Their setup and interface can be seen in Figure 11. Cheok et al. [50] improved this game concept by adding multi-player

capability with collaboration and large area interaction. AR Defender [51] game is one other example from many different AR games designed for mobile platforms, where the game playground is virtually rendered (augmented) on a paper where specifically designed marker which is used for tracking, is printed on.

(a) The setup (b) Augmented image

Figure 11: ARQuake, the first AR outdoor computer game [20].

AR is well suited for visualization both indoors and outdoors where it annotates objects and environments with useful information. The work of Rose et al. [21] matches model of a real-world object to the real object and visually annotates the real components with information from the corresponding model, as can be seen in Figure 12. The mobile application Anatomy 4D [52] gives interactive visualized information about human anatomy using augmented reality technology.

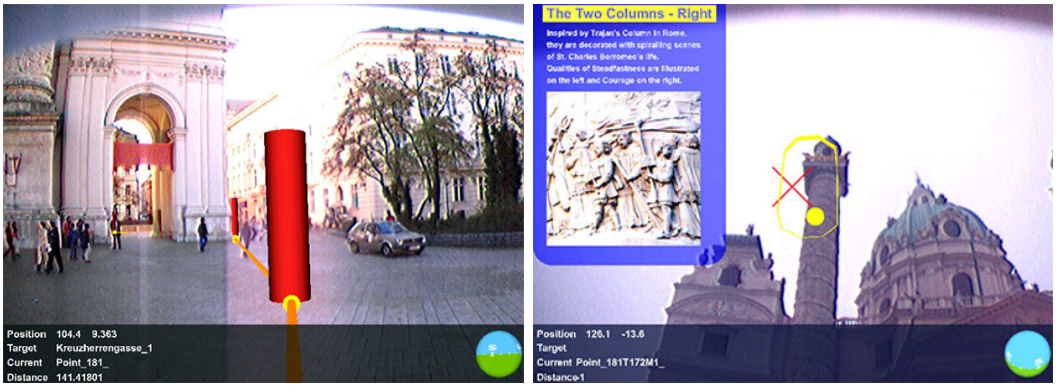


(a) AR view of an engine (b) User examines an engine part

Figure 12: Annotating real-world objects using augmented reality[21].

For outdoor environments several location-based information services similar to

Wikitude [17] use augmented reality browsers. AR applications can also provide collaborative navigation while giving information about users' surroundings [22]. The interface of an example collaborative AR system is shown in Figure 13. SkyORB from Realtech VR [53] is a mobile AR application for observing the stars planets and other objects in space using information from mobile device's compass and gyroscope.

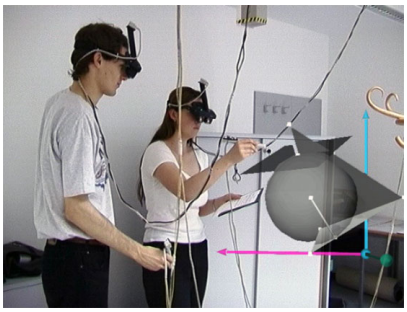


(a) Navigation (b) Annotation

Figure 13: AR application provides navigation and shows annotation to user [22].

There are many research studies on augmented reality (AR) tools that support education with 3D objects [54, 55] and some investigate the effects of AR on learning [56]. Kaufmann [23], Kaufmann et al. [57] developed a 3D construction tool called Construct3D which is specifically designed for improving learning in mathematics and geometry. Construct3D's sample augmented image for vector algebra can be seen in Figure 14(a). Figure 14(b) shows Sprout from HP [24], a personal computer with AR capabilities. Sprout has a 3D scanner and second projected screen where users can blend real and virtual objects.

Augmented reality enables new forms of advertising by connecting 3D graphics and videos with printed publications, journals, newspapers, billboards and product catalogues. One example is advertising campaign by MINI a printed marker in magazines that displays a 3D model as shown in Figure 15(a), of the MINI cooper on a computer screen when user holds the marker to a PC webcam [25]. Figure 15(b)



(a) Vector algebra

(b) Sprout

Figure 14: Construct 3D [23] and HP Sprout [24].

shows Disney's outdoor augmented reality application where users can capture an image interactively with Disney characters.

(a) Mini magazine advertising

(b) Disney outdoor interactivity

Figure 15: Mini [25] and Disney [26] augmented reality applications.

Vlahakis et al. [27] present an augmented reality project that reconstructs a cultural heritage site. Visitors can use the system to view and learn ancient architecture and historical information as shown in Figure 16. Similar applications have been developed for different sites around the world [58].

At the time of the development of this thesis work, Sony HMZ-T1, Carl Zeiss Cinemizer and Vuzix Wrap 1200VR were three HMD products that are relatively inexpensive and easily available on the market. Cinemizer lacks decent display resolution and supports an uncommon resolution of 870x500 pixels. Wrap 1200VR has no HDMI connectivity option which is very important for a modular and future-proof

(a) Present state (b) Augmented video

Figure 16: The ArcheoGuide [27] AR application images for a historical site.

system. Sony HMZ-T1 is the only option from these three that supports HDMI connection and a common high-res display resolution at the same time. This is the main reason why the work in thesis employs a Sony HMZ-T1 as part of the AR system.

1.4 Contribution of the thesis

The main contribution of this thesis work is to provide head-mounted display (HMD) augmented reality (AR) system with all hardware and software components fully integrated as a platform to be used for AR research applications. Visual and inertial data capturing, camera image rectification, synchronization between cameras and inertial sensors, rendering 3D virtual scene objects, stereo image display for 3D viewing, etc. all are handled by the system out of the box, allowing researchers to spend more time on their AR applications research topic rather than dealing with such end-to-end system integration tasks. For example, solving camera and inertial sensor synchronization alone may require working with electronic components and even firmware modifications for sensor controller chips. This and providing a reusable and a modular system for AR research projects are the main goals of this thesis. Development of this system is part of a bigger research project where it was also used for different augmented reality works [47, 40, 59].

1.5 Outline of the thesis

This thesis is structured as follows:

Chapter 1 gives general introduction by taking a brief look at augmented reality systems and applications and also sets the objective of the thesis.

Chapter 2 contains a detailed information about general augmented reality system components and the technical details of the one used in this thesis work.

Chapter 3 starts with overview of the head-mounted system proposed in this thesis work and presents all the details of the HMD system, explaining function of each component, giving details about every task processed by the software part of the system.

Chapter 4 is to serve as a review of the outputs of the work, including calibration and animation results.

Chapter 5 contains a summary of the study, contribution to the literature and suggestions for future work.

CHAPTER II

BACKGROUND

This chapter provides an overview of the head-mounted display (HMD) products, 3D animation rendering engines and Extended Kalman Filter (EKF) based tracking. The specification and description of the components of the system reported in this thesis work are covered in detail, including the cameras, the inertial sensors and the 3D rendering engine.

2.1 Overview of the HMD products in the market

Every year developing an augmented reality (AR) system into commercial product is becoming more practical and affordable because performance of all the components has improved while their cost decreased. The AR technology and it's applications are not limited to the expensive academic, industrial or military projects anymore. Today, high quality head-mounted display (HMD) products are available from different technology companies at an affordable cost in the consumer-level market.

At the time of the development of this thesis work, Sony HMZ-T1, Carl Zeiss Cinemizer and Vuzix Wrap 1200VR are inexpensive and easily available HMD products. While HMZ-T1 and Wrap 1200VR have been replaced by a newer versions [60, 4], Cinemizer line is now being discontinued [28].

HMZ-T1 is a personal 3D viewer headset with OLED display panels and headphones [5], designed to give big virtual screen experience for games and movies. Similar to HMZ-T1, Cinemizer offered same multimedia playback features with integrated rechargeable battery for mobile use. Vuzix Wrap 1200VR adds head tracking capability on top of multimedia playback [29]. All three are shown in Figure 17.

Sony HMZ-T2 is the next iteration of the HMZ-T1 model with minor updates such

(a) Sony HMZ-T1 (b) Carl Zeiss Cinemizer (c) Vuzix 1200VR

Figure 17: HMD products by Sony [5], Carl Zeiss [28] and Vuzix [29].

as lighter body and no built-in headphones. It has the same technical specifications as the previous generation. The last product from the HMZ series is the HMZ-T3W headset which brings optional battery powered operation and wireless video connection for mobility.

Sony PlayStation VR is a virtual reality (VR) headset equipped with OLED 1920x1080 pixel display and accelerometer and gyroscope sensors to detect user's head movement with low latency. There are also LEDs mounted on the headset, to be used by an external camera unit to make positional tracking more accurate [30]. PlayStation VR is shown in Figure 18.

HTC Vive has 1080x1200 pixel displays per eye, inertial and laser position sensors. It can track user's movement in a small space if it used with two tracking base stations which provide laser light to the position sensors on the headset [31]. The Rift from Oculus has similar features with HTC Vive. It uses infrared LEDs which are detected by an external camera, to improve tracking performance by combining information from the camera and the inertial sensors[14]. Both HMDs are shown in Figure 18.

Google Glass is an optical see-through HMD which features smartphone like functionality with wireless connectivity and houses 640x360 pixel liquid crystal on silicon (LCoS) display, 5-megapixel camera, IMU (accelerometer and gyroscope) sensors [12].

The ZEISS VR ONE GX and Oculus Gear VR are mobile phone mounting headsets. They are designed to benefit from already existing high quality mobile phone

(a) Sony Playstation VR (b) HTC Vive (c) Oculus Rift

Figure 18: Next generation HMDs by Sony [30], HTC [31] and Oculus VR [14].

displays with a body made of relatively affordable materials and a pair of lens in front of viewers' eyes. VR ONE GX is one examples from many different variations of the Google Cardboard platform which only offers reference design and assembly instructions [33]. They are shown in Figure 19.

(a) ZEISS VR ONE GX (b) Zaak, Google Cardboard (c) Oculus Gear VR

Figure 19: HMDs to mount mobile phones by Carl Zeiss [32], Zaak [33] and Oculus VR [14].

Microsoft HoloLens is an untethered (cordless and battery powered) optical see-through augmented reality HMD, similar to Google Glass, which does not require any external device to operate. HoloLens maps real environment and overlay virtual scene objects on top of real world objects using built-in depth camera, video camera and IMU (accelerometer, gyroscope, and magnetometer) sensors [3, 61]. It can be seen in Figure 20.

Figure 20: Microsoft HoloLens [3].

2.2 Specifications of the HMD, cameras and inertial sensors used in the thesis work

The HMD unit that is used in the setup of this work is Sony HMZ-T1 which contains stereo OLED displays. The 0.7-inch displays can output 1280x720 pixel resolution video at 60 frames per second, creating a 750-inch virtual image in 20 meter distance to viewer's eyes. HMZ-T1 requires a processor unit to operate which is connected to a video source via an HDMI cable. Table 1 lists detailed specifications of the HMD while the processing unit and detailed views of HMZ-T1 are shown in Figure 21.

Table 1: Sony HMZ-1 Specifications [5].

Display Type	OLED
Display Resolution	1280x720
Video Frame Rate	60 fps
Aspect Ratio	16x9
Field of View	45°
Video Input	HDMI
Audio	Headphones
Virtual Image Size	750 inch in 20 meters
Weight - HMD	420 grams
Weight - Processor Unit	600 grams
Dimension - HMD	210 x 196 x 110 mm
Dimension - Processor Unit	180 x 36 x 168 mm

The two cameras that we use are Point Grey Flea3 FL3-U3-32S2C-CS which

(a) Wearer's view (b) Bottom controls (c) Processing unit

Figure 21: Sony HMZ-T1 details by Sony [5].

can capture 2080x1552 pixel frames at 60 Hz, feature GPIO (general-purpose input/output) pins to receive trigger signals and connect to host computer via USB3 interface. Both camera's standard lenses are replaced with Fujinon YV2.8x2.8SA-2 to increase field of view (FOV). Figure 22 shows the Flea3 body and Fujinon lens. Table 2 and 3 list detailed specifications of the Flea3 camera and the Fujinon lens.

