

Augmented Reality Applications for Industrial Robots

Björn Löfvendahl

February 20, 2014

Master's Thesis in Interaction Design, 30 credits

Supervisor at TFE-UmU: Kalle Prorok

Supervisor at ABB: Elina Vartiainen

Examiner: Håkan Gulliksson

UMEÅ UNIVERSITY
DEPARTMENT OF APPLIED PHYSICS AND ELECTRONICS
SE-901 87 UMEÅ
SWEDEN

Abstract

Augmented reality (AR) is a way of overlaying digital information onto a picture or a video feed and has been used in industrial contexts for more than 20 years. This Master's Thesis examines if AR can be used to help maintenance engineers set up and maintain robot environments by visualizing robot movement and safety zones.

The main result of the Master's Thesis is a prototype application for a tablet computer. The user points the tablet towards a robot filming it and the video feed is displayed on the screen. This video feed is augmented with a virtual zone displayed around the robot, illustrating the area where the robot is allowed to move. The application fetches the coordinates for the zone from the safety system SafeMove – a system designed by ABB to increase safety and allow closer human-robot collaboration.

The visualization of a SafeMove configuration is currently limited to an image of a two-dimensional coordinate system showing the zone as a set of different coordinates. This makes it difficult grasping the full layout of the three-dimensional zone. By using the application the user gets a better view of the layout, allowing the user to look at the robot from different sides and see the safety zone projected around the robot.

User tests show that people working with SafeMove could benefit from using the application to verify the configuration of SafeMove systems and the conclusion is that AR, if used right, greatly can improve robot interaction and maintenance.

Contents

1	Introduction	1
2	Background	3
2.1	The Company	3
2.1.1	ABB Corporate Research	3
2.1.2	ABB Robotics	3
2.2	Augmented Reality	3
3	Problem Description	5
3.1	Thesis Goal	5
3.2	Thesis Background	5
3.3	Requirements	5
3.4	Methods	6
3.4.1	Literature Study	6
3.4.2	Concept and Ideas	6
3.4.3	Prototype Development	6
3.4.4	User Evaluation	6
4	Technology Study	7
4.1	Industrial Augmented Reality	7
4.1.1	Different Areas of Industrial Augmented Reality	7
4.1.2	Augmented Reality in the Future	9
4.2	Software	10
4.2.1	Unity3D and MonoDeveloper	10
4.2.2	Vuforia and Metaio	10
4.2.3	RobotStudio and SafeMove	11
5	Development	13
5.1	Concept	13
5.2	The Prototype	14
5.2.1	Safety Zone	14
5.2.2	Tracking	14

5.2.3	Zone Adjustment	16
5.2.4	Occlusion Model	17
5.2.5	Calibrating the Markers	18
5.2.6	Connecting to the Robot	19
5.2.7	Interface	19
5.2.8	AR Testing	20
5.2.9	Development Challenges and Solutions	21
6	Results	25
6.1	The Prototype	25
6.1.1	User guidance	25
6.1.2	Interface	26
6.2	The Evaluation	28
7	Conclusion	29
7.1	Discussion	29
7.2	Limitations	30
7.3	Future Work	30
8	Acknowledgements	31
	References	33

List of Figures

4.1	Game engine Unity3D together with text editor MonoDeveloper.	11
5.1	The concept functions: safety zone, color condition and TCP trail.	14
5.2	SafeMove coordinate input interface.	15
5.3	Different forms of safety zones.	15
5.4	Three of Metaio's ID markers (fiducial markers).	16
5.5	The ABB IRB140 robot.	17
5.6	Wireframe with colored spheres for size adjustment.	17
5.7	The occlusion model turned on and off.	18
5.8	TCP trail.	19
5.9	The user interface (in opened state).	20
5.10	AR test flow.	20
5.11	The application-to-robot connection.	22
5.12	Child-parent relationship of the robot model's parts.	23
5.13	The markers should have a white border to improve tracking.	24
6.1	User holding the tablet.	25
6.2	Calibration of one of the markers.	26
6.3	The application with settings menu open.	27

Chapter 1

Introduction

Augmented reality (AR) is a way of overlaying digital information onto a picture or a video feed and has been used in industrial contexts for more than 20 years. It can e.g. be used during the development phase of a project blurring the line between the real and the virtual world or as an interactive aid for workers, guiding them through their tasks using a head-mounted display.

ABB is one of the world's largest companies offering solutions for power technology and industrial automation. ABB is also a leading developer of high performance industrial robots. They wanted to investigate how AR could be used to visualize information about industrial robots and how this information could benefit engineers working with maintenance.

This thesis work explores the AR field and introduces conceptual models of how AR can be used to visualize different types of data. The thesis work includes developing a functional prototype of a tablet application. By using AR this application displays a safety zone around a connected robot informing the user of in which area the robot is allowed to move. The application was tested on engineers used to working with ABB robots and the result showed that a visual aid such as this application really could make the calibration of safety zones easier.

The report begins by giving background information about the project, which is followed by a description of the given problem. Next comes a study of how AR is used in the industry today and a summary of the different technologies used for the development of the application. This development is described along with the results of the user evaluations. Lastly the thesis work's findings are discussed and proposals of future work are described.

Chapter 2

Background

2.1 The Company

This master's thesis has been done at and together with ABB.

ABB is one of the leading companies in power technologies and industrial robots. They have more than 150,000 employees and operate in around 150 different countries with headquarters in Switzerland. ABB was formed when the Swedish company ASEA (Allmänna Svenska Elektriska Aktiebolaget) and the Swiss company BBC (Brown, Boveri & Cie) merged in 1988.

2.1.1 ABB Corporate Research

ABB Corporate Research (CRC) consists of seven different R&D¹ centers around the world. They research and develop technologies for ABB's five main divisions. The Swedish center is located in Västerås and has around 250 employees.

