

Insert here your thesis' task.

CZECH TECHNICAL UNIVERSITY IN PRAGUE
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF SOFTWARE ENGINEERING



Bachelor's thesis

Building (3D object) visualization using augmented reality on iOS

Samuel Šušla

Supervisor: Ing. Martin Půpitel

11th May 2015

Acknowledgements

I would like to thank Ing. Martin Půlpitel for all the suggestions he made during writing the thesis and for opportunity to choose this topic. Further I would like to thank my parents and sister for support throughout my studies.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on 11th May 2015

.....

Czech Technical University in Prague

Faculty of Information Technology

© 2015 Samuel Šušla. All rights reserved.

This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Information Technology. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).

Citation of this thesis

Šušla, Samuel. *Building (3D object) visualization using augmented reality on iOS*. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2015.

Abstrakt

Hlavným účelom práce bolo nájsť optimálne riešenie pre 3D vizualizáciu budov na iOS zariadeniach. Neexistuje jedno univerzálne riešenie pre všetky prípady preto boli preskúvané viaceré možnosti a frameworky. Napriek viacerým frameworkom dostupným na trhu, bolo implementované vlastné riešenie používajúce accelerometer a gyroscope SceneKitu a CoreMotion.

Kľúčová slova Rozšírená realita, iOS, iPhone, iPad, Swift, Core Motion, SceneKit, Vuforia MetaIO, OpenCV

Abstract

The main goal of the thesis is to find the optimal solution for 3D building visualization on iOS devices. As a single best solution does not exist, an examination of multiple approaches and frameworks was conducted. Even though there are couple of frameworks on the market, a solution using accelerometer and gyroscope was implemented using SceneKit and CoreMotion.

Keywords Augmented Reality, iOS, iPhone, iPad, Swift, Core Motion, SceneKit, Vuforia, MetaIO, OpenCV

Contents

Introduction	1
Motivation	2
Structure	2
1 History and definition	5
1.1 History	5
1.2 Definition	6
2 Applications	7
2.1 Personal wearable computing	7
2.2 Industrial applications	7
2.3 Assembly	8
2.4 Maintenance	9
2.5 Medical applications	9
2.6 Entertainment	9
2.7 Collaboration	9
3 Techniques	11
3.1 Mechanical and ultrasonic	11
3.2 Global position system	12
3.3 Radio	12
3.4 Inertial	13
3.5 Optical	13
3.6 Hybrid	13
4 Augmented reality in smart phones	15
4.1 Wikitude World Browser	15
4.2 Ingress	16
4.3 Augment	16
4.4 Ikea Catalog	17

5	Requirements	19
5.1	Functional requirements	19
5.2	Non-functional requirements	20
6	Analysis	21
6.1	Global positioning system	21
6.2	Marker based augmented reality	22
6.3	Enhancing global positioning by image recognition	23
6.4	Marker combined with motion sensors	24
6.5	GPS with Bluetooth or Wi-Fi	25
6.6	Computer vision	25
6.7	Static observation	26
7	Technologies	29
7.1	iOS 8 and SceneKit	29
7.2	iPhone 6 plus	30
7.3	Swift	30
7.4	Core Motion Framework	30
8	Implementation	31
8.1	User Interface	31
8.2	Accessing motion sensors	32
8.3	3D scene	35
8.4	Combining SceneKit with CoreMotion	35
	Conclusion	37
	Evaluation	37
	Further development	38
	Bibliography	39
	A Acronyms	43
	B Contents of enclosed CD	45

List of Figures

1.1	Milgram's Virtuality Continuum	6
2.1	A car body is realized in Spacedesign using 3D devices and VR/AR visualization	8
3.1	Diagram showing principle of triangulation	12
4.1	Wikitude	16
4.2	Ingress	17
4.3	IKEA Catalog	18
6.1	Example of location based application, Bing Maps	22
6.2	Example of marker based augmented reality application in a mobile device	24
6.3	Application of Canny edge detector	26
6.4	Application of hough lines transformation	27
8.1	Three axes that run through the iOS device	33
8.2	3D model used in demo application	36

Introduction

Imagine investors pulling out their phones and viewing a building which only exists as a sketch in a computer at place where it could be built at the time, or tourists seeing a building that has been destroyed in the past. With the newest augmented reality technologies this could be possible and accessible to anyone with a smart phone.

Augmented reality used to be a thing of sci-fi movies. In the past decade augmented reality could be seen in television. In weather forecast program the green screen behind the forecaster is considered augmented reality, as well as during a swimming competition when names of the competitors appear on the surface of water.

Constantly increasing capabilities of mobile devices allow users to take augmented reality with them. With the newest mobile technologies developers can bring this high tech experience of augmented reality to people's hands and no longer a question of expensive gear or having a team of scientists anymore. Especially in last couple of years the market with these applications and devices has emerged significantly. [1] For instance Google Glass¹ was a project with an ambition to bring augmented reality through glasses, however google pulled the product off the market because it was still in the early development and not ready for production use.

Mobile platform has become the most popular platform for augmented reality applications because of its penetration on the market. All companies developing technologies for augmented reality are well aware of this and they focus their portfolio for developers particularly on mobile platforms. With these frameworks it is easier to develop such an application and therefore more applications using augmented reality appear on the market.

There are a couple of approaches and frameworks for augmented reality therefore it was decided to research all major approaches and frameworks. Afterwards the best one for visualization of 3D building will be implemented.

¹Google Glass (<https://www.google.com/glass/start/>)

Primary goal of the thesis is to visualize buildings for civil engineering. In this industry computer software is already being used to design and present buildings in programs such as 3D Max². This material would be just reused in the augmented reality application, it would be another visualization tool which would help users image how the structure is going to play along with the surrounding architecture.



Examples of augmented reality in various forms

Motivation

Even though augmented reality has been around for over 60 years there is still much to discover. Augmented reality has very promising future due to innovations in mobile devices and wearables. The combination of virtual world with the real one is useful concept that could provide the user with useful visualization or information.

Structure

In the thesis a lot of emphasis is placed on the research. In the chapter 1 history and definition of augmented reality is examined, what it exactly is and how the term was shaped through out the years. In the following chapters

²3D Max (<http://www.autodesk.com/products/3ds-max/overview>)

existing approaches and applications are examined. Later requirements for desired application are defined. Last two chapters focus on technologies used and details of implementation.

History and definition

Augmented reality is a broad and powerful concept, it is a full-scale medium composed of a number of technologies. This chapter shortly characterizes concept of augmented reality, its history and how the term was shaped throughout the years.

1.1 History

The term augmented reality has been established around 1990 but the concept started as early as in 1968 when American scientist Ivan Edward Sutherland build first system using augmented reality. [2] It was a transparent display attached to user's field of view, user's movement was monitored using mechanical parts of the system and ultrasound. Based on the movement of user system would draw simple information on the display.

The phrase Augmented Reality is supposed to have been coined by Professor Tom Caudell while working in Boeing's Computer Services' Adaptive Neural Systems Research and Development project in Seattle. In a search to find an easier way to help the aviation company's manufacturing and engineering process he began to apply virtual reality technology and eventually came up with some complex software that could overlay the positions of where certain cables in the building process were supposed to go. It meant the mechanics didn't have to ask or try to translate from what they found described in abstract diagrams in manuals. [3]

Next milestone in augmented reality was the definition of the term in 1992 at fifth Hawaii International Conference on Systems Science (HICSS). [4] Tom Caudell and David Mizell presented term augmented reality and showed its principle, inserting computer generated images into scenery of the real world.