The standard lens comes with the Flea3 camera does not have sufficient field of view, because of this it can capture image of only a small part of a scene. This affects both the performance of tracking the camera position in the scene and reduce to 3D viewing quality of a user. Figure 23 illustrates the difference between the lens which comes with the camera and a wide angle Fujinon lens which has viewing angle of about 100°.

(a) Flea3 camera body (b) Fujinon wide angle lens (c) Camera with lens attached

Figure 22: Point Grey Flea3 and Fujinon lens by Point Grey [6] and FUJIFILM [7].

The IMU device that we use is STM STEVAL-MKI062V2 "iNEMO" (**iNE**rtial

Table 2: Point Grey Flea3 Specifications [6].

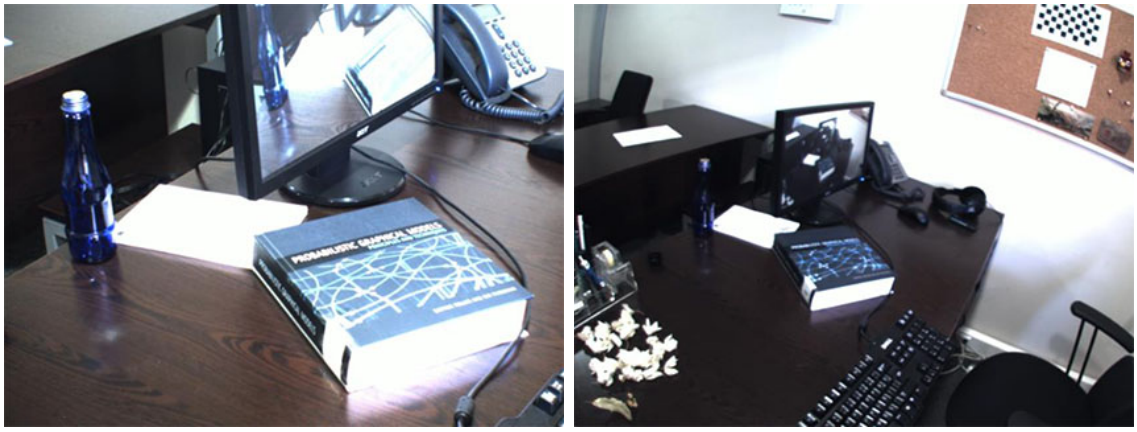
Model Number	FL3-U3-32S2C-CS (Color)
Max. Resolution	2080x1552
Frame Rate (Max.)	60 FPS
Sensor Type	Sony IMX036 CMOS
Readout Method	Rolling shutter, global reset
Sensor Format	1/2.8”
Pixel Size	2.5µm
Lens Mount	CS-mount
Exposure Range	0.01 ms to 32 seconds
External Trigger Modes	IIDC Trigger Modes 0, 1 and 15
Synchronization	Hardware or software trigger
I/O Ports	1 input, 1 output, 2 bi-directional
GPIO Connector	8-pin Hirose HR25
Interface	USB 3.0
Dimensions	29 x 29 x 30 mm
Weight	35 grams

Table 3: Fujinon Lens Specifications [7].

Model Number	YV2.8x2.8SA-2
Mount	CS
Focal length	2.8 mm - 8 mm (2.8x)
Focus	Manual
Iris Range	F1.2 - Close
Iris	Manual
Zoom	Manual
Weight	50 grams

MOdule) V2 development board, which houses both triple-axis accelerometer and triple-axis gyroscope sensors supporting data rates up to 400 Hz and can output logic signals from its GPIO pins. Figure 24 shows structure of the board.

Since STEVAL-MKI062V2 is a development board, it’s firmware can be easily modified for custom applications. For example, it can be setup for a custom frame rate or the GPIO pins can be used trigger output for synchronization. In our setup,



(a) With standard lens (b) With wide angle lens

Figure 23: Two images taken at the same position, with (a) the standard Flea3 lens and (b) the wide angle Fujinon lens.

one of the GPIO pins is connected to the trigger input of the camera in order to trigger frame capture. The detailed specifications of the board are listed in Table 4.

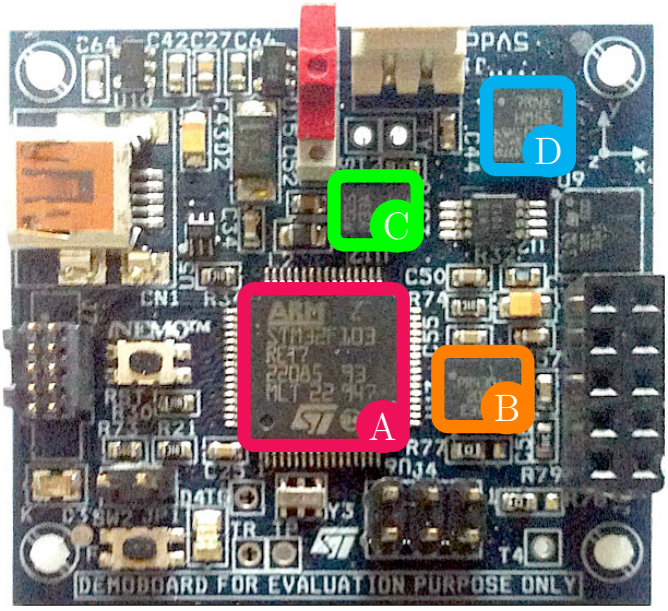


Figure 24: STM iNEMO development board. (A) Microcontroller unit, (B) Pitch and roll gyroscope, (C) Yaw gyroscope, (D) Accelerometer.

Table 4: STM iNemo Specifications [8].

Model Number	STM STEVAL-MKI062V2 iNEMO V2
Microcontroller	STM32F103RE ARM Cortex M3 32-bit
Interface	CAN, I2C, USB 2.0
Gyroscope-1	LPR430AL, Pitch and roll, 300°/s
Gyroscope-2	LY330ALH, Yaw, 300°/s
Accelerometer	LSM303DLH, ±2/±4/±8g
Pressure	LPS001DL, 300-1100mbar
Temperature	STLM75, –55 to +125°C
I/O Ports	SPI interface and 4 GPIOs
Frame Rates	1, 10, 25, 30, 50, 100, 400 FPS
External Memory	Micro-SD card slot
Dimension	4x4 cm

2.3 Overview of 3D animation rendering engines

Although requirements of every augmented reality application differ, the core functionality of a 3D animation rendering engine in an AR application is to generate virtual scene objects in real-time. Other than real-time rendering, a typical 3D engine provides integrated physics engine, collision detection, 3D sound, scripting, shaders, networking, authoring tools and scene graph. Each of these features can be used in different types of AR applications, for example using 3D sound to give more realistic feeling of immersion to users, or integrating networking to bring multi-player capability with collaboration.

There are many different 3D engines available and many online resources [62, 63] list detailed specifications of them. To decide which engine is best suited for a specific application, some important points should be considered:

Hardware acceleration: Almost every modern personal computer, smartphone or tablet has a dedicated 3D graphics processing unit called GPU. Using GPU for all graphics rendering tasks improve graphics performance considerably, allowing main processing unit to have more time on other tasks which results in overall performance

improvement. There are two types of interfaces to graphics card hardware, OpenGL and Direct3D. While Direct3D supports only Windows operating systems, OpenGL is only truly cross-platform option. Some 3D engines support only Direct3D or OpenGL, some other support both interfaces. For most cases, it is logical to select a 3D engine which integrates OpenGL, because OpenGL supports wide range of devices covering almost all desktop personal computer systems, modern smartphones, tablets and even some embedded system like set-top-boxes and smart TVs.

Multi platform support: Having more than one platform supported by 3D engine allows flexibility and efficiency for both development and for production of any application. For example developing a mobile application on a personal computer where 3D rendering results of the application can be tested, saves a lot of time, or even targeting multiple platforms at the same time without a major modification for the development. Nowadays, due to the popularity and the ease of access to smartphones and tablets, selecting a 3D engine which at least supports a desktop and mobile platform at the same time has important advantages over a single platform engine.

Programming languages and scripts: It is possible to develop with the language which is used to create a 3D engine. Most of the performance oriented engines use low-level languages such as C, C++ as the primary language. They optionally provide scripting language support to their programming interfaces, making development easier and practical for some cases. It is recommended to research before integrating a 3D engine and then select a development language and/or platform which is compatible with the engine. To eliminate any possible issues and inefficiencies, the primary language of the engine can be used to development of the application (e.g. C/C++).

Model file authoring tools: Some 3D engines have dedicated 3D modelling applications specifically designed to cover all aspects of their features to generate virtual

scene objects. Even in some cases it's possible to bundle an application inside the authoring tool, including programming logic and any other assets. Some others provide plug-ins, for already available and most of the time popular modelling applications, to create virtual content assets in a format which engine can read from. There is a trade-off between learning a new but very powerful tool while it's limited to a specific target and using one which has a lot of resource already available but comes with a lot of limitations compared to the first choice. For small projects or projects that have very tight schedule, going with an already existing modelling application to create virtual assets can save time and energy, while it gives an option to download freely or purchase online contents from different 3D content marketplaces or libraries.

Community: Having an active online community is a very important point to consider when deciding which engine to use for an application, even before checking technical capabilities of the engine. No official user manual or official tutorial can replace what an active community provides. Especially for community supported engines, this is the only way to get help when stuck at some point in development.

Licensing: There are different licensing options for 3D engines. Based on the licensing options, game engines can be roughly categorized in two: commercial and non-commercial engines. Commercial engines require payments, in different forms, for a license to use. Open source engines and non-commercial ones are free to use for almost all application types. It is important to point out some commercial engines have special licensing terms, mostly free license to use, for educational and research institutions.

References: For most cases, having many applications successfully developed using a specific engine gives helpful hint to understand what an engine is capable of in terms of technical features. It can be considered a good practice to investigate different applications, especially new and popular ones, during the decision making process, to see how well a specific 3D engine integrated with the application and

whether features, that are planned to use, are implemented and each perform well.

The list of these criterias played a big role on deciding the engine to be used in this thesis work. After a brief survey we found that OGRE open source 3D graphics engine seems to be well-balanced in terms of technical capabilities, community support and ease of use [9]. That's the main reason why the work in thesis employs OGRE (Object-Oriented Graphics Rendering Engine) for rendering and animation of the virtual objects of the augmented scene.

2.4 Description of the 3D engine used in the thesis

OGRE is a scene-oriented multi-platform 3D rendering engine written in C++ supports hardware acceleration for both OpenGL and Direct3D. It is one of the most popular open-source graphics rendering engines with an active community and has been used in several commercial products [64]. OGRE features object-oriented programming interface, content exporters for most 3D modelling software including 3D Studio Max, Maya, and Blender, plug-in architecture that allows addition of features, animation with hardware-accelerated skeletal animation and special effects including particle systems and full-screen postprocessing.

Although OGRE has been used to make games, it's not designed to be a game engine, its main purpose is to provide graphics rendering. It does not provide sound, networking, physics, etc. These features needs to be integrated separately. Table 5 lists detailed features of OGRE.

The OGRE's main role in this thesis work is to render stereo 3D virtual scene objects with the virtual stereo camera pair positioned and configured to mimic human eye. Before rendering virtual scene, it transforms virtual objects with the 3D tracking information and animates any animatable objects.

Table 5: OGRE Features [9].

Primary Language	C++
Scripting	Lua (Luabind)
API Support	Direct3D, OpenGL (incl. ES, ES2, ES3 and OGL3+), WebGL
Shader Support	Cg, DirectX9 HLSL, GLSL
Animation	Skeletal, Shape, Scene-nodes
Special Effects	Compositor, Full-screen Postprocessing, Particle Systems, Skyboxes, Billboarding
Cross-platform	Yes
Platforms	Linux, Windows, OS X, Windows Phone 8, iOS, Android
Model Exporters	3DS Max, Blender, Maya, Softimage, Cinema 4D, LightWave, Google Sketchup, RenderMonkey
License	MIT

2.5 Brief overview of EKF based tracking using visual and inertial data

Kalman filter (KF) takes series of measurements from a system over time where the measurement data contains noise and inaccuracies, and runs recursively to produce an estimation of the system state that is expected to be more accurate than ones obtained by using a single measurement [65]. The basic idea behind Kalman filter is to filter out the noise as much as possible and converge to the real value. It is an optimal linear estimator which utilizes all suitable information collected from a system. It estimates the current state of the system by processing all measurement data, regardless of measurement device accuracy, with employing predefined models of the system including measurement device characteristics, measurement errors, definition of the system noise, any uncertainty of the models, and any information available for the initial conditions of the system state [66].