In 1994 Paul Milgram and Fumio Kishino further shaped the meaning of the term. In their paper they focus on mixed reality visual displays which is a "particular subset of Virtual Reality (VR) related technologies that involve the merging of real and virtual worlds somewhere along the "virtuality

continuum” which connects completely real environments to completely virtual ones.” [5] A figure 1.1 Milgram’s Reality-Virtuality continuum and corresponding interaction styles. The upper part depicts Milgram’s continuum, which range from the real (i.e. physical) environment to immerse virtual environments. Between these extremes, there is a mix between real and virtual, hence the term Mixed Reality. [6]

The first paper that surveys the field of Augmented Reality was published in 1997 by Ronald T. Azuma. It goes through the technologies developed by the year 1997. The paper doesn’t bring anything new to augmented reality however at that time it was summary of all previous uses of augmented reality and summarizes the trade offs and approaches taken and speculates on future directions. [7]

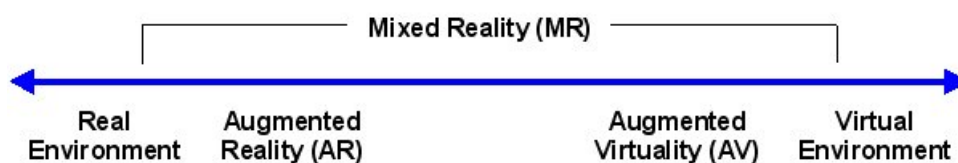


Figure 1.1: Milgram’s Virtuality Continuum

1.2 Definition

Until today a clear definition of augmented reality does not exist. Although first augmented reality like systems started to appear in 1960s, Augmented Reality only separated itself from virtual reality and became a research field at the beginning of the 1990s. Today two main definitions exist that describe Augmented Reality. Due to a lack of an official agreement on the term, both are accepted. Following the definition of Azuma an augmented reality system has to fulfill the three requirements. [8] [7]

1. Combines real and virtual
2. Interactive in real time
3. Registered in 3D

The first requirement is fundamental description of augmented reality, combination of real world with virtual. The second requirement separates augmented reality from the more general concepts of mixed reality or mixed media by requiring that the virtual content must be registered in 3D within the real world. Finally “Interactive in real time” requires the system to react to the user and update in real time which distinguishes augmented reality from all off-line augmentations such as the use of computer graphics in movies. [8]

Applications

Augmented reality could be used in various areas. The first systems focused on military, industrial and medical application. However augmented reality systems for commercial use and entertainment appeared soon after. This section discusses areas where augmented reality can find its application.

2.1 Personal wearable computing

Höllerer and Feiner believe one of the biggest potential markets for AR could prove to be in personal wearable computing. Augmented reality may server as more natural user interface for wearables and mobile devices. For example, augmented reality could be integrated into phone and e-mail communication with context aware overlays or manage personal information in relation to user's location. Geographers, archaeologists and many other professions could use augmented reality to place notes or signs in the environment they are working in. [9]

Navigation in certain environments has been tested. Heads up displays that overlay routes, highway exits, follow-me cars, dangers, fuel prices, etc. [10] Results clearly show that augmented reality displays result in a significant decrease in navigation errors.

ArcheoGuide project that reconstructs a cultural heritage site in Olympia, Greece. [11] With this system, visitors can view ancient architecture. Similar system has been developed for the Pompeii site. [10]

2.2 Industrial applications

Design, assembly, and maintenance are fields where augmented reality might find great application.

Fiorentino introduced the SpaceDesign MR workspace which allows visualisation and modification of car body, figure 2.1. [12] The MR Lab created

2. APPLICATIONS

Clear and Present Car, a simulation where user can open the door of a virtual car concept and experience the interior, dash board and interface design for usability testing. [10]

Another interesting application is the visualisation of robot programs such as the automated vacuum cleaner.

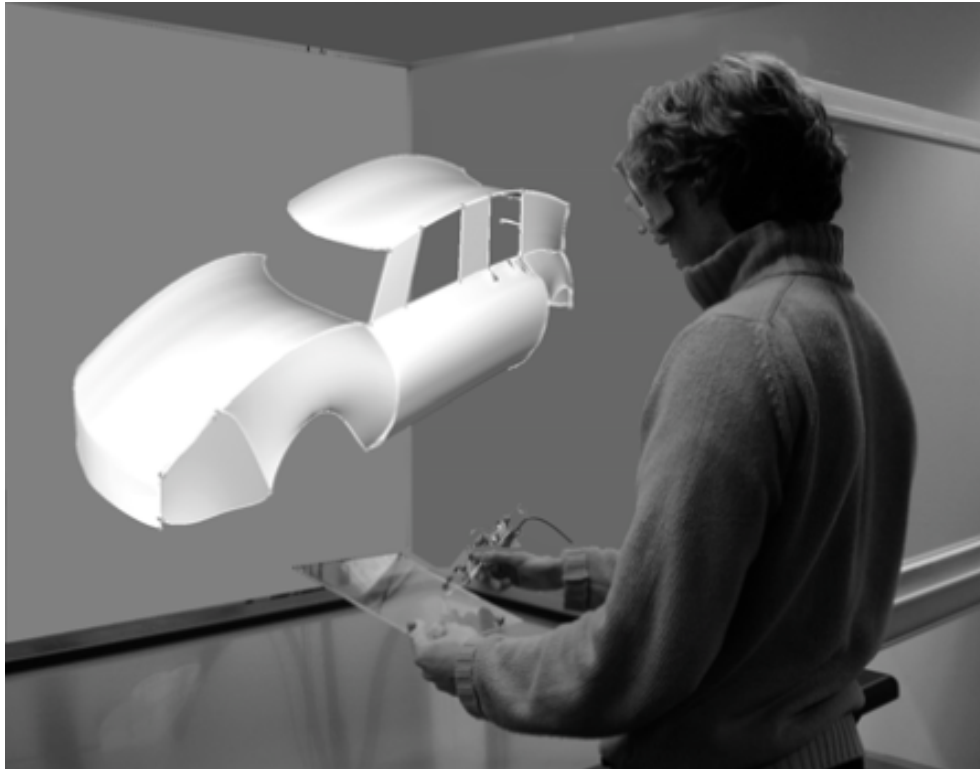


Figure 2.1: A car body is realized in Spacedesign using 3D devices and VR/AR visualization

2.3 Assembly

BMW has already experimented with augmented reality to improve welding processes on the cars. Volkswagen uses augmented reality in construction to analyze interfering edges, plan production lines and workshops, compare variance and verify parts. [13]

Benefit of using augmented assembly and construction is the possibility to constantly monitor and schedule individual processes in order to manage large complex construction projects. [10]

2.4 Maintenance

Complex machinery or structures require a lot of skill from maintenance personnel and AR is proving useful in this area, for instance in providing “x-ray vision” or automatically probing the environment with extra sensors to direct the users attention to problem sites.

Honda and Volvo already ordered Nomad Expert Vision Technician systems to assist the technicians with vehicle history and other useful information. System is supposed to be ”forgettable” which means that you forget you are wearing it. [14]

2.5 Medical applications

Nurses and doctors could benefit from important information being delivered directly to their glasses. However the level of accuracy surgeons require is beyond what augmented reality systems can deliver. [15] There are countless of medical technologies being developed.

2.6 Entertainment

Augmented reality can be used to create games but also to point out important game aspects in life sport broadcasting. It can also serve as advertisement to show virtual ads and product placements.

Like VR, AR can be applied in the entertainment industry to create AR games, but also to increase visibility of important game aspects in life sports broadcasting. In these cases where a large public is reached, AR can also serve advertisers to show virtual ads and product placements. On figure we can see how augmented reality informs us about nationality and name of each contestant. One augmented reality game will be discussed later in the thesis.

2.7 Collaboration

Multiple people viewing, discussing and interacting with the same 3D model in real time is a potential usage of augmented reality. This kind of collaboration would allow integration with existing tools and would enhance remote activities that would otherwise be impossible. There is number of major areas where augmented collaboration may apply. Doctors diagnosing 3D scan data, environmental planners, urban development, construction engineers. [16]

Techniques

Before an augmented reality system can display virtual objects into a real environment, it has to obtain information about the environment and track the viewer's movement and orientation. Movement uses three variables for position (x, y, z) and rotation has three angles for orientation (yaw, pitch, roll). This is called six degrees of freedom (6DOF). [10]

Most environments have to be prepared before an augmented reality system can reliably track 6DOF movement and not all tracking approaches work in all environments. To this day, determining the orientation of a user is still a complicated problem with no single best solution. [10]

Augmented reality tracking devices require higher accuracy and longer ranges. Distance of the objects to be annotated is crucial, the further away an object the less impact errors in position tracking have.

Tracking is easier in indoor settings because we have predefined environment that can be easily changed to system's needs and tracking device does not have to be mobile and deal with weather, abuse or shock. Unprepared outdoor environments are still problematic to track. [17]

In this chapter different approaches to track the viewer's movement will be discussed.