Figure 25 illustrate a typical Kalman filter application for a system where the known controls drive the system while measuring devices provide the data of specific

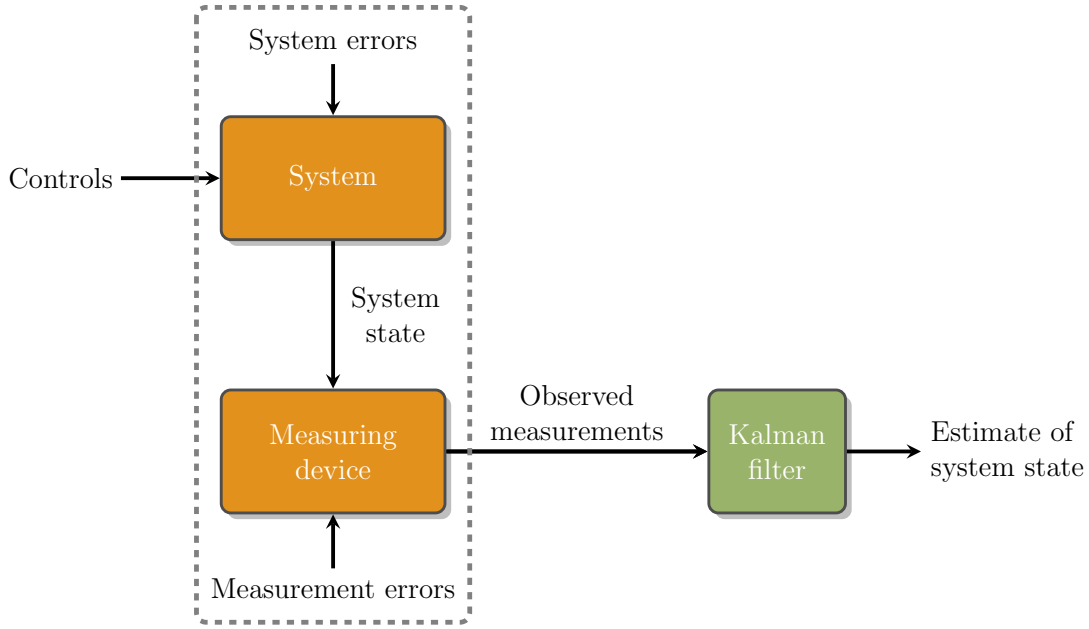


Figure 25: Typical Kalman filter application [34].

system variables. The controls are the inputs, the measuring devices are outputs of the system and the "state" is variables of interest which cannot be measured directly. Only the knowledge of inputs and outputs of the physical system are explicitly available to outside for estimating the system state. A Kalman filter estimates the specific system variables minimizing the error statistically by combining all available measurement data, and predefined models of the measurement devices and system [66]. It is "optimal" because it minimizes errors statistically, and "recursive" because uses prior knowledge about the system and measuring devices, under the condition that the state at a time t is derived from the prior state at time $t-1$. There is no need to have all previous data reprocessed every time, using only the present measurements data and the previously estimated state is sufficient.

In the Kalman filter algorithm there are two steps as shown in Figure 26, namely "prediction" and "update". A Kalman filter estimates the current state variables and uncertainties of the estimates in the prediction step. In the update state, the estimates of the state variables are updated by applying a weighted average, and the

estimates with higher certainty get more weight [65].

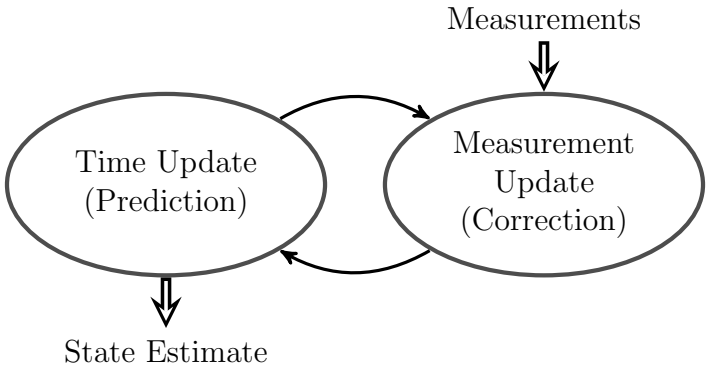


Figure 26: Kalman filter cycle.

It can be proven that Kalman filter is the best filter for a system that can be expressed using a linear model, assuming in that system, the system and measurement noise characteristics are Gaussian and white [66]. This means it can be applied only to linear Gaussian models, while almost all practical systems involve non-linearities of some kind. Extended Kalman filter (EKF) is an extension of Kalman filter which can be applied to nonlinear systems by approximating the system with a linear model. Due to linear approximation EKF does not have optimality guarantee. However, it has proven to be a de facto standard for making good estimation of the system state of many nonlinear systems, such as GPS and navigation systems [67].

The seamless and realistic blending of the virtual objects and real world scene is one of the essential requirements of an immersive augmented reality (AR) application. The accuracy of tracking the 3D pose of the HMD determines the performance of seamless and realistic blending effect. It is possible to track the 3D pose of the HMD by only using images captured by a camera, but common approach is improving the accuracy of tracking by incorporating measurement data from inertial and visual sensors. Accelerometers and gyroscopes can be used as inertial sensors and cameras can be used as visual sensors. Although, the tracking performance of cameras under slow motion can provide accurate tracking, they suffer form motion blur and relatively

low frame rate under fast motion that cause degraded tracking performance. On the contrary, inertial sensors (IMUs) can provide accurate measurements data under fast motion. But they can not be used alone for tracking, because of the errors buildup by accumulation of noisy derivatives which would cause tracking to be failed. Thus, when visual sensors are used together with inertial sensors, they complement each others weak points providing more accurate tracking performance.

The main objective of sensor fusion is to increase the tracking accuracy by incorporating the same or similar physical measurement data from different type of sensors. Fusing the measurement data from visual and inertial sensors is the common approach for estimating 3D camera pose and tracking. "Loosely coupled" and "tightly coupled" are the main methods for fusing visual and inertial measurement data. Inertial measurement data are used to improve the estimated camera pose in "loosely coupled" method [68], while visual and inertial measurement data are fused in a statistical filter in "tightly coupled" method [69].

For example, [69] implements a loosely coupled method to increase the accuracy of the pose estimation of the camera by using inertial sensor measurements for finding features in blurred images. The work of [70, 47] and [71] are some examples for tightly coupled method, where Kalman and particle filters used to fuse data from sensors. There are also other methods, where visual sensor is not part of the system, instead some other types of sensors are used (e.g. optical trackers) [72], benefit from Kalman filter for sensor fusion to improve tracking performance.

Since, the EKF algorithm has two steps, namely "prediction" and "correction" (Figure 26), gyroscope and accelerometer measurement data can be fused in four different approaches, specifically, the two used as control inputs at the time update stage, the two used as measurement inputs at the measurement update stage, and one used at the time update stage and the other one used at the measurement update stage. [47] gives extensive performance comparison of every possible combination of

fusing accelerometer and gyroscope data at different motion speeds in an EKF.

The sample results in this work were obtained with the tightly coupled Kalman filter method for fusing inertial and visual sensor data, using gyroscope and accelerometer data as measurement inputs at the correction stage.

In order to get a successful data fusion, accurate calibration of the camera and the inertial data is required [73]. Temporal and spatial calibration are two components of the calibration process between the camera and the inertial sensor. If cameras and inertial sensors are not calibrated effectively, errors from out of sync data would build up may cause loss of tracking. Topics related to calibration process will be discussed in more detail in the next chapter.

CHAPTER III

STEREO HMD SYSTEM

This chapter gives detailed description of the head-mounted system proposed in this thesis work. The function, inputs and outputs of each hardware and software system component, visual and inertial data capture, camera image rectification, spatial and temporal calibration issues, synchronization between cameras and inertial sensors, rendering 3D virtual scene, stereo image display for 3D viewing are explained in detail.

3.1 Overview of the system

The developed system includes a stereo HMD, a camera pair, an IMU and an AR application. The application's main purpose is to demonstrate and compare the performance of the tracking and rendering visually for different systems configurations. It is responsible for capturing visual and inertial data from the cameras and IMU sensors, processing the captured data, rendering 3D virtual scene objects and displaying augmented image on the HMD.

Figure 27 shows an overview of the system: The real world images captured by the cameras are preprocessed for lens distortion correction, rectification, and cropping. The processed left view of the rectified camera images are used for tracking, along with the data captured from the IMU sensors. The output of the tracking operation is a 3D pose (position and rotation) of the HMD system in the real world. The 3D pose information is used to transform virtual scene objects to render a virtual scene which reflects movements of the HMD. The rendered virtual scene images are overlayed on top of the real world images from the preprocessing step. The output of the overlaying operation is an augmented stereo image pair which is displayed on

the HMD for 3D viewing.

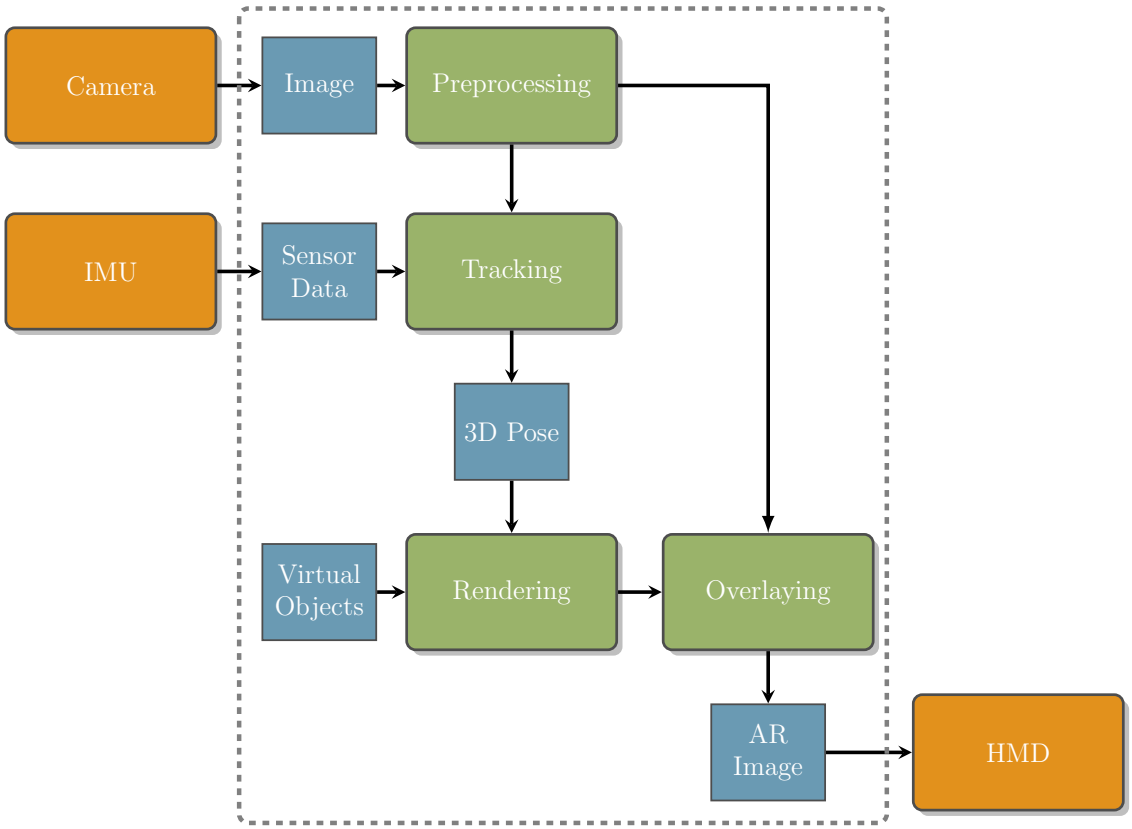
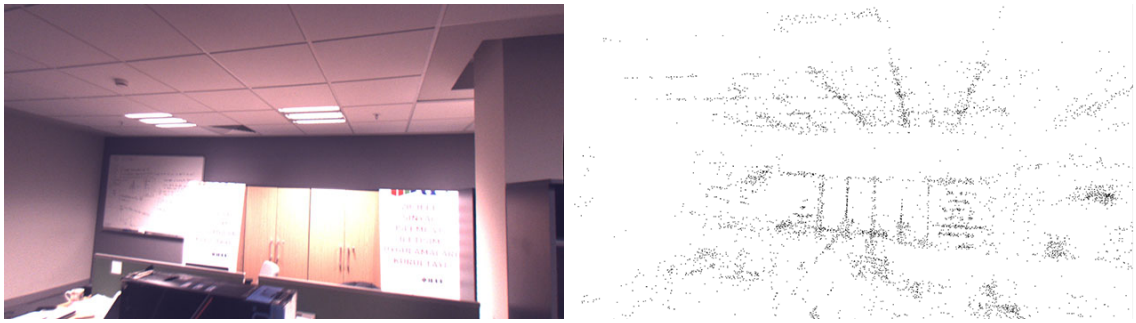


Figure 27: Overview of the HMD system.