3.1 Mechanical and ultrasonic

These tracking approaches are restricted to indoor use because they have to be placed around the user. For instance first augmented reality system, head-mounted display [2], was using mechanical tracking. Then there are devices that send and receive ultrasonic sounds to perceive user's position. [18]

3.2 Global position system

Plain GPS is accurate to 10 to 15 meters. This can be increased to 3 to 4 meters using wide area augmentation system or using differential global position system for which the area has to be prepared by building a local station that sends differential error-correction signal, this way GPS can achieve accuracy of about 1 to 3 meters. [10]

3.3 Radio

For radio tracking method to be put in use preparation of the target environment is required. Preparation is quite extensive, access points have to be installed and target area has to be measured at various places to obtain signal strength. Wireless networking might be used, or iBeacons³. The accuracy depends on the density of deployed access points. The principle of radio based tracking is show on figure 3.1. [19]

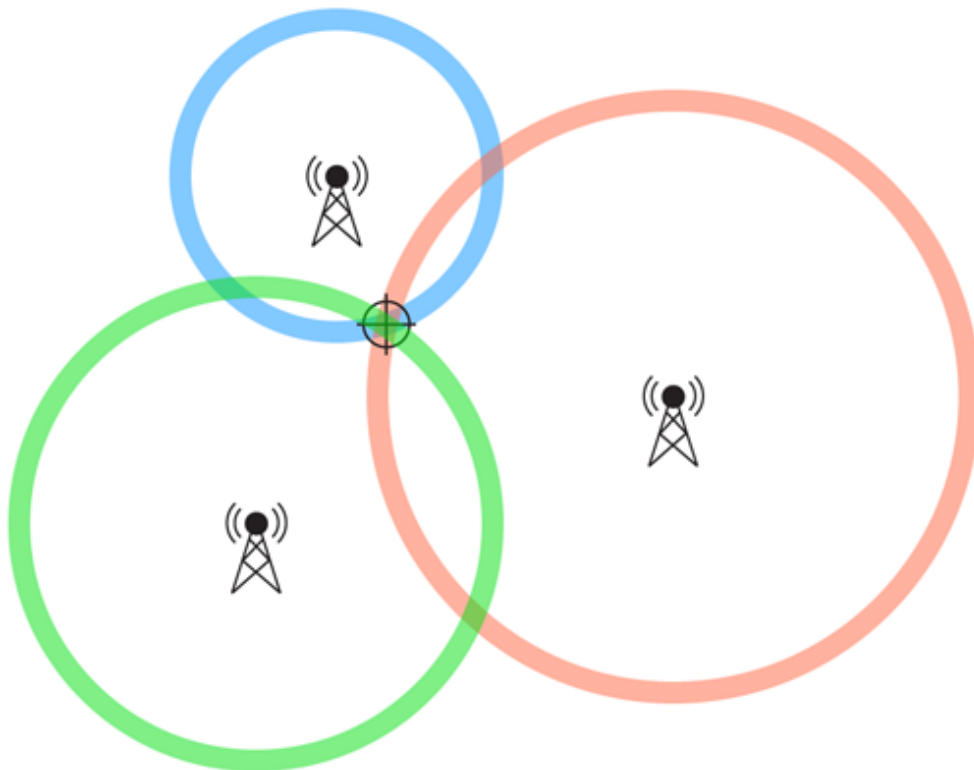


Figure 3.1: Diagram showing principle of triangulation

³iBeacons (<https://developer.apple.com/ibeacon/>)

3.4 Inertial

Accelerometers and gyroscopes, usually part of hybrid tracking systems, do not require prepared environments. They can provide estimated position when combined with accurate heading information. They have to be periodically updated to limit errors due to drift. They can also act as pedometers. Combining accelerometer and gyroscope mobile device can obtain its dislocation and this way track its position in 3D space, however for this approach to be precise it requires very accurate sensors. [20]

3.5 Optical

6DOF pose estimation of users and objects in general settings are vision-based. The field of view of the camera matches what the user is viewing therefore allowing for pixel perfect placement of virtual objects. Model-based approaches can recognize landmarks or detect relative movement between frames by using one or two small cameras. There are couple of techniques to detect scene geometry and camera motion. [10] While most of the approaches nowadays use markers to see how and where to place virtual object, markerless augmented reality is being extensively researched and promises amazing results. For instance homography to estimate trans-location and rotation from frame to frame, Harris feature detector to identify target points and the random sample consensus (Ransac) algorithm to validate matching. [21]

3.6 Hybrid

Hybrid tracking systems use multiple sources simultaneously. For instance electromagnetic compass, gravitational tilt sensors and gyroscopes for orientation tracking and ultrasonic, magnetic and optical for positional tracking. This approach appears to be the most promising way to tackle difficulties in indoor and outdoor mobile augmented reality environments. [9]

Augmented reality in smart phones

Augmented reality is a natural complement to mobile devices. With growing number of mobile devices, wide spread of augmented reality applications and frameworks has quickly followed. The biggest trend in modern augmented reality is its usage in marketing. [22] Except for marketing augmented reality applications can be found in all other categories of applications, from navigation to sporting apps and games. Let's take a look at some examples of augmented reality applications and technologies they use

4.1 Wikitude World Browser

Wikitude World Browser ⁴ 4.1 is showcase application from augmented reality framework going by the same name Wikitude⁵ which is leading augmented reality framework on the market right now. Application provides you with camera which shows you interest points around you and direction in which they are located. There are more than 3500 content providers and there is about any geographically-relevant information you may find valuable during traveling. The information is presented in form of Wikipedia articles giving you information about specific landmark or directions to the nearest ATM, any relevant information. Applications also makes a great use of services such as Yelp and Trip Adviser to find accommodation. [23] [24]

⁴Wikitude World Browser (<https://itunes.apple.com/us/app/wikitude/id329731243?mt=8&ign-mpt=uo%3D4>)

⁵Wikitude (<https://www.wikitude.com>)



Figure 4.1: Wikitude

4.2 Ingress

Ingress⁶ 4.2 is an augmented reality massive multi-player online role-playing location dependent game developed by Niantic Labs within Google. Players need to establish "portals" at places of cultural significance such as museums and monuments and link them to create triangular control fields over areas. There are two factions fighting for territory and you need to capture territory with the most "minds" or in other words people in it. The app uses GPS to obtain data about user's location. [25] [23]

4.3 Augment

Augment⁷ is an application that lets users visualize 3D model. It is designed for marketing purposes where you can bring more interactivity and user engagement to print media. For instance you could have a page about car in a magazine and through this application user can view the car as 3D model placed on top of the magazine. It is marker based technology.

⁶Ingress (<https://itunes.apple.com/sk/app/ingress/id576505181?mt=8>)

⁷Augment (<https://itunes.apple.com/us/app/augment-3d-augmented-reality/id506463171?mt=8&ign-mpt=uo%3D4>)



Figure 4.2: Ingress

4.4 Ikea Catalog

Ikea Catalog⁸ 4.3 makes great use of metaio⁹ augmented reality framework to allow their users to view furniture how it is going to look like in their homes. Usage is very similar to previous example, people scan desired furniture from magazine and afterwards place the magazine at the spot where they want to see the furniture appear. The magazine serves as marker and from close distance it works great.