While the tracking and rendering steps are online, the 3D feature map reconstruction of the scene and calibration steps needs to be processed offline. The camera intrinsic, extrinsic and distortions parameters are calculated in the calibration step using static images of a calibration pattern that is specifically designed for calibration, from random position and orientations while keeping the pattern gravity aligned. The calculated camera calibration parameters are used to undistort and rectify the captured static images. After this process, The Bundler tool [74], a "Structure from Motion" (SfM) system for image collections that are not ordered, is used to construct a 3D feature map of the scene and the camera pose from a short arbitrary motion of the HMD that includes captured video images and IMU data. Figure 28 shows a sample from a set of images of the scene used for 3D reconstruction of the camera

and scene geometry and the corresponding 3D point cloud which is generated by the Bundler application.



(a) Camera image (b) 3D point cloud

Figure 28: 3D scene reconstruction using Bundler.

3.2 HMD setup

The assembled head-mounted display system can be seen in Figure 29. A safety helmet is used as a platform for housing the components to create a single structure unit and to provide comfortable wearing of this relatively heavy system. The Sony HMZ-T1 HMD and a metal plate, where the Point Grey Flea3 FL3-U3-32S2C-CS cameras and the STM STEVAL-MKI062V2 development board are mounted on, are directly attached to the helmet. Both accelerometer and a gyroscope inertial sensors resides on the same STEVAL-MKI062V2 board. The board has custom firmware specifically modified to generate digital trigger output signals to the camera pair for synchronized stereo video and inertial data capture. Captured visual and inertial data are delivered to the host computer to be processed by the AR application. There are two cameras in the system, together they create depth perception of the scene. Both cameras are positioned side by side pointing same direction and making their optical axes parallel as best as possible.

The cables from the cameras and the IMU enter the inside of the helmet from an opening at the front, then they are routed between the protective inner mesh and the outer skin of the helmet. With the video cable of the HMD, they form a single wire

(a) Front view (b) Top view

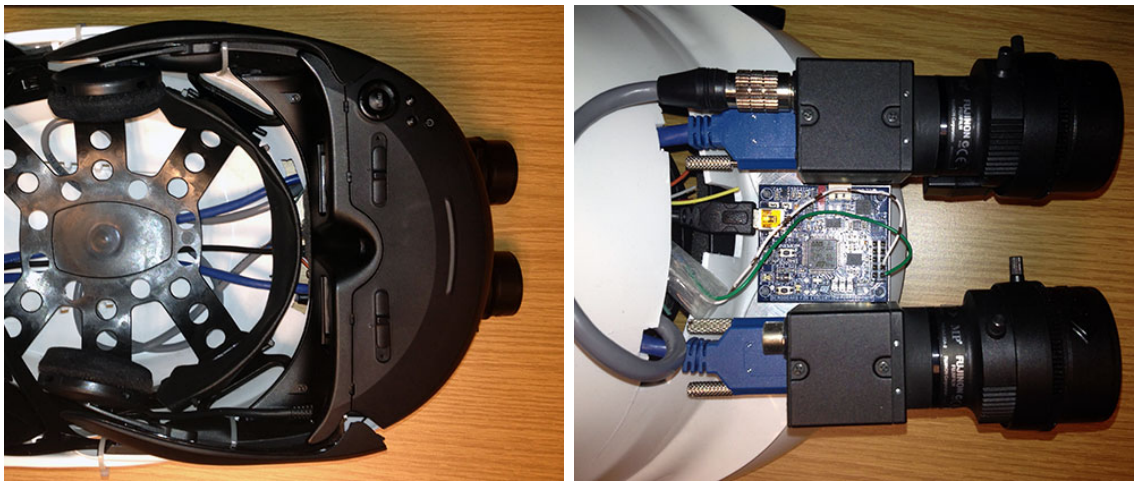
Figure 29: The HMD system with a stereo camera pair and an IMU module.

harness at the back and connect to the host computer.

Figure 29(b) and 30(b) show an additional trigger cable, connected to the IMU board and the left camera, which carries synchronization signals from the IMU to the cameras. There is only one trigger connection in this setup because the Flea3 cameras can sync each other if they are on the same interface bus (USB or FireWire). The bottom view of the setup showing the Sony HMZ-T1 HMD and part of the wire harness can be seen in Figure 30(a).

The AR application is a software module, running on Microsoft Windows 7 operating system, with user interface control elements for configuring the system for different AR development setups. The application's main task is to generate virtual scene blended with the real video captured from the cameras. It starts with preprocessing video image to get non-distorted (rectified) output, before generating stereo 3D virtual scene using the OGRE 3D graphics engine, then it transforms virtual objects with the 3D tracking information, and finally delivers the stereo output image to the HMD for 3D viewing.

Some selected frames from the sequence of the augmented scene animation can be seen in Figure 31. The virtual object is rendered by two virtual cameras inside



(a) Bottom view (b) Close-up view

Figure 30: The bottom view of the HMD setup showing the stereo displays and close-up view of the cameras and the IMU board.

the virtual scene of the 3D rendering engine. The virtual cameras are positioned and configured to mimic human eye to get realistic blending effect for the virtual objects and real world scene.

(a) Pair 1 (b) Pair 2

(c) Pair 3 (d) Pair 4

Figure 31: Sample stereo frames (left and right) of the augmented video.

The connection diagram of the HMD system can be seen in Figure 32. The cable between the IMU and the camera carries the trigger signal from the IMU to the camera.

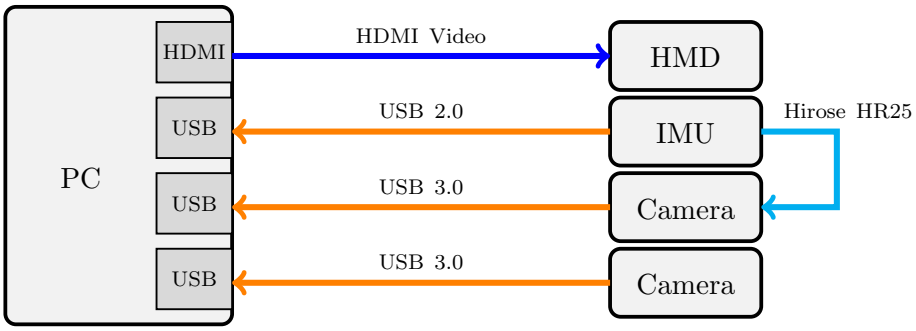


Figure 32: Connection diagram of the HMD system.

3.3 Calibration of camera and inertial sensors

One of the main contribution of this work is a set of effective and straightforward techniques for estimating the temporal offset (time lag) and spatial offsets (relative pose) between the inertial sensors and the camera. The use of these techniques do not need a complex system setup for collecting data and require simple equations to be solved.

The camera intrinsic parameters, relative pose of the two cameras, the temporal offset between the camera and the IMU, and the spatial offset between the camera and the IMU are the variables need to be calculated in the calibration process. The camera calibration method described in [75] is used to calculate the intrinsic parameters of the camera and the relative pose between the two cameras by processing multiple images of an object whose 3D geometry is known with good precision. The following two sections give detailed information about the temporal and spatial calibration methods.

Figure 33 illustrates the coordinate systems and the notations used in the following two sections. The subscripts s, w and c represent coordinate systems of sensor,

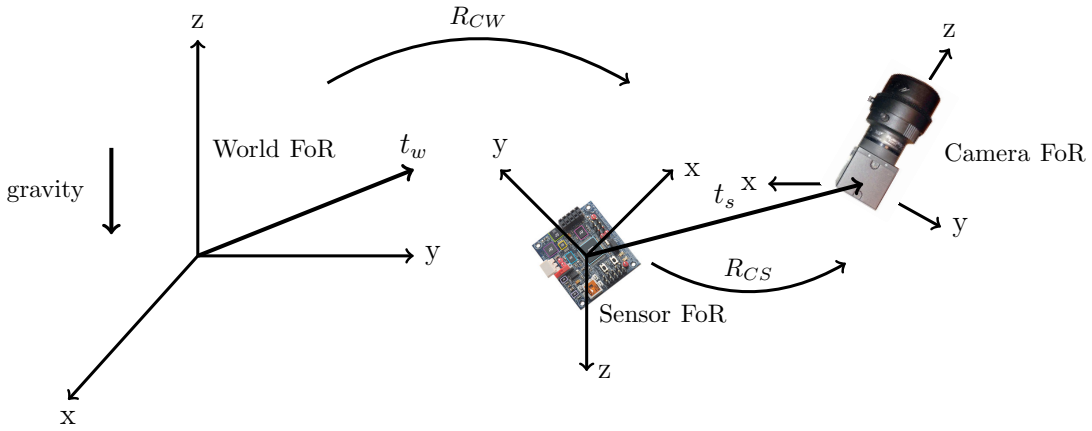


Figure 33: The inertial sensor, camera, and world coordinate systems and their relations.

world and camera, respectively. The rotation matrix from world coordinate system to camera coordinate system is represented by R_{wc}. R_{cs} stands for the rotation matrix from camera coordinate system to sensor coordinate system.

3.3.1 Temporal calibration

The main objective of temporal calibration is to determine the time delay between the camera frame and IMU data captures. There will be a predictable time delay between the camera and the IMU frames when the IMU device triggers the camera while the time delay value should be expected to be mostly unpredictable when the cameras are triggered using software methods. Note that, capturing the camera image always takes longer compared to reading data from the inertial sensors. This states that the calculated camera pose is obtained at an instant within the interval of frame capture. This time difference also effects to the time delay between the IMU and the camera.

The sample diagram explaining the timing and data capture between the camera and IMU is shown in Figure 34. This figure also illustrates the hardware synchronization scenario for a system where the camera is triggered by the IMU device . Note that, the time delay Δt is constant during data capture if the exposure-time value of

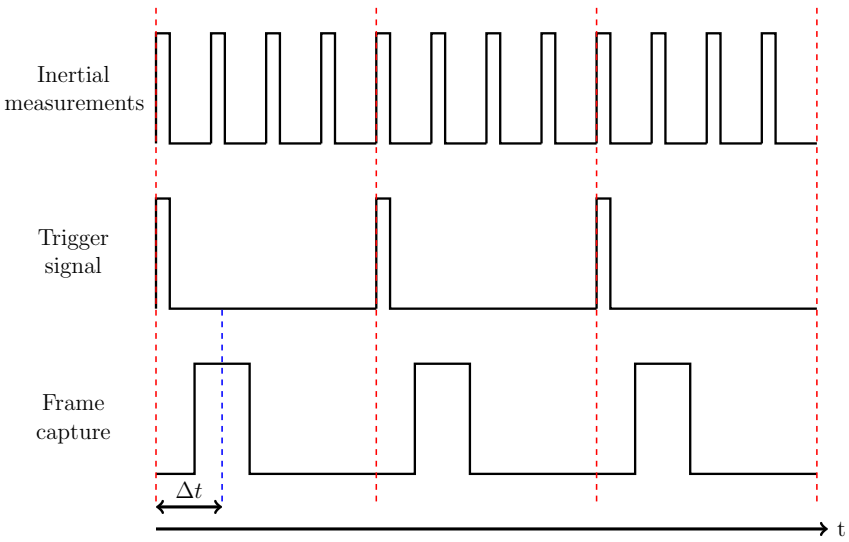


Figure 34: IMU and camera timing diagram.

the camera is not changed.

Since the camera and the IMU device are mounted on the same rigid frame, their relative pose are fixed all the time. So, the angular velocity of the IMU expected to be the same as the angular velocity of camera up to a certain rotation. The cross-correlation between the angular velocity of the camera and IMU needs to be estimated, in order to calculate the temporal offset.

Angular velocity of the camera is represented by ω_w. The relationship between the orientation of the camera R_{wc} and ω_w is given by

$$\frac{dR_{wc}}{dt} = [\omega_w]_\times R_{wc} \tag{1}$$

where subscripts ω represents world coordinate system, c represents camera coordinate system, subscripts $\times$ is cross product in the matrix form, and the rotation matrix from world to camera coordinate system is represented by R_{wc}. So, the angular velocity of the camera can be calculated using following equation

$$[\omega_w]_\times = \frac{dR_{wc}}{dt} R_{wc}^T. \tag{2}$$

Relationship between angular velocity of IMU and camera is expressed as following

$$\omega_s = R_{cs}^T \omega_w, \tag{3}$$

where subscripts s stands for IMU coordinate system, R_{cs} stands for the rotation matrix between the gyroscope and the camera. Note that, R_{cs} and R_{wc} are rotation matrices, making the magnitudes of ω_s and ω_w to be equal.

$$||\omega_s|| = ||\omega_w||.$$

Therefore, once the cross-correlation between the angular speeds of the camera and IMU is known, the time delay Δt between them can be easily calculated with the help of the following equation:

$$\Delta t = \underset{\Delta t}{\operatorname{argmax}}(\sum \omega_s(t)\omega_w(t + \Delta t)) \tag{4}$$

The IMU and the camera can be called to be temporally calibrated when the time delay value is calculated and that value is used to offset the IMU readings back in time in order to sync the IMU and camera frames.

3.3.2 Spatial calibration

Both accelerometer and a gyroscope inertial sensors resides on the same circuit board. Therefore, in most cases, data sheet of the IMU device provides the orientation value between the gyroscope and accelerometer sensors. For the case of the IMU device used in this work, the data sheet reports the orientation as "identity".

The gyroscope measures the angular velocity of itself and the measurements do not depend how it is positioned. Thus, the calibration process do not require the relative pose between the camera and the gyroscope. On the contrary, the operation of fusing the gyroscope measurements in the estimation of camera pose to be successful, the

rotation between the gyroscope and the camera should be known beforehand. This rotation is represented by R_{cs} in the following parts of this section.

The accelerometer measures mainly sum of linear acceleration and gravity but the measurements also combined with the effect of the angular motion. Since the relative pose between the camera and the accelerometer is not identity, the measured acceleration value will not be same with real acceleration of the camera even there is no gravity force applied to the accelerometer. The relative position (displacement vector) between the camera and the accelerometer is represented by t_s in the following parts of this section.

The main objective of spatial calibration is to estimate the value of R_{cs} matrix and t_s displacement vector. A set of static images of a gravity aligned calibration pattern is used to estimate R_{cs}. Captured measurement data during a rotational motion of the HMD system around all three axis is required to obtain the value of t_s. The following equation can be used to calculate the value of R_{cs}, when the accelerometer is not moving at a particular orientation:

$$R_{cw} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} = R_{cs}\gamma_s \tag{5}$$

where R_{cw} stands for the rotation matrix between world and camera coordinate systems, γ_s stands for accelerometer measurements, and g stands for gravity. Multiple images of a known geometry is used as calibration pattern to compute R_{cw} matrix.

One additional way to calculate the relative orientation between the camera and the gyroscope is to use the relationship between their angular velocity vectors which is given by Eq.3. This equation can be reformatted to be used for calculating the value of R_{cs}.

Since the camera and the IMU device are mounted on the same rigid frame, their

relative orientation and displacement are fixed and do not vary. Thus, values for the spatial calibration can be estimated only once as an offline step and this process is repeated only when the structure of the system is changed. Estimation of t_s utilizes a simple calibration method which is described in [76]. If the camera has a rotational motion, the accelerometer measures different acceleration value, under the assumption that the the gravity is subtracted form the measurements, than the value observed from the camera. The difference in these two accelerations is expressed as following

$$R_{sw}a_w - \gamma_s = \omega_s \times (\omega_s \times t_s) + \dot{\omega_s} \times t_s \tag{6}$$

where a_w stands for the camera acceleration in world coordinate system. Eq. 6 is used to calculate the displacement vector t_s between the IMU and the camera. This calibration technique does not require building complex systems and intricate calibration operations [77] in order to obtain the calibration parameters.

For the calculations, static images of a gravity aligned calibration pattern were captured from random position and orientations with the camera and IMU system (Figure 35). The Z axis of the pattern coordinate system needs be aligned with the gravity vector. The pattern has 6 vertical and 9 horizontal, total 54 2.5x2.5cm squares whose dimensions are known beforehand. The captured set includes images both the pattern fully visible and partially visible and they are used to calculate the position of the camera with the RANSAC (**RAN**dom **SA**mple **C**onsensus) method.

Figure 35: Gravity aligned calibration pattern.

3.4 Synchronization of camera and inertial sensors

The performance of a high-accuracy 3D tracking of an AR application highly depends on a synchronized stereo video and inertial data capture. Out of sync cameras and the inertial sensors would cause tracking divergence which may prevent giving a realistic feeling of immersion to users. There are two common basic methods to accomplish synchronization between the cameras and the inertial sensors: hardware and software.

Although implementing synchronization using software tools is relatively easy and does not require special features on the hardware to be synchronized, it introduces more latency and less predictable delays. Because performance of communicating hardware devices from a software layer relies on mostly uncontrollable and independent parameters from different levels such as the hardware and operating system of the host computer, the framework and programming language used to develop the software application, the driver of the hardware and even the optimization of the application, etc.

On the other hand, the hardware method takes advantage of low-level and direct hardware-to-hardware communication without needing a proxy (e.g. software application). This method requires hardware communication interfaces and ports (e.g. GPIO pins) dedicated to signaling, which is not supported by all types of hardware devices and implementing hardware synchronization may also require interfacing with

low-level programming on different platforms at once, and even dealing with electronic circuitry. The benefit of this approach is considerably less latency when compared to the software method and more predictable delays without any interference from the high-level components, which makes the hardware synchronization method ideal solution for applicable cases.

The STM STEVAL-MKI062V2 iNEMO IMU sensor board and the Point Grey Flea3 FL3-U3-32S2C-CS cameras were specifically selected for this HMD system setup. Because both devices allow a hardware synchronization implementation.

The Flea3 camera supports IIDC trigger "Mode-0", "1" and "15". The Mode-0 is the standard external trigger mode and it allows the camera to be triggered by using the GPIO pins as external hardware trigger or by using a software trigger. Falling or rising edge of the external hardware trigger input signal makes the camera start integration of the incoming light and the integration time is determined by the exposure time. In the Trigger Mode-1 which is also known as bulb shutter mode, external trigger input signal starts the camera integration of the incoming light. Integration time is equal to low state time of the external trigger input. It is possible to send a single software or hardware trigger signal to the camera in the Trigger Mode-15 where the camera captures and streams fixed number of images [78].

Since only Trigger Mode-0 is designed to be used in real-time video capturing, it is the only applicable option for the system setup proposed in this work, for sending external hardware trigger signals to the camera. There is an 8-pin GPIO connector on the back of the camera case. Three of the GPIO pins which are GPIO-0, GPIO-2 and GPIO-3 can be configured as trigger sources. Note that, it is not possible to trigger the camera at full frame rate using Trigger Mode-0. Because when an external source triggers a rolling shutter type camera, the maximum frame rate of the camera is determined as the half the maximum frame rate in free-running mode, independent of the frame rate that is requested from the camera [78]. Since the maximum supported

free-running frame rate is 60 fps for a Flea3 camera, the maximum rate in Trigger Mode-0 is 30 fps.

The iNEMO board has four GPIO pins and it comes with a development kit including the firmware library. This allows the board to be programmed to do certain tasks including sending custom signals from the GPIO pins at desired intervals. The firmware of the board used in the HMD system setup was customized to send trigger signals from one of its GPIO pins. The trigger signal is carried to the left camera which is configured to accept external trigger signals. Since the Flea3 cameras can sync each other if they are on the same interface bus, one trigger connection is sufficient. The plan is to send a trigger signal at every "Nth" of the board frames. Since the target frame rate of the camera is 30 fps, the 6 of available 7 standard frame rate values of the board are modified to make it divisible by 30, keeping the one rate for testing purposes. This method takes advantage of existing timing system used for the standard frame rates of the board which makes updating original firmware code relatively easy. Table 6 summarizes this modification.

Table 6: Frame rate values of the iNemo board

Original FPS	Modified FPS	Send Trigger at
1	1	every frame
10	60	every 2nd frame
25	120	every 4th frame
30	180	every 6th frame
50	240	every 8th frame
100	300	every 10th frame
400	360	every 12th frame

As a physical setup, there is a Hirose HR25 8 pin connector cable attached to the camera and two of its wires are connected to the iNEMO board. In this setup the GPIO-0 pin (pin-1) of the camera is connected to the GPIO-2 pin (pin-3) of the iNEMO board and ground (GND) connection is made between the ground pins of

both devices (pin-3 for the board and pin-5 for the camera). Figure 36 shows the wiring diagram of the synchronization setup. The camera capture starts with the falling or rising edge (depending on the configuration) of the trigger signal (logic output) sent from the IMU board. If the frame rate of the IMU board is selected to 60 fps, the camera will receive a trigger signal from the board at every 2nd frame and run at 30 fps synchronized with the IMU.

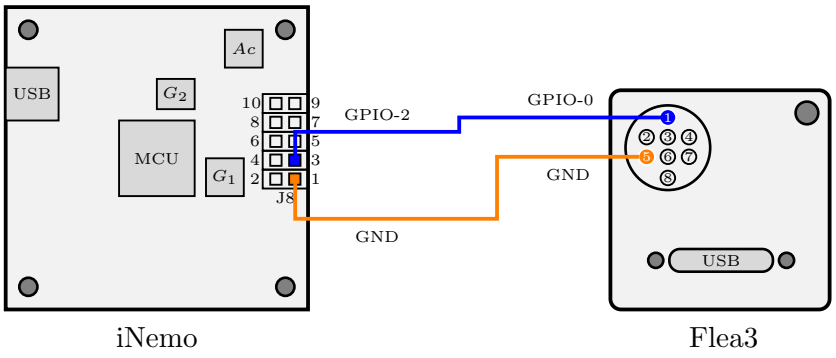


Figure 36: Synchronization trigger diagram between the IMU board and the camera.

3.5 Preprocessing of data before it is sent to the tracker

The camera calibration is the process where intrinsic, extrinsic and distortions parameters of the camera are calculated for lens distortion correction, stereo rectification, and undistorted image cropping.

Pinhole cameras introduce two major distortions to images, radial distortion and tangential distortion. Radial distortion makes straight lines appear curved. The effect is more apparent at the points away from the center of image. The tangential distortion which occurs because camera lens is not aligned perfectly parallel to the imaging plane. As a result, some areas in image may look closer than expected. The geometry, shape and scale of objects in the image are modified compared to the real-world due to these distortion effects.