⁸Ikea Catalog (<https://itunes.apple.com/us/app/ikea-catalogue/id386592716?mt=8>)

⁹metaio (<http://www.metaio.com>)

4. AUGMENTED REALITY IN SMART PHONES

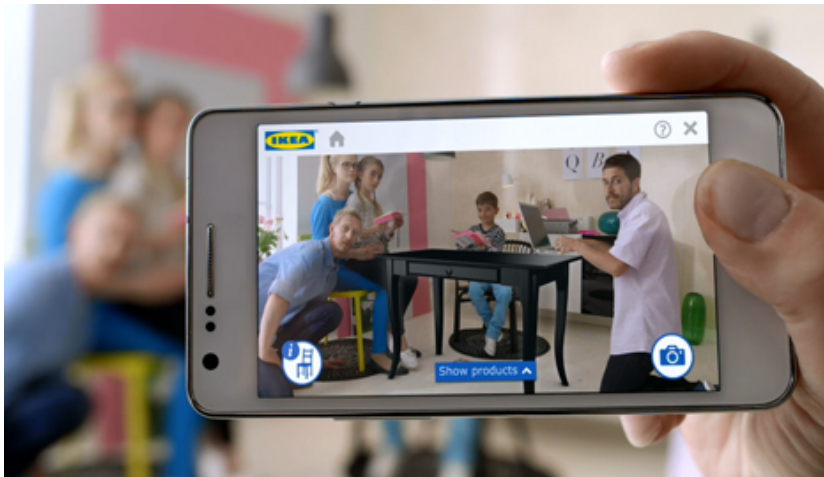


Figure 4.3: IKEA Catalog

Requirements

The following chapter defines requirements for the desired applications. Defined requirements have to be met for the augmented reality application. Available technologies and solutions were taken into account when defining the requirements. Every requirement is defined by its identifier, has short description and priority. MoSCoW method is a way to define various priorities for software requirements and is used in the analysis of requirements. [26]

5.1 Functional requirements

Functional requirements define function of a system or application so we know what is expected from our application and its functionality.

ID of requirement: RF-M1

Priority: Must have

Description: Application must be able to visualize building or another 3D object in distance.

ID of requirement: RF-M2

Priority: Must have

Description: Application must be able to visualize building or another 3D in real life size.

ID of requirement: RF-M3

Priority: Must have

Description: Application must be able to visualize object in all lighting conditions.

ID of requirement: RF-M4

Priority: Must have

Description: Application must be able to visualize object in outdoor conditions.

ID of requirement: RF-m4

Priority: Could have

Description: Application would allow user to view the object from different angles and distances by tracking his position.

5.2 Non-functional requirements

Non-functional requirements define non-functional parts of the application and tell us about performance, quality, platform etc.

ID of requirement: RN-M1

Description: Application must be available on iOS platform, iPad and iPhone must be both supported.

ID of requirement: RN-M2

Description: Supported iOS devices will be iPhone 5 and iPad 2 and higher.

ID of requirement: RN-M3

Description: User should have the feeling that the object really belongs to the scene.

ID of requirement: RN-M4

Description: Application must support 3D models in Collada (DAE) format.

ID of requirement: RN-M5

Description: Application should require minimal environment preparation.

Analysis

In this chapter different approaches to obtain 6DOF will be evaluated. Most approaches fall under hybrid category where multiple inputs are combined together. Augmented reality requires high level of accuracy therefore having multiple inputs that can substitute each other is beneficial. Level of accuracy depends not only on the model but also on the distance of the object to be displayed. The further away the object is, the less impact errors in position tracking have and more impact errors in orientation have.

6.1 Global positioning system

Research began with the possibility of using global positioning system to detect user's altitude and latitude on the globe and the direction of heading. The newest iPhone can combine GSM, GPS and Wi-Fi signals to get the most accurate position on the globe, to get heading it adds gyroscope, accelerometer and compass to the equation.

The procedure is as follows, model of the building would have fixed geographical coordinates. Afterwards the mobile device would use its location API to obtain data about location and direction in which the phone is pointed. If the building's coordinates would be in the view, it would be rendered on the screen at appropriate place and with correct rotation. It is ideal for applications such as one on figure 6.1 where the accuracy isn't critical and few meters or degrees do not make difference in overall user experience.

The approach is computationally very inexpensive and is easy to implement. Core Location framework provided by Apple already does most of the heavy lifting by combining all of the sensors and providing us the best possible location accuracy. There are numerous frameworks available such as Wikitude that would compute angles and distances afterwards render the information on screen for the user to see.

The problem with this approach is hidden in the accuracy. User would rely on global position system which is known to be very inaccurate under various

conditions, the accuracy dramatically lowers in densely populated areas because the signal is altered by surrounding constructions. The deviation of GPS is in tens of meters but mobile devices combine various sources of location, such as GSM and Wi-Fi network, to provide best accuracy. The inaccuracy causes the rendered model to constantly move on the screen and therefore neglect the effect of the model really belonging into its surroundings. The accuracy can be improved with Differential Global Position System (DGPS) which would bring the accuracy down to about 1-3 meters, however a station correcting the signal would have to be installed. [27]



Figure 6.1: Example of location based application, Bing Maps

6.2 Marker based augmented reality

Common solution for augmented reality is using a marker. On the spot where we want the model to be rendered we place a marker. It could be in form of a picture, however the picture has to meet certain criteria, it has to be rugged enough for the software to determine its relative position to the marker.

The instance phone sees the marker it renders the model in relation to the position of the marker. From just seeing the image through edge detection algorithms we are able to approximate position of the phone in relation to the marker. Image has to be full of features that can determine direction and distance, edges are ideal for this.

The advantages of this approach are that it is completely independent of location, therefore even devices without GPS unit can use this solution, only

camera with sufficient resolution is needed. Also the software is using the same perception like the user, vision, therefore the result appears to be very natural and realistic.

Disadvantage is marker itself, it has to be big enough and the marker has to be visible during the entire presentation. There is a workaround for that by using phone's gyroscope. After we lose sight of the marker we can approximate the position by determining the phone's rotation from motion sensors. This is called extended tracking¹⁰. With markers scale of the objects depends on size of the marker and it is not usable from bigger distance.

Every available marker framework for augmented reality implements marker based solution, one of them is Vuforia vuforia¹¹. Vuforia is an open source project under the roof of Qualcomm. Vuforia returns a matrix which determines position of observer in relation to the marker it can be used in coexistence with 3D render engine such as OpenGL ES¹² or SceneKit¹³.

6.3 Enhancing global positioning by image recognition

GPS localization combined with computer vision. Given location will be photographed from all angles so result images create panorama picture. To these images meta-data would be provided. The image from device's camera would be compared to panorama picture and from this we could determine position and this way know exactly where to render the model.

This solution is in theory very accurate and reliable. However it is very dependent on the place, in a city it would work great but at places such as forests or plains it would not be as effective. It is also time consuming to prepare each spot, photo shooting, filtering of bad images and processing.

It sounds like a very promising approach for other uses but it also depends on client-server architecture where client just keeps sending images to server and server does the analysis. This does require network connection and latency is very high neglecting the augmented reality effect. Therefore it is not suitable for our use case. [28]

¹⁰extended tracking (<https://developer.vuforia.com/library/articles/Training/Extended-Tracking>)

¹¹vuforia (<https://www.qualcomm.com/products/vuforia>)

¹²OpenGL ES (https://developer.apple.com/library/ios/documentation/3DDrawing/Conceptual/OpenGL_ES_ProgrammingGuide/Introduction/Introduction.html)

¹³SceneKit (https://developer.apple.com/library/mac/documentation/SceneKit/Reference/SceneKit_Framework/index.html)

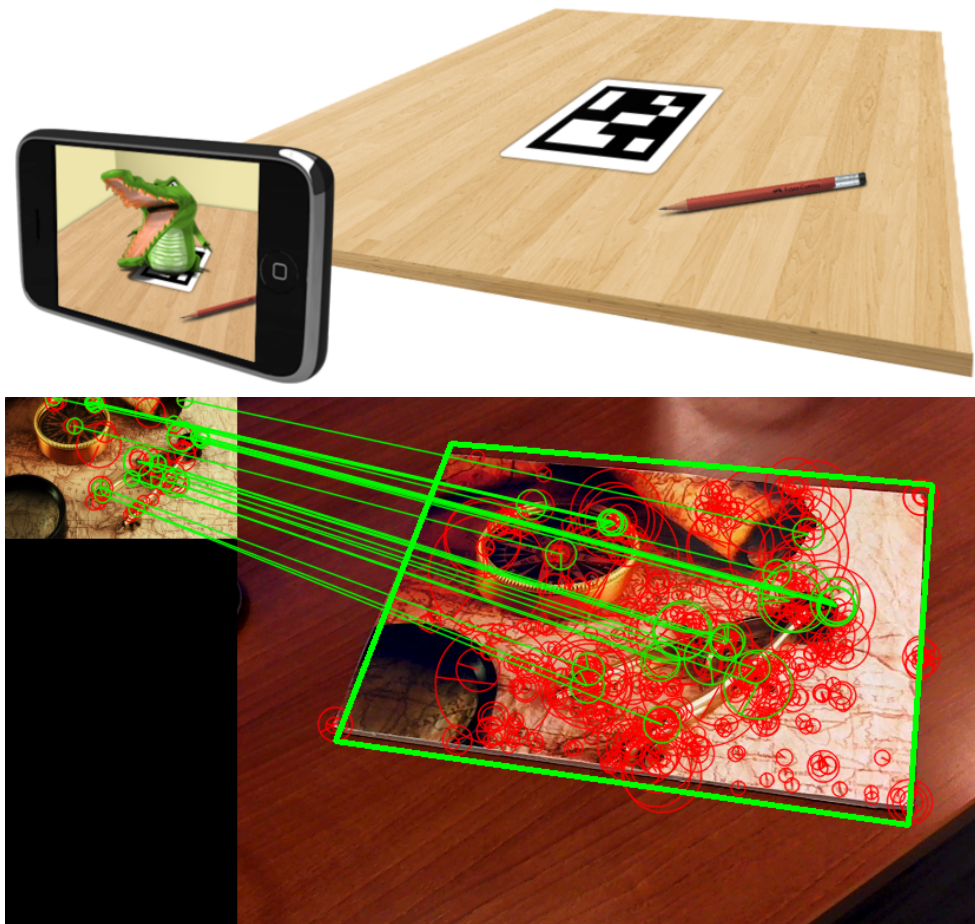


Figure 6.2: Example of marker based augmented reality application in a mobile device

6.4 Marker combined with motion sensors

Scanning would be exactly the same like in marker based solution. However the instance the marker would be recognized, this place would be marked as the beginning and marker scanning would turn off. From this moment we would rely only on smart-phone's sensors to determine the location of the phone. It is possible from phone's acceleration determine the distance and direction in which it has traveled by applying double integral. However this approach requires highly accurate and calibrated sensors because even the lowest deviation results in significant miscalculations. The deviation is in units of meters. There are functional prototypes however these are not mobile devices and they use highly specialized equipment to achieve this result. [20]

For this inaccuracy or need of expansive equipment we cannot use this approach but it appears to be very promising in the future.

6.5 GPS with Bluetooth or Wi-Fi

It would only improve accuracy of GPS unit by combining GPS with other sources of location. By measuring strength of a signal which source's location is known we can obtain distance from this source and having three sources like this we can approximate our position within space by using triangulation. Places with Wi-Fi antennas can use this signal otherwise iBeacons would have to be installed.

iBeacon is an indoor position system, Apple describes it as "new class of low-powered, low-cost transmitters that can notify devices of their presence." [29] It emits signal with information about its identity, UUID, major, minor.

Using Wi-Fis or iBeacons to improve the accuracy would require these devices to be installed at desired places and also measurements. User can also shade the signal with his own body which dramatically lowers the accuracy.

6.6 Computer vision

Computer vision is the best approach for vision based augmented reality because it uses the same input data as the output media is, the screen and camera. Marker based augmented reality uses computer vision to detect the image and extract positional data from it however markers are designed to be easily recognizable by computer vision software and without marker custom computer vision recognizer would have to be implemented.

As a computer vision option we examined framework OpenCV¹⁴ which has the biggest community support with over 47 thousand users. OpenCV was developed with performance in mind, it takes advantage of multiple cores.

OpenCV offers different algorithms, first one Canny Edge Detector¹⁵ which tries to satisfy three main criteria, low error rate, good localization and minimal response. By localizing edges we could determine the position of "platform" on which the object is supposed to be placed. [30]

Another option would be using Hough Line Transform¹⁶ algorithm 6.4, which is as well available in OpenCV library. Hough Line Transform is a transform used to detect straight lines, to apply this transform we must pre-process the image with edge detection. [31]

Using computer vision is also the most complex of all solutions because every single spot and model would require dedicated setup. Computer vision also faces problem with lighting conditions, under different weather conditions such as clouds or sunshine the perceived image would be alternated and therefore the software might not be able to render the model properly leaving the user without an alternative.

¹⁴OpenCV (<http://opencv.org>)

¹⁵Canny Edge Detector (http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html)

¹⁶Hough Line Transform (a)

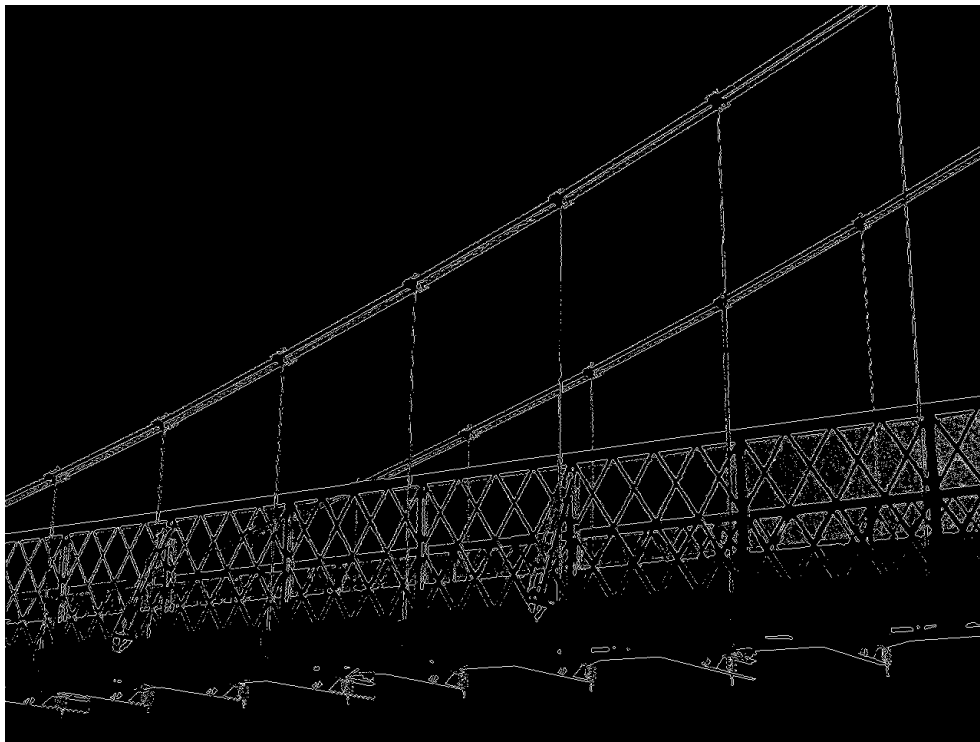


Figure 6.3: Application of Canny edge detector

Performance is also very critical, old devices such as iPhone 5 might not be able to analyze image fast enough, even though OpenCV takes advantage of multi-core environment iPhone 5 has limited computational power and would impact the resulting effect.

6.7 Static observation

Static observation uses device's gyroscope and accelerometer to determine devices altitude and this way adjust the position of model viewed on screen. Particular places would be picked from which the building is observable. iBeacons will be placed there to notify the application about the user's location. However the user would have to remain stationary and if he wanted to see the building from different angle or distance he would have to find another predefined spot. There are certain deviations such as height of the observer and length of his arm however these can be neglected due to higher distance from the object. Resulting effect is satisfying for our needs. Sensors in iOS devices provide very accurate results.

This is the approach that will be implemented because it satisfies our requirements even though it does not allow user to move around. Multiple



Figure 6.4: Application of hough lines transformation

view points will be set up to allow observation of the building from different angles and distances.

Technologies

In the following chapter technical choices will be explained. For the demo purposes the user will place and rotate the model according to his needs because otherwise we would need to predefine the position.

iOS 8.3¹⁷ is the target platform and iPhone 6 plus¹⁸ will serve as the test device. Swift and SceneKit will be used.

7.1 iOS 8 and SceneKit

iOS 8 was introduced in half of 2014 at WWDC by Apple. iOS 8 was the biggest OS release for developers bringing hundreds of changes to cocoa touch framework APIs and even new programming language Swift¹⁹. [32] iOS 8 also introduced SceneKit²⁰ which was until then only available for OSX. SceneKit is the key feature in iOS 8 that will be used for visualization of the 3D building model.