The distortion effects need to be corrected so that the image exemplifies the real-world and can be used in a stereo AR application (e.g tracking). Figure 37 shows

how objects which are supposed to be straight (red lines), like the table top and the columns are curved in the image.

Figure 37: A image taken with a Point Grey Flea3 to show effects of distortions.

In addition to the distortion, intrinsic and extrinsic parameters of the camera are required for calibration. Intrinsic parameters are specific to a camera, which includes focal length and optical centers of the camera. These parameters do not depend on the scene viewed. If the focal length of the camera is fixed (same lens and zoom), this information can be stored and re-used for different scenes. The extrinsic parameters form a transformation matrix which describes the position and the orientation (or camera motion) of the camera in a static scene. Equivalently, they can be used to describe rigid motion of an object in front of a still camera.

In the pinhole camera model, a scene is formed by transforming the coordinates of 3D points onto the image plane of the camera using perspective projection [79]:

$$sp = A[R|t]P \tag{7}$$

Equivalently, this model can be written as:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{8}$$

where, X, Y, Z are 3D point coordinates in the world coordinate space, u, v are the coordinates of projected point in pixels, A is a camera intrinsic matrix, c_x, c_y is a point at the image center and f_x, f_y are the camera focal lengths in pixels. When a resize operation with a scale factor is applied to an image captured from the camera, same factor should be used to scale all of these parameters.

The transformation matrix $[R|t]$ is the matrix of extrinsic camera parameters, which transforms coordinates of a point (X, Y, Z) to the camera coordinate system. When $z \neq 0$, this transformation can also be expressed as following [79]:

$$\begin{aligned} \begin{bmatrix} x \\ y \\ z \end{bmatrix} &= R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \\ x' &= \frac{x}{z} \\ y' &= \frac{y}{z} \\ u &= f_x * x' + c_x \\ v &= f_y * y' + c_y \end{aligned} \tag{9}$$

the extended model to cover radial distortion and tangential distortion:

$$
\begin{aligned}
\begin{bmatrix} x \\ y \\ z \end{bmatrix} &= R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \\
x' &= \frac{x}{z} \\
y' &= \frac{y}{z} \\
x'' &= x' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2(r^2 + 2x'^2) \\
y'' &= y' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_2 x' y' + p_1(r^2 + 2y'^2)
\end{aligned} \tag{10}
$$

where,

$$
\begin{aligned}
r^2 &= x'^2 + y'^2 \\
u &= f_x * x'' + c_x \\
v &= f_y * y'' + c_y
\end{aligned} \tag{11}
$$

p_1 and p_2 are coefficients of tangential distortion, k_1, k_2, k_3, k_4, k_5, and k_6 are coefficients of radial distortion.

In the system of this thesis work, the camera is a Point Grey Flea3 FL3-U3-32S2C-CS with a Fujinon YV2.8x2.8SA-2 wide angle lens attached. To find all the cameras parameters (f_x, f_y, c_x, c_y, k_1, k_2, k_3, k_4, k_5, k_6, p_1, p_2) of this setup, a checkerboard pattern specifically designed for calibration, which has 6 vertical and 9 horizontal 2.5x2.5cm squares, was used. The parameters were calculated by mapping already known 3D coordinates of the corners (of the patterns squares) to the 2D coordinates on the image. The pattern was moved to position it in as many different locations as possible as shown in Figure 38, especially to the corners where the distortion effect is most visible.

The lens distortion corrected camera image of Figure 37 can be seen in Figure 39. The last operation of the camera calibration is to get the maximum useful rectangle of the corrected image. Since the distortion effect is concave, the closest edge points

Figure 38: Mosaic of images used for camera calibration.

to the center on the distorted image, are also the closest ones in the undistorted image. So, the closest points found on the distorted image coordinates can be used to calculated new points for the undistorted image. Figure 40 shows the cropping rectangle and the cropped output image. Note that, after the cropping operation, dimension of the output image will be different from the source, so the width, height and the center point need to be updated to reflect this change.

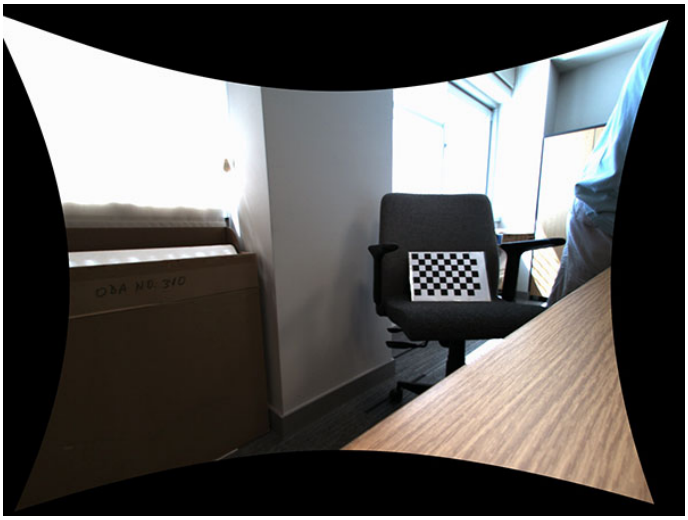
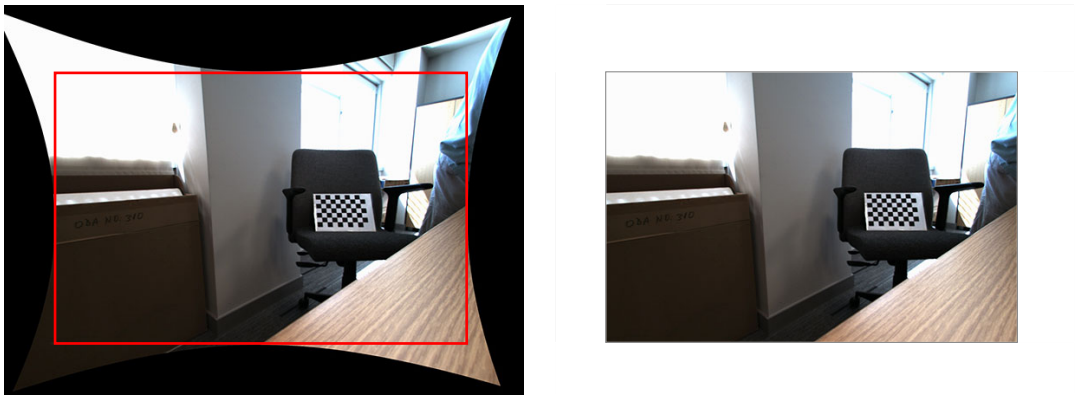


Figure 39: Distortion corrected camera image.

In the setup of this thesis work, the left and right camera were configured to capture 1600x1200 pixel images. After the lens correction process each captured image is cropped to 1152x768 pixel. The Sony HMZ-T1 HMD supports side-by-side stereo 3D at maximum of 1280x720 resolution. To create a stereoscopic 3D image on the HMD, the cropped images from the left and right cameras need to be fit inside the horizontally split left and right sections of the HMD display. Specifically, scaling the 1152x768 pixel image to 640x720 pixel size or even smaller size for some cases where a small amount of offset is required around the border of the image and/or between the left and right images.



(a) Cropping rectangle (b) Cropped image

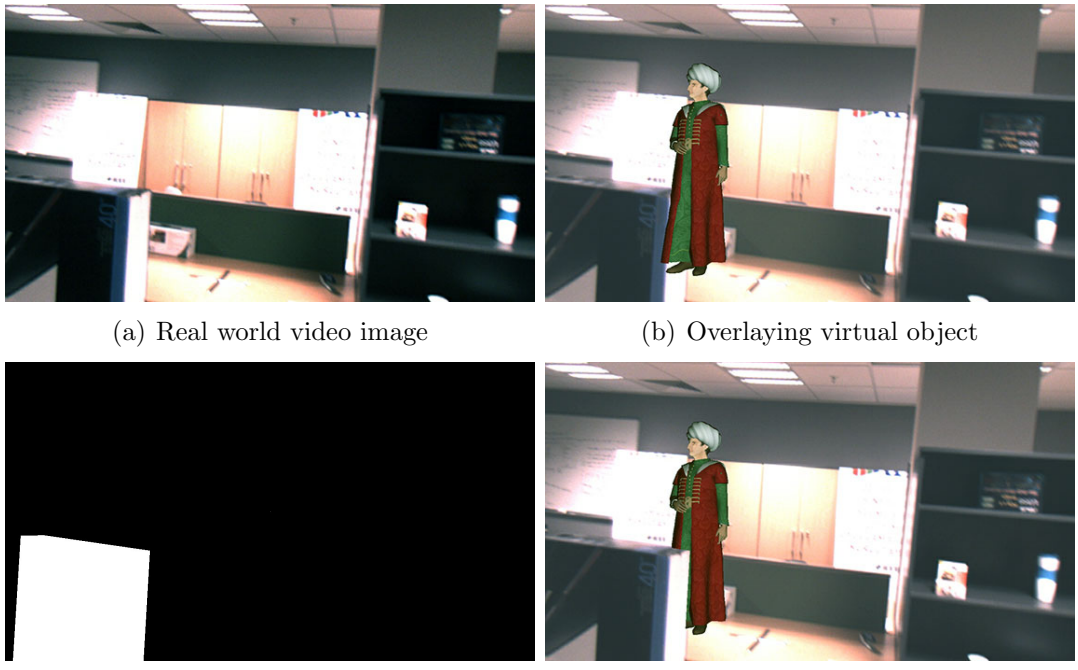
Figure 40: Cropping undistorted camera image.

OpenCV (Open Source Computer Vision) is a popular cross-platform computer vision library which implements the camera calibration approach [79] described in this section and provides all necessary tools to use it (e.g. programming interfaces, sample applications). It was used to calculate all required camera parameters in this work. The application of the HMD setup also takes advantage of OpenCV by using a pixel mapping file, which is generated as an additional output during the parameter calculation to make online correction operation without any complex calculation, to undistort a raw camera image easy and fast.

3.6 Stereo animation rendering

The OGRE graphics engine was integrated into the AR application of the HMD system to render and animate virtual objects of the augmented scene. The virtual scene rendered by the graphics engine using pair of virtual cameras and the real scene captured by the real cameras of the HMD system are blended into one image to create the final image of the augmented scene.

OGRE provides set of API (application programming interface) methods to manage virtual scene, including loading the objects which form the virtual scene from file assets, animating any animatable scene objects, placing and controlling virtual cameras inside the scene, rendering the scene from the view of the virtual cameras and delivering the rendered image to the host application to be processed further or display on a screen.



(a) Real world video image (b) Overlaying virtual object

(c) Occlusion mask (d) Final image mask applied

Figure 41: Creating an augmented scene from real and virtual scenes.

The software application of the HMD system creates the virtual scene with an animatable character which is rendered from a stereo camera pair positioned and

configured to mimic human eye. The rendering of both camera views takes place on every frame capture of the real camera image. The character is transformed based on the tracking information obtained from the tracker module. As the user moves the HMD system around the real scene, its 3D pose is computed by the tracking module and delivered to the rendering engine to position the virtual character in the real scene. It is also possible to occlude the virtual objects using the information obtained from the 3D reconstruction of the scene geometry. The Figure 41 shows the steps to augment a real environment where additionally the virtual character is positioned in such a way that some part of the character is occluded by a real box in front of it. The image of the occlusion mask of the box is created from the offline box geometry.

The final stereo animation rendering results of the sample scene displaying 3D position tracking of the system, real-time character animation and occlusion masking can be seen in Chapter 4.

CHAPTER IV

RESULTS

4.1 Calibration results

4.1.1 Temporal calibration

The technique described in 3.3.1 is used to calculate the time offset between captured camera images and IMU data. We have found that the time offset value for our HMD system setup is "48 milliseconds". This value is used to offset the IMU frames back in time to match the camera frames. Eq. 5 is used to calculate the relative orientation between the accelerometer and the camera, and Eq. 6 is used to calculate position offset between the IMU and the camera.