SceneKit is an Objective-C framework for building apps and games that use 3D graphics, combining a high-performance rendering engine with a high-level, descriptive API. SceneKit supports the import, manipulation, and rendering of 3D assets. [33] Scene Kit sits below Cocoa and above OpenGL in the graphics architecture [34]. SceneKit integrates with Image Kit and Core Animation, therefore programmer does not need advanced 3D graphical programming skills. However it provides the user with lower control over the rendering process, only single model will be rendered on the screen therefore low level optimization is not necessary.

¹⁷iOS 8.3 (<https://www.apple.com/ios/>)

¹⁸iPhone 6 plus (<https://www.apple.com/iphone-6/>)

¹⁹Swift (<https://developer.apple.com/swift/>)

²⁰SceneKit (https://developer.apple.com/library/mac/documentation/SceneKit/Reference/SceneKit_Framework/index.html)

SceneKit is also capable of displaying Collada²¹ 3D asset through simple API.

7.2 iPhone 6 plus

The new iPhone 6 plus comes with Apple A8 chip²² and Apple M8 chip²³ which are highly usable in augmented reality applications. Apple A8 chip is 64-bit ARM based system on a chip, Apple states that it has 25% more CPU performance and 50% more graphics performance while drawing only 50% of the power compared to its predecessor, the Apple A7. [35] M8 are motion co-processors used by Apple in their mobile devices. Their function is to collect sensor data from integrated accelerometers, gyroscopes and compasses and offload the collecting and processing of sensor data from the main central processing unit (CPU). [36]

7.3 Swift

Swift is one of the newest programming languages. It was only introduced in 2014 at WWDC however it quickly gained momentum from community and is the fastest growing language in programming community. Swift is designed to work with Cocoa and Cocoa Touch frameworks and is easily integrate-able into an existing app. Apple claims that swift is twice as fast as Objective-C and since to achieve the best augmented reality effect 60 frames per second are needed, fast programming language is needed. [37]

7.4 Core Motion Framework

Core Motion Framework²⁴ is Apple's framework for interaction with device's hardware sensors. The framework offers both raw and processed data. For devices with built-in gyroscope, Core Motion Framework provides us with API to retrieve the raw gyro data as well as processed data reflecting the attitude and rotation rates of the device. [38]

²¹Collada (<https://collada.org>)

²²Apple A8 chip (http://en.wikipedia.org/wiki/Apple_A8)

²³Apple M8 chip (http://en.wikipedia.org/wiki/Apple_M7)

²⁴Core Motion Framework (https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CoreMotion_Reference/)

Implementation

The implemented solution for augmented reality will combine frameworks provided by Apple and is included in every iOS device with iOS 8 and above. In this chapter implementation details and user interface will be discussed.

8.1 User Interface

For creation of user interface we are going to use two frameworks provided by Apple and one open source library from github.

8.1.1 Cocoa Touch

Cocoa Touch²⁵ is iOS framework for building user interfaces. Cocoa Touch is based on the Mac OS X Cocoa API and is primarily written in Objective-C language, Cocoa Touch framework is accessible from Swift. Cocoa Touch strongly follows Model-View-Controller (MVC) software architecture. Cocoa Touch contains set of graphical control elements among one of them is UISlider which will allow user to adjust the object's rotation and position. [39]

8.1.2 AV Foundation Framework

AV Foundation²⁶ is part of Cocoa Touch and it provides essential services for working with audiovisual media on iOS. AV Foundation has modern Objective-C interface that allows easy access to camera. Objective-C is accessible from Swift therefore there won't be any problem using AV Foundation with Swift. AV Foundation allows us to retrieve image from devices back camera in desired resolution and create "preview layer" which we will place over the entire screen. However AV Foundation's AVCaptureSession object provides us video with aspect ratio 4:3, iPhone 6 plus display has aspect ratio 16:9. We will simply

²⁵Cocoa Touch (http://en.wikipedia.org/wiki/Cocoa_Touch)

²⁶AV Foundation (<https://developer.apple.com/av-foundation/>)

stretch the image to fill out the entire screen. The image from camera this way looks only slightly alternated and is hard for user to tell that the image has been modified. [40]

8.1.3 Snap

Snap²⁷ is a autolayout framework which only provides easier and less verbose syntax for Apple's autolayout.

8.2 Accessing motion sensors

For augmented reality as an input we will use Core Motion Framework which has number of options which are examined in following sections.

8.2.1 Core Motion Framework

Core Motion allows a developer to observe and respond to orientation of an iOS device by inspecting the raw and processed data from a combination of built-sensors such as accelerometer, gyroscope and magnetometer.

Accelerometer and gyroscope data are presented in terms of three axes that run through the iOS device. For an iPhone in portrait orientation the X-axis runs through the device from left (negative values) to right (positive values), the Y-axis through the device from bottom (-) to top (+), and the Z-axis runs perpendicularly through the screen from the back (-) to the front (+). 8.1

To sensors we have to instantiate CoreMotionManager. CoreMotionManager can both, "pull" data and "push" data. To pull motion data we can access the current status of any sensor as property of CoreMotionManager. To receive or get data pushed we give CoreMotionManager a block/closure that receives the updates at specific interval. Only data needed is the latest data and do not need all the updates therefore periodically sampling the motion data is preferred. By setting timer to update the position of object 60 times per second the desired quality and effect should be achieved

Accelerometer returns data in CMAccelerometerData which includes x,y, and z values. Each of these shows the amount of acceleration in Gs for given axis. To better explain it, if you placed your device flat on table with screen facing upwards, it would read (0,0,1). This does tell us something about phone's position but only towards the ground.

Accelerometer itself does not provide enough data to determine phones rotation and therefore its point of view. This is where Sensor fusion comes handy. Sensor fusion is the combination of sensory data, such as accelerometer and gyroscope to provide much more accurate and useful results. Sensor fusion

²⁷Snap (<https://github.com/Masonry/Snap>)

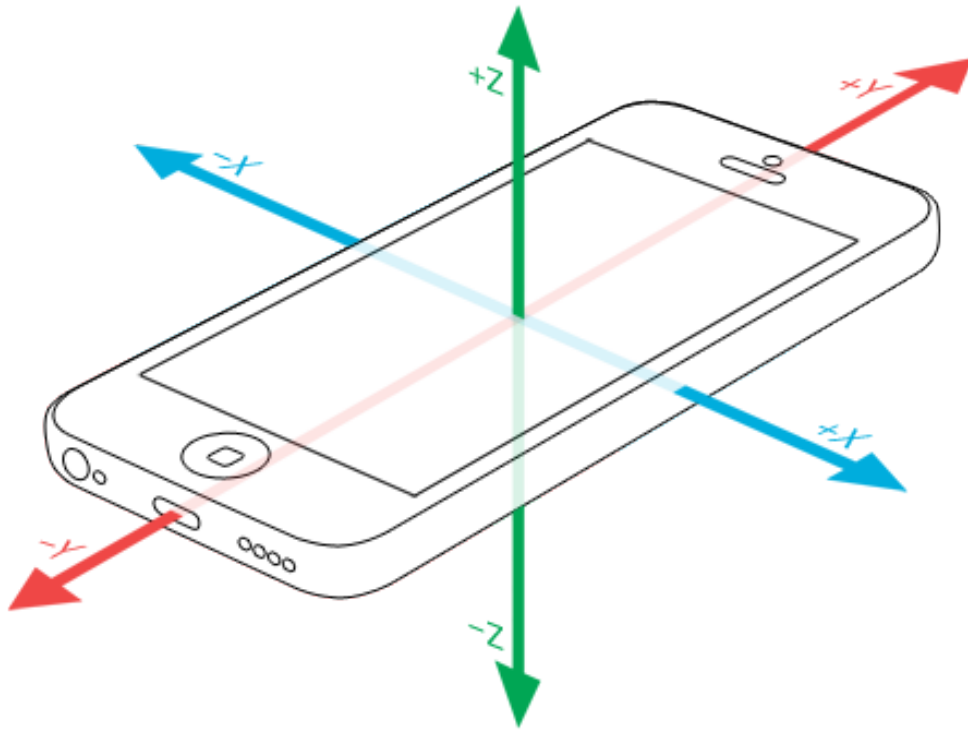


Figure 8.1: Three axes that run through the iOS device

algorithms are already built into Core Motion Framework and sensors will be accessed through this API.