4.1.2 Spatial calibration

The first step spatial calibration, the rotation matrix R_{cs} from camera to sensor coordinate system was calculated using 15 different captured images of the calibration pattern and their corresponding IMU sensor data. The rotation matrix values found with this calculation is:

$$R_{sc} = \begin{pmatrix} -0.0373 & 0.0117 & -0.9992 \\ 0.9991 & -0.0209 & -0.0376 \\ 0.0213 & 0.9997 & 0.0109 \end{pmatrix}$$

The RMSE (root-mean-square error) of the calculated rotation matrix was found 0.000273. At the second step, the displacement vector t_s between the optical center of the camera and the accelerometer was calculated. For this calculation, a subset of the captured image and sensor data measurement was selected, whose corresponding accelerometer and gyroscope data values are fairly different from others in the

set. Four of the measurements were used to find the displacement vector, all of the measurements were used to verify the results. Table 7 shows the values of the displacement vector for both the four measurements which were used to find the displacement vector and for all measurements. The RMSEs of the displacement vector values were found 2.2214 for the result of the four measurements, and 2.7254 for all measurements.

Table 7: Calculated values of the displacement vector between the camera and the accelerometer

t_s	t_x	t_y	t_z
four measurements (cm)	4.5031	2.8745	12.2595
all measurements (cm)	4.0544	2.5677	13.7885

4.1.3 Camera lens distortion correction

The distortion effects need to be corrected so that the image exemplifies the real-world and can be used in a stereo AR application. Table 9 and Table 8 list all the camera parameters obtained from a set of sample images taken with the Flea3 cameras from 30 unique positions which are used for lens distortion correction.

Table 8: Camera intrinsic parameters

Camera 1	Camera 2
$f_x = 1089$	$f_x = 1089$
$f_y = 1094$	$f_y = 1094$
$c_x = 810$	$c_x = 810$
$c_y = 653$	$c_y = 655$

Table 9: Camera distortion parameters

Camera 1	Camera 2
$k_1 = -0.383149$	$k_1 = -0.333974$
$k_2 = 0.238617$	$k_2 = 0.065913$
$k_3 = -0.000752$	$k_3 = 0.0005.8$
$k_4 = 0.003363$	$k_4 = 0.001793$
$k_5 = 0.070925$	$k_5 = -0.565567$
$k_6 = 0.0$	$k_6 = 0.0$
$p_1 = 0.0$	$p_1 = 0.0$
$p_2 = 0.214567$	$p_2 = -0.777799$

4.2 Performance of sensor fusion

The work of Erdem and Ercan [47] provide a detailed comparison of different fusion approaches in an EKF for tracking the HMD system developed in this thesis work under varying motion speeds. Their work compares nine different tracking approaches where the camera is used as measurement unit for all cases. The list starts with the camera-only case to provide ground truth data for comparisons, and includes eight possible combinations of using accelerometer and gyroscope sensor data as measurement or control inputs.

The major finding of their work is that fusing inertial sensor data improves 3D tracking accuracy. The resulting RMSE values obtained from their work confirm that tracking accuracy is improved more when both sensors are used as measurements inputs across all speeds. The results also show accelerometer helps to improve 3D position accuracy more compared to 3D orientation and gyroscope helps to improve 3D orientation accuracy more compared to 3D position. They also mention that gyroscope is a better option and it should be used as measurement input for a case where only one sensor is to be used.

4.3 Sample 3D animations

A sample AR application was developed as part of the this thesis work to demonstrate the performance of the whole system, specifically the 3D tracking accuracy, 3D rendering quality and sense of immersion which users experience. In the test setup a virtual animated 3D character is positioned on a table as the user moves the HMD system around the real scene. OGRE creates the virtual scene images by animating and than rendering the virtual scene objects. Captures frames from the cameras and the virtual scene images are blended into one as an augmented image to be displayed on the HMD screen. Some selected frames from the sequence of the augmented scene animation can be seen in Figure 42.

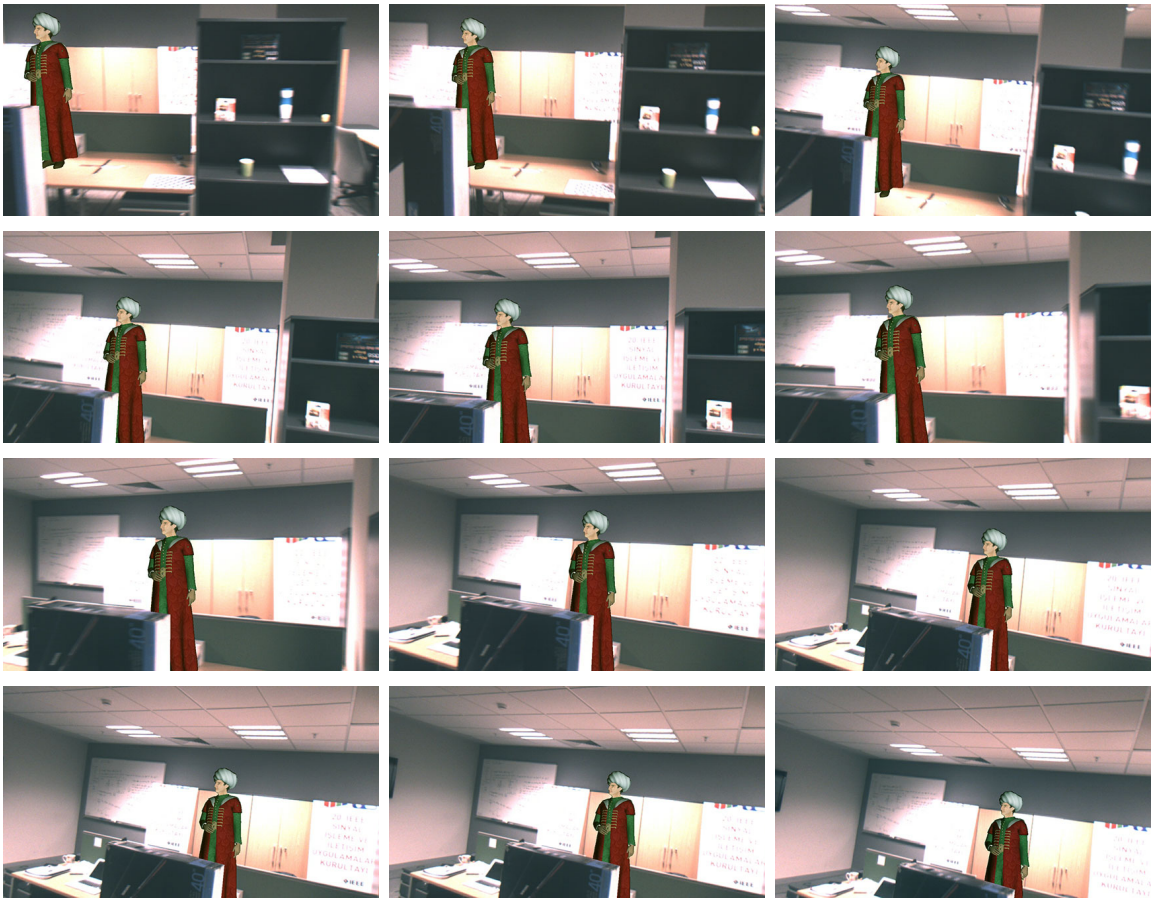


Figure 42: Stereo animation results.

CHAPTER V

CONCLUSION

This chapter summarizes the thesis, discusses its contributions and outlines directions for future work.

5.1 Contributions of the thesis

5.1.1 What are the specific outcomes of the thesis

In this thesis work, we have developed a multi sensor stereo HMD system for AR applications. The system contains a camera pair as both visual sensor and real video feed, accelerometer and gyroscope as inertial sensors, a head-mounted display as viewing screen for users, a personal computer where, a rendering engine which generates virtual objects and main processing software which orchestrates all mentioned hardware and software modules, run. It is specifically designed to be a research tool to provide a platform for researchers working on augmented reality, 3D tracking, sensor fusion, occlusion, calibration, etc. topics.

5.1.2 How the thesis contributed to the literature

This thesis work addresses spatial and temporal calibration topics between camera and IMU sensor. The presented calibration techniques do not require building complex systems for collecting sample measurement data or solving complex mathematical equations. Section 3.3 gives detailed information about the calibration techniques. Their efficiency and simplicity for obtaining accurate calibration between the IMU and the camera is covered thoroughly in [40]. The sample animation results obtained during our tests prove that the performance of this HMD system is satisfactory for both tracking and rendering.

5.2 Future work

There is clearly much research work to be done in the area of augmented reality. The work presented in this thesis work could be further developed in a number of different ways:

Perhaps the most direct extension of this work is by the means of using a new, up-to-date hardware systems, such as HMDs, cameras, inertial sensors, computing units, etc., which would boost both the performance of the AR system and the improve immersiveness.

Computational unit can be replaced with a mobile device. The AR software application can run on a tablet, smartphone or single board computer (e.g. Raspberry Pi), allowing the whole system to be structured as a single wearable hardware unit, similar to the concept wearable smart devices.

Powering the whole system by a battery would make it mobile which improves its usability and also gives more comfort to users.

Different methods for calibration, sensor fusion, structure from motion and tracking can be integrated into this system where they can be configured via a simple to use user interface controls for research purposes.

Tracking with different methods can be investigated. For example using laser or infrared emitter and receiver units to estimate the pose of the HMD system. Fusing these with visual and inertial sensor could be another addition to the work done in this thesis. This would lead to a better tracking performance.

The optimization of the software by utilizing hardware acceleration where possible. For example using graphics card's processing power (GPU) or using dedicated hardware components such as FPGA platforms for computing specific tasks to improve the performance of the software.

Bibliography

[1] Q. Hoellwarth, "Head-mounted display apparatus for retaining a portable electronic device with display," Apr. 1 2010. US Patent App. 12/242,911.

[2] Google, "Google Glass." `https://www.google.com/glass`, [Online; accessed 1-November-2015].

[3] Microsoft, "Microsoft HoloLens." `http://www.microsoft.com/microsoft-hololens`, [Online; accessed 1-November-2015].

[4] Sony, "Sony HMD." `http://www.sony.co.uk/electronics/head-mounted-display/t/head-mounted-display`, [Online; accessed 1-November-2015].

[5] Sony, "Sony HMZ-T1." `http://www.sony.com.au/product/hmz-t1`, [Online; accessed 10-November-2015].

[6] P. Grey, "Flea3 3.2 mp color usb3 vision (sony imx036) point grey usb 3.0," 2015. `https://www.ptgrey.com/flea3-32-mp-color-usb3-vision-sony-imx036-camera`, [Online; accessed 17-November-2015].

[7] Fujinon, "Yv2.8x2.8sa-2 — megapixel vari-focal 1/3", dc manual iris," 2015. `http://www.fujifilmusa.com/products/optical_devices/security/vari-focal/1-3-manual/yv28x28sa-2/`, [Online; accessed 17-November-2015].

[8] STM, *iNEMO: inErtial MOdule V2 demonstration board based on MEMS sensors and the STM32F103RE*, 2010. Rev. 2.

[9] OGRE, "Ogre - features – open source 3d graphics engine," 2015. `http://www.ogre3d.org/about/features`, [Online; accessed 19-November-2015].

[10] I. E. Sutherland, "A head-mounted three dimensional display," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pp. 757–764, ACM, 1968.

[11] T. P. Caudell and D. W. Mizell, "Augmented reality: An application of heads-up display technology to manual manufacturing processes," in *System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on*, vol. 2, pp. 659–669, IEEE, 1992.

[12] Wikipedia, "Google glass — wikipedia, the free encyclopedia," 2015. `https://en.wikipedia.org/w/index.php?title=Google_Glass&oldid=688952372`, [Online; accessed 4-November-2015].

[13] T. P. GmbH., "Trivisio engineering, prototyping and production of micro-electro-optical devices." http://www.trivisio.com, [Online; accessed 25-November-2015].

[14] Oculus, "Oculus Rift." https://www.oculus.com/en-us/rift/, [Online; accessed 11-November-2015].

[15] Ovrvision, "Ovrvision pro," 2015. http://ovrvision.com, [Online; accessed 4-November-2015].

[16] LEGO, "Lego digital box," 2015. http://stores.lego.com/en-us/stores/de/koeln/store-features/digital-box, [Online; accessed 5-November-2015].

[17] Wikitude, "Wikitude ar," 2015. http://www.wikitude.com, [Online; accessed 5-November-2015].

[18] C. Botella, M. Juan, R. M. Baños, M. Alcañiz, V. Guillén, and B. Rey, "Mixing realities? an application of augmented reality for the treatment of cockroach phobia," *Cyberpsychology & Behavior*, vol. 8, no. 2, pp. 162–171, 2005.

[19] T. Salonen, J. Saaski, C. Woodward, O. Korkalo, I. Marstio, and K. Rainio, "Data pipeline from cad to ar based assembly instructions," in *ASME-AFM 2009 World Conference on Innovative Virtual Reality*, pp. 165–168, American Society of Mechanical Engineers, 2009.

[20] W. Piekarski and B. Thomas, "Arquake: the outdoor augmented reality gaming system," *Communications of the ACM*, vol. 45, no. 1, pp. 36–38, 2002.

[21] E. Rose, D. Breen, K. H. Ahlers, C. Crampton, M. Tuceryan, R. Whitaker, and D. Greer, "Annotating real-world objects using augmented reality," in *Computer Graphics: Developments in Virtual Environments (Proceedings of CG International'95 Conference)*, pp. 357–370, 1995.

[22] G. Reitmayr and D. Schmalstieg, *Collaborative augmented reality for outdoor navigation and information browsing*. na, 2004.

[23] H. Kaufmann, "Construct3d: an augmented reality application for mathematics and geometry education," in *Proceedings of the tenth ACM international conference on Multimedia*, pp. 656–657, ACM, 2002.

[24] HP, "Sprout," 2015. https://sprout.hp.com/us/en/, [Online; accessed 7-November-2015].

[25] Geekologie, "Cool: Augmented reality advertisements," 2015. http://geekologie.com/2008/12/cool-augmented-reality-adverti.php, [Online; accessed 7-November-2015].

[26] Disney, "Make disney magic with augmented reality experience at disney store times square," 2015. http://blog.disneystore.com/blog/2011/11/, [Online; accessed 7-November-2015].

[27] V. Vlahakis, N. Ioannidis, J. Karigiannis, M. Tsotros, M. Gounaris, D. Stricker, T. Gleue, P. Daehne, and L. Almeida, "Archeoguide: an augmented reality guide for archaeological sites," *IEEE Computer Graphics and Applications*, no. 5, pp. 52–60, 2002.

[28] C. Zeiss, "The Cinemizer." `http://www.zeiss.com/cinemizer-oled/en_de/home.html`, [Online; accessed 4-November-2015].

[29] Vuzix, "Vuzix launches wrap 1200vr (virtual reality) video eyewear," 2015. `https://www.vuzix.com/wp-content/uploads/docs/_news/2011_News/Vuzix_Launches_Wrap_1200VR_Virtual_Reality_Video_Eyewear_09-20-2011.pdf`, [Online; accessed 10-November-2015].

[30] Sony, "Project morpheus, playstation vr," 2015. `https://www.playstation.com/en-us/explore/project-morpheus/`, [Online; accessed 7-November-2015].

[31] HTC and Valve, "Htc vive - steam vr," 2015. `http://www.htcvr.com`, [Online; accessed 7-November-2015].

[32] C. Zeiss, "The zeiss vr one gx," 2015. `http://zeissvrone.tumblr.com/cardboard/`, [Online; accessed 4-November-2015].

[33] Google, "Google cardboard," 2015. `https://www.google.com/get/cardboard/`, [Online; accessed 4-November-2015].

[34] P. S. Maybeck, "The kalman filter: An introduction to concepts," in *Autonomous Robot Vehicles*, pp. 194–204, Springer, 1990.

[35] E. Britannica, "Augmented reality in: Encyclopædia Britannica," 2015. `http://global.britannica.com/technology/augmented-reality`, [Online; accessed 1-Nov-2015].

[36] I. E. Sutherland, "The ultimate display," in *Proceedings of the IFIP Congress*, pp. 506–508, 1965.

[37] R. T. Azuma *et al.*, "A survey of augmented reality," *Presence*, vol. 6, no. 4, pp. 355–385, 1997.

[38] 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, pp. 34–47, Nov 2001.

[39] J. P. Rolland and H. Fuchs, "Optical versus video see-through head-mounted displays in medical visualization," *Presence: Teleoperators and Virtual Environments*, vol. 9, no. 3, pp. 287–309, 2000.

[40] A. Kermen, T. Aydin, A. O. Ercan, and T. Erdem, "A multi-sensor integrated head-mounted display setup for augmented reality applications," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2015*, pp. 1–4, IEEE, 2015.

[41] Wikipedia, "Oculus rift — wikipedia, the free encyclopedia," 2015. https://en.wikipedia.org/w/index.php?title=Oculus_Rift&oldid=688908311, [Online; accessed 4-November-2015].

[42] W. Steptoe, S. Julier, and A. Steed, "Presence and discernability in conventional and non-photorealistic immersive augmented reality," in *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pp. 213–218, IEEE, 2014.

[43] F. Mammano, "Looking through the oculus rift - augmented reality, virtual reality and gesture interaction," 2015. http://federico-mammano.github.io/Looking-Through-Oculus-Rift/, [Online; accessed 4-November-2015].

[44] Oculus, "Gear vr," 2015. https://www.oculus.com/en-us/gear-vr/, [Online; accessed 10-November-2015].

[45] F. Zhou, H. B.-L. Duh, and M. Billinghurst, "Trends in augmented reality tracking, interaction and display: A review of ten years of ismar," in *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 193–202, IEEE Computer Society, 2008.

[46] J. Newman, D. Ingram, and A. Hopper, "Augmented reality in a wide area sentient environment," in *Augmented Reality, 2001. Proceedings. IEEE and ACM International Symposium on*, pp. 77–86, IEEE, 2001.

[47] A. Erdem and A. Ercan, "Fusing inertial sensor data in an extended kalman filter for 3d camera tracking," *Image Processing, IEEE Transactions on*, vol. 24, pp. 538–548, Feb 2015.

[48] H. Fuchs, M. A. Livingston, R. Raskar, K. Keller, J. R. Crawford, P. Rademacher, S. H. Drake, A. A. Meyer, *et al.*, *Augmented reality visualization for laparoscopic surgery*. Springer, 1998.

[49] T. Sielhorst, T. Obst, R. Burgkart, R. Riener, and N. Navab, "An augmented reality delivery simulator for medical training," in *International Workshop on Augmented Environments for Medical Imaging-MICCAI Satellite Workshop*, vol. 141, 2004.

[50] A. D. Cheok, F. S. Wan, X. Yang, W. Weihua, L. M. Huang, M. Billinghurst, and H. Kato, "Game-city: A ubiquitous large area multi-interface mixed reality game space for wearable computers," in *Wearable Computers, 2002.(ISWC 2002). Proceedings. Sixth International Symposium on*, pp. 156–157, IEEE, 2002.

[51] Bulkypix, "Ar defender 2," 2015. https://itunes.apple.com/en/app/ar-defender-2/id559729773?mt=8, [Online; accessed 7-November-2015].

[52] DAQRI, "Anatomy 4d," 2015. http://daqri.com/project/anatomy-4d/, [Online; accessed 7-November-2015].

[53] R. VR, "Skyorb," 2015. http://www.realtech-vr.com/skyorb/index.php, [Online; accessed 7-November-2015].

[54] M. Billinghurst, "Augmented reality in education," *New Horizons for Learning*, vol. 12, 2002.

[55] H. Kaufmann, "Collaborative augmented reality in education," *Institute of Software Technology and Interactive Systems, Vienna University of Technology*, 2003.

[56] N. Slijepcevic, "The effect of augmented reality treatment on learning, cognitive load, and spatial visualization abilities," 2013.

[57] H. Kaufmann and D. Schmalstieg, "Mathematics and geometry education with collaborative augmented reality," *Computers & Graphics*, vol. 27, no. 3, pp. 339–345, 2003.

[58] G. Papagiannakis and N. Magnenat-Thalmann, "Mobile augmented heritage: Enabling human life in ancient pompeii," *International Journal of Architectural Computing*, vol. 5, no. 2, pp. 396–415, 2007.

[59] C. Hüroğlu, "A platform for developing and testing real-time ekf tracking applications for augmented reality," Master's thesis, Özyeğin University, 2014.

[60] Vuzix, "Augmented reality products," 2015. https://www.vuzix.com/augmented-reality/, [Online; accessed 10-November-2015].

[61] Wikipedia, "Windows holographic — wikipedia, the free encyclopedia," 2015. https://en.wikipedia.org/w/index.php?title=Windows_Holographic&oldid=690113011, [Online; accessed 17-November-2015].

[62] Wikipedia, "List of game engines — wikipedia, the free encyclopedia," 2015. https://en.wikipedia.org/w/index.php?title=List_of_game_engines&oldid=690925144, [Online; accessed 19-November-2015].

[63] WorldOfLevelDesign, "16 recommended 3d game engines (updated)," 2015. http://www.worldofleveldesign.com/categories/level_design_tutorials/recommended-game-engines.php, [Online; accessed 19-November-2015].

[64] OGRE, "Ogre - about – open source 3d graphics engine," 2015. http://www.ogre3d.org/about, [Online; accessed 19-November-2015].

[65] Wikipedia, "Kalman filter — wikipedia, the free encyclopedia," 2015. [Online; accessed 20-November-2015].

[66] P. S. Maybeck, *Stochastic models, estimation, and control*, vol. 3. Academic press, 1982.

[67] Wikipedia, "Extended kalman filter — wikipedia, the free encyclopedia," 2015. https://en.wikipedia.org/w/index.php?title=Extended_Kalman_filter&oldid=691236552, [Online; accessed 20-November-2015].

[68] Y. Yokokohji, Y. Sugawara, and T. Yoshikawa, "Accurate image overlay on video see-through hmds using vision and accelerometers," in *Virtual Reality, 2000. Proceedings. IEEE*, pp. 247–254, IEEE, 2000.

[69] G. S. Klein and T. W. Drummond, "Tightly integrated sensor fusion for robust visual tracking," *Image and Vision Computing*, vol. 22, no. 10, pp. 769–776, 2004.

[70] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "Gps/imu data fusion using multisensor kalman filtering: introduction of contextual aspects," *Information Fusion*, vol. 7, no. 2, pp. 221–230, 2006.

[71] L. Bai and Y. Wang, "A sensor fusion framework using multiple particle filters for video-based navigation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 348–358, 2010.

[72] R. Azuma and G. Bishop, "Improving static and dynamic registration in an optical see-through hmd," in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pp. 197–204, ACM, 1994.

[73] J. D. Hol, "Pose estimation and calibration algorithms for vision and inertial sensors," 2008. Phd.thesis.

[74] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM transactions on graphics (TOG)*, vol. 25, pp. 835–846, ACM, 2006.

[75] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.

[76] A. T. Erdem, A. O. Ercan, and T. Aydin, "Internal calibration of a camera and an inertial measurement unit," in *Signal Processing and Communications Applications Conference (SIU), 2013 21st*, pp. 1–4, IEEE, 2013.

[77] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 561–575, 2007.

[78] Point Grey, *Flea3 USB 3.0 Digital Camera Technical Reference*, 2014. Rev. 7.0.

[79] OpenCV, "Camera calibration and 3d reconstruction — opencv 2.4.12.0 documentation," 2014. http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html, [Online; accessed 25-November-2015].

VITA

Ahmet Kermen received Bachelor of Science in Electronics and Communication Engineering from İstanbul Technical University in 1999. In June 2010, he entered the Graduate School of Sciences and Engineering in Özyeğin University where he has been working on his Master of Science thesis. He has been working as a software engineer at Momentum Digital Media Technologies since 2005.