To get device's attitude we also need to define reference frame, there are four options.

1. **CMAttitudeReferenceFrameXArbitraryZVertical** describes a device laying flat (vertical Z-axis) with an "arbitrary" X-axis. In practice, the X-axis is fixed to the orientation of the device when you first start device motion updates.
2. **CMAttitudeReferenceFrameXArbitraryCorrectedZVertical** is essentially the same but uses the magnetometer to correct possible variation in the gyroscope's measurement over time. Using the magnetometer adds a CPU (and therefore battery) cost.
3. **CMAttitudeReferenceFrameXMagneticNorthZVertical** describes a device laying flat, with its X-axis (i.e., the right side of the device) pointed toward magnetic north. This setting may require your user to perform that figure-eight motion with their device to calibrate the magnetometer.

4. **CMAttitudeReferenceFrameXTrueNorthZVertical** is the same as the last, but this adjusts for the magnetic/true north discrepancy and therefore requires location data in addition to the magnetometer.

For our purposes, the **CMAttitudeReferenceFrameXArbitraryCorrectedZVertical** will be satisfactory. There are three attitude representations that we can choose from.

1. **Euler Angles** are the most readily understood among other representations, as they simply describe rotation around each of the axes. Pitch is rotation around the X-axis, increasing as the device tilts toward you, decreasing as it tilts away; roll is rotation around the Y-axis, decreasing as the device rotates to the left, increasing to the right; and yaw is rotation around the (vertical) Z-axis, decreasing clockwise, increasing counter-clockwise. Each of these values follow right hand rule.
2. **Rotation matrix** is often use to express the orientation of a camera, or a spacecraft relative to a reference axes-set. Rotation matrix expresses numerically three direction vectors in world coordinates, they together comprise the columns of the rotation matrix R that transforms directions into equivalent directions expressed in platform-local coordinates. Rotation matrix is extensively used in computer graphics. [41]
3. **Quaternion**, compared to Euler angles are simpler to compose and avoid gimbal lock. Compared to rotation matrices they are more numerically stable and are more efficient since rotation matrix is composed of 9 numbers and quaternion has only 4. [42]

Rotation matrix is our choice, Apple recommends using rotation matrix because it can be easily handed to 3D rendering engines such as OpenGL or in our case SceneKit. Rotation matrix from MotionManager defines the attitude in matrix of dimension 3×3 , however SceneKit uses transformation matrix is of dimensions 4×4 . Also rotation matrix can be multiplied by another rotation matrix which applies additional transformation which allows easy manipulation of the scene. [43]

The Rotation matrix returned by MotionManager 3×3 matrix

$$\begin{pmatrix} m11 & m12 & m13 \\ m21 & m22 & m23 \\ m31 & m32 & m33 \end{pmatrix}$$

The desired 4×4 matrix that SceneKit uses to make the necessary transformation which is applied to the scene.

$$\begin{pmatrix} m11 & m12 & m13 & 0 \\ m21 & m22 & m23 & 0 \\ m31 & m32 & m33 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Following code handles the transformation

```
func transformMatrix3ToMatrix4(inout mIn: CMRotationMatrix)
-> SCNMatrix4 {
    var mOut = SCNMatrix4Identity

    mOut.m11 = Float(mIn.m11)
    mOut.m12 = Float(mIn.m12)
    mOut.m13 = Float(mIn.m13)
    mOut.m14 = 0.0

    mOut.m21 = Float(mIn.m21)
    mOut.m22 = Float(mIn.m22)
    mOut.m23 = Float(mIn.m23)
    mOut.m24 = 0.0

    mOut.m31 = Float(mIn.m31)
    mOut.m32 = Float(mIn.m32)
    mOut.m33 = Float(mIn.m33)
    mOut.m34 = 0.0

    mOut.m41 = 0.0
    mOut.m42 = 0.0
    mOut.m43 = 0.0
    mOut.m44 = 1.0

    return mOut
}
```

8.3 3D scene

SceneKit has object `SCNScene` which we initialize with our Collada (DAE) object. In scene contains projected object and a reference to it is kept because later transformations may be applied to it. Initially the object will be positioned at certain distance in front of the camera without any additional rotation applied. A camera has to be added to the scene programatically, it will be positioned at `[0,0,0]` coordinates also without any rotation. The scene only requires these two objects

8.4 Combining SceneKit with CoreMotion

The key part of entire implementation is combining SceneKit and CoreMotion together. The app asks the system 60 times per second what is attitude of



Figure 8.2: 3D model used in demo application

the phone and this exact transformation then is applied to the camera in scene. However there is a little catch to this because raw rotation matrix from MotionManager only transformed to 4x4 matrix and fed to SceneKit would cause to render the model under the user and not in front of him. For this very reason rotation matrix which will rotate camera by 90 degrees has to be applied to matrix from MotionManager to adjust it.

Conclusion

This thesis was rewarding and enriching for me. Augmented reality is promising field of computer science and worth of discovering. I would like to make an evaluation and suggestions for further development of implemented solution.

Evaluation

In the thesis modern usages of augmented reality in mobile devices were analyzed as well as number of popular applications and their underlying frameworks. It is beyond the scope of this work to look at every single augmented reality approach however the most common ones were researched and evaluated in relation to the goal. Every approach is followed by a list of pros and cons, along with explanation of its inadequacy.

Researched frameworks did not provide desired functionality therefore the decision was made to try out a new approach based on using phone's motion sensors as the input media for augmented reality and transfer the attitude changes onto 3D scene in order to keep the scene in sync with phone's rotation. This approach differs from most other augmented reality applications because it doesn't use computer vision or phone's location. Demo application was designed and implemented to show this approach, all parts of the applications are explained in detail.

The application combines two Apple frameworks and uses them as input and output medium. SceneKit, which was announced for iOS platform at WWDC in 2014, serves as render engine to provide user with extended information over the real life scene. The input media is phone's motion sensors, these sensors are accessed via CoreMotion framework. Demo application is only available on iOS platform. Similar solution could be implemented on other platforms or alternatively using one of the multi-platform solutions which let you interact with motion sensors through unified API.

Whole augmented reality functionality is contained within one view controller class which can be easily included and modified for further uses. The application is designed as a showcase of the approach and is not intended to be published in the App Store.

The demo application uses different approach than the researched frameworks however it meets the criteria that were defined. The research of most popular augmented reality frameworks for mobile devices revealed their lack of usage of motion sensors and they rely mostly on markers and location services. The demo application does not provide user with as persuasive result as marker based augmented reality however it is suited for different use cases such as visualizing objects in long distance where markers wouldn't be visible.

Further development

Static observation approach could be enhanced by using computer vision. Application could use static observation in case computer vision failed so it would provide at least some kind of visual experience the the user. Also having other source of users location would allow user to move and use the implemented solution for orientation only. This is the next milestone.

Bibliography

- [1] MarketsandMarkets. Augmented Reality and Virtual Reality Market worth \$1.06 Billion by 2018 [online]. 2014, [cit. 2015-05-05]. Available from: <http://www.marketwatch.com/story/stater-bros-charities-announces-2013-award-recipients-2013-03-13>
- [2] Ivan Sutherland. A head-mounted three dimensional display [online]. 1968, [cit. 2015-05-03]. Available from: http://90.146.8.18/en/archiv_files/19902/E1990b_123.pdf
- [3] Sung, D. The history of augmented reality [online]. 2011, [cit. 2015-05-03]. Available from: <http://www.pocket-lint.com/news/108888-the-history-of-augmented-reality>
- [4] Caudell, T.; Mizell, D. Augmented reality: an application of heads-up display technology to manual manufacturing processes. 1992.
- [5] Milgram, P.; Kishino, F. A TAXONOMY OF MIXED REALITY VISUAL DISPLAYS [online]. 1994, [cit. 2015-05-03]. Available from: http://etclab.mie.utoronto.ca/people/paul_dir/IEICE94/ieice.html
- [6] Henrysson, A. *Bringing Augmented Reality to Mobile Phones* [online]. ISBN 978-91-85895-43-4, [cit. 2015-05-03]. Available from: <http://www.diva-portal.se/smash/get/diva2:16967/FULLTEXT01.pdf>
- [7] Azuma, R. T. A Survey of Augmented Reality [online]. 1997, [cit. 2015-05-03]. Available from: <http://www.cs.unc.edu/~azuma/ARpresence.pdf>
- [8] Wagner, D. Handheld Augmented Reality [online]. [cit. 2015-05-03]. Available from: http://studierstube.icg.tugraz.at/thesis/Wagner_PhDthesis_final.pdf

BIBLIOGRAPHY

- [9] Hughes, C. E.; Stapleton, C. B.; Hughes, D. E.; et al. Mixed reality in education, entertainment, and training. volume 25, 2005, ISSN 0272-1716. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1528429
- [10] van Krevelen, D.; Poelman, R. A Survey of Augmented Reality Technologies, Applications and Limitations [online]. 2010, [cit. 2015-05-03]. Available from: <http://kjcomps.6te.net/upload/paper1%20.pdf>
- [11] Stricker, D.; Gleue, T.; Daehne, P.; et al. Archeoguide: An Augmented Reality Guide for Archaeological Sites [online]. 2010, [cit. 2015-05-03]. Available from: http://webserver2.tecgraf.puc-rio.br/~mgattass/ra/ref/RA_Ruinas/01028726.pdf
- [12] Fiorentino, M.; de Amicis, R.; Monno, G.; et al. Spacedesign: A Mixed Reality Workspace for Aesthetic Industrial Design [online]. 2002. Available from: http://sketch.inesc-id.pt/publications/fu-83_final.pdf
- [13] Schowengerdt, B. T.; Seibela, E. J.; Kelly, J. P.; et al. Binocular retinal scanning laser display with integrated focus cues for ocular accommodation [online]. 2003, [cit. 2015-05-03]. Available from: <ftp://ftp.hitl.washington.edu/pub/publications/r-2006-35/r-2006-35.pdf>
- [14] Kasai, I.; Tanijiri, Y.; Endo, T.; et al. A Forgettable Near Eye Display [online]. [cit. 2015-05-03]. Available from: <http://monet.cs.columbia.edu/courses/mobwear/resources/kasai-iswc00.pdf>
- [15] Hayward, V.; Astley, O. R.; Cruz-Hernandez, M.; et al. Haptic interfaces and devices [online]. [cit. 2015-05-03]. Available from: <http://www.roblesdelatorre.com/gabriel/VH-OA-MC-DG-GR-04.pdf>
- [16] Billingham, M.; Kato, H. Collaborative Mixed Reality [online]. [cit. 2015-05-03]. Available from: <ftp://ftp.hitl.washington.edu/pub/publications/r-91-2/r-98-36/r-98-36.html>
- [17] Azuma, R. T. The Challenge of Making Augmented Reality Work Outdoors. 1999.
- [18] Tamura, H. Steady Steps and Giant Leap Toward Practical Mixed Reality Systems and Applications [online]. [cit. 2015-05-03]. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.619&rep=rep1&type=pdf>
- [19] Bahl, P.; Padmanabhan, V. N. RADAR: An In-Building RF-based User Location and Tracking System. [cit. 2015-05-03]. Available from: <http://research.microsoft.com/en-us/people/padmanab/infocom2000.pdf>

-
- [20] Liu, J.; Pan, Z.; Li, X. An Accelerometer-Based Gesture Recognition Algorithm and its Application for 3D Interaction [online]. [cit. 2015-05-03]. Available from: <http://www.doiserbia.nb.rs/img/doi/1820-0214/2010/1820-02141001177L.pdf>
- [21] imon Julier; King, R.; Colbert, B.; et al. The Software Architecture of a Real-Time Battlefield Visualization Virtual Environment [online]. [cit. 2015-05-03]. Available from: http://www.nrl.navy.mil/itd/imda/sites/www.nrl.navy.mil.itd.imda/files/pdfs/cp_VR99b.pdf
- [22] Russell, M. 11 Amazing Augmented Reality Ads [online]. 2012, [cit. 2015-05-05]. Available from: <http://www.businessinsider.com/11-amazing-augmented-reality-ads-2012-1?op=1>
- [23] Widder, B. BEST AUGMENTED REALITY APPS [online]. [cit. 2015-05-03]. Available from: <http://www.digitaltrends.com/mobile/best-augmented-reality-apps/>
- [24] GmbH, W. Wikitude [online]. [cit. 2015-05-03]. Available from: <https://www.wikitude.com>
- [25] Labs, N. Ingress (game) [online]. [cit. 2015-05-03]. Available from: <https://www.ingress.com>
- [26] of Business Analysis., I. I. A Guide to the Business Analysis Body of Knowledge. 2009.
- [27] Authority, A. M. S. Differential Global Positioning System. [cit. 2015-05-03]. Available from: <https://www.amsa.gov.au/navigation/services/dgps/>
- [28] Marimon, D.; Adamek, T.; Bonnin, A.; et al. Enhancing global positioning by image recognition [online]. [cit. 2015-05-03]. Available from: <http://infoscience.epfl.ch/record/183627/files/Marimon11.pdf>
- [29] Cavallini, A. iBeacons Bible [online]. [cit. 2015-05-03]. Available from: <https://meetingofideas.files.wordpress.com/2013/12/ibeacons-bible-1-0.pdf>
- [30] community., O. Canny Edge Detector [online]. [cit. 2015-05-03]. Available from: http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html
- [31] community., O. Hough Line Transform [online]. [cit. 2015-05-03]. Available from: http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html

BIBLIOGRAPHY

- [32] Inc., A. Apple Unveils iOS 8, the Biggest Release Since the Launch of the App Store. [cit. 2015-05-03]. Available from: <http://www.apple.com/pr/library/2014/06/02Apple-Unveils-iOS-8-the-Biggest-Release-Since-the-Launch-of-the-App-Store.html>
- [33] Inc., A. SceneKit Framework Reference [online]. [cit. 2015-05-03]. Available from: https://developer.apple.com/library/mac/documentation/SceneKit/Reference/SceneKit_Framework/index.html
- [34] Inc., A. Introduction to Scene Kit [online]. [cit. 2015-05-03]. Available from: https://developer.apple.com/library/mac/documentation/3DDrawing/Conceptual/SceneKit_PG/Introduction/Introduction.html
- [35] Savov, V. iPhone 6 and iPhone 6 Plus have a new faster A8 processor [online]. 2014, [cit. 2015-05-03]. Available from: <http://www.theverge.com/2014/9/9/6127111/iphone-6-and-iphone-6-plus-have-a-new-faster-a8-processor>
- [36] Inc., A. Apple M7 [online]. [cit. 2015-05-03]. Available from: <https://www.apple.com/iphone-6/technology/>
- [37] Inc., A. Swift [online]. [cit. 2015-05-03]. Available from: <https://developer.apple.com/swift/>
- [38] Inc., A. Core Motion Framework Reference [online]. [cit. 2015-05-03]. Available from: https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CoreMotion_Reference/
- [39] Inc., A. AV Foundation [online]. [cit. 2015-05-03]. Available from: <https://developer.apple.com/documentation/AVFoundation/Reference/AVFoundationFramework/index.html>http://en.wikipedia.org/wiki/Cocoa_Touch
- [40] Inc., A. AV Foundation [online]. [cit. 2015-05-03]. Available from: <https://developer.apple.com/av-foundation/>
- [41] Weisstein, E. W. Rotation Matrix [online]. [cit. 2015-05-03]. Available from: <http://mathworld.wolfram.com/RotationMatrix.html>
- [42] Weisstein, E. W. Quaternion [online]. [cit. 2015-05-03]. Available from: <http://mathworld.wolfram.com/Quaternion.html>
- [43] Cook, N. CMDeviceMotion [online]. [cit. 2015-05-03]. Available from: <http://nshipster.com/cmdevicemotion/>

Acronyms

AR Augmented reality

GPS Global position system

WWDC World Wide Developer Conference

API Application programming interface

UUID Universally unique identifier

Contents of enclosed CD

	readme.txt	the file with CD contents description
	impl	source codes
	thesis	the directory of \LaTeX source codes of the thesis
	text	the thesis text directory
	└ thesis.pdf	the thesis text in PDF format