2.1.2 ABB Robotics

ABB Robotics develops and manufactures industrial robots. In Sweden they are located in Västerås and Göteborg and have around 600 employees. They have robot factories in Västerås and in China which together produce around 20,000 robots every year.

2.2 Augmented Reality

Augmented Reality (AR) is a technology where the reality is augmented – enhanced – with different types of virtual information [5],[3]. This information can be e.g. 3D models, text and images.

With AR the user see this information as an overlay on top of the real world. Unlike virtual reality where the user it totally immersed in the virtual world and cannot see anything but the virtual environment [3].

To be able to place the overlay in the correct position the AR software can use different types of techniques. Some of these techniques are marker tracking, image recognition and the use of embedded sensors [5].

The name augmented reality was coined by Boeing scientists when they were working with simplifying the process of assembling cables on airplanes back in 1990 [5].

¹Research and Development

Chapter 3

Problem Description

3.1 Thesis Goal

The main goal of this master's thesis was to design and implement an application for a tablet computer, an application that would be used by maintenance engineers to help them with their daily tasks. In addition it should use AR to visualize information about industrial robots.

The thesis had three subgoals/phases:

- Develop a concept for the application.
- Implement a working prototype for a tablet computer.
- Evaluate the prototype with user tests.

3.2 Thesis Background

CRC has used AR in several projects in the past. Together with ABB Robotics they were interested in using AR to visualize information about industrial robots in different ways. ABB Robotics believed that maintenance engineers could benefit from using AR in their daily work and wanted to explore the AR area to see in which ways this technology could be used.

3.3 Requirements

The setup for the thesis project was very open and there were no specific requirement on what type of information that should be visualized or which way the technology should be used. However, the following aspects should be taken into consideration.

- The main requirement was that the application should use AR. In which way, however, was not specified.
- The application should be a working prototype and not only a concept, so it could be tested in a productive way.

- ABB Robotics uses Windows 8 as their software platform so they wanted the application to work in this environment.
- The end-users were maintenance engineers working with industrial robots so the application should be made to cater to some of their needs.

3.4 Methods

The project was divided into four different parts described below:

3.4.1 Literature Study

In order to learn more about the subject the project began with a literature study. This study did not only examine different AR technologies but also ABB robots, software development and other related areas.

3.4.2 Concept and Ideas

In the second phase research on how AR is used in the industry were performed. It also involved building a concept showing different ideas of how AR could be used within this project. This concept was then presented to ABB Robotics in order to decide which functions the final application should have.

3.4.3 Prototype Development

Next was the development phase, which was the main part of the project. In this phase the tablet application was developed and different AR techniques were tried and tested.

3.4.4 User Evaluation

The final phase of the thesis was a user evaluation where the developed application was tested on a number of people working with industrial robots.

Chapter 4

Technology Study

This chapter presents a study of some of the various ways AR is, and can be, used in the industry today. It also describes the different software, frameworks and applications used during the development of the application.

4.1 Industrial Augmented Reality

AR has a wide field of applications but is mostly used in advertisements and to promote new products [5]. This study will, however, focus on the industrial uses for AR, sometimes called industrial augmented reality (IAR). A number of different areas of IAR has been found and are presented below.

4.1.1 Different Areas of Industrial Augmented Reality

Design and Development

Design and development is an area where AR could have a big impact and lead to big savings [12]. When designing and developing for different industrial purposes it can be really expensive to create models and test different design solutions, it can also take a lot of time creating these models [4].

Testing vehicle ergonomics is one field where the use of AR could be beneficial. Together with Volkswagen Berssenbrügge et al. developed an AR-based test bench which could be used to simulate new car models [4]. The test bench consisted of a real car without ceiling and dashboard. While driving the car the driver wears a head-mounted display (HMD) with integrated cameras, which augments the vision of the car and the track with virtual models. This way new design concepts can be evaluated without changing any hardware. Using a combination of real world and virtual models also gives a more life-like scenario than using only virtual models.

In another automotive related project AR is used to compare real car parts with their corresponding construction data [10]. This is done to make sure that the parts has been produced with enough precision and corresponds to the correct car model. Virtual CAD data is superimposed onto the object as transparent models enabling an easy way to see if there is any mismatch.

In a related industry – the aerospace industry – there is also many uses for AR. Regenbrecht, Baratoff and Wilke gives two examples of how they have used AR to improve the

design process [12].

The first example involves airplane cabin design. While it is relatively easy displaying some parts of the cabin environment, e.g. seat placement, in a digital mock-up it can be much more difficult displaying other types of information. One example of this is simulations of different physical properties such as temperature and pressure. Regenbrecht, Baratoff and Wilke constructed a prototype that displayed a 3D visualization of computational fluid dynamics data as an overlay onto the real world. It could be used to highlight areas with high pressure values in airplane cabins.

In their second example, Regenbrecht, Baratoff and Wilke describe how the early development of cockpit controls could work. By wearing an HMD and placing magnetic markers on a whiteboard the designers can test different settings. Each marker corresponds to a different gauge or set of controls and many designers can work together.

Another vehicle industry that uses AR is the naval industry. One example is a tool for planning the design of pipe layout in large ships [11]. There can often be problems with discrepancies between construction data and the real ship and the pipe layout might have to be modified. Today this is handled by operators using wire models. The operators uses these special wires to check if the pipe layout is correct. If not the wires are bent to fit. The wires are then measured and the real pipes are bent according to the data. This can be time consuming and instead an AR tool can be used. With the tool pipes can be virtually visualized using a tablet. If the pipes do not fit they can be modified using the tool. The modified layout is saved and can be directly transferred to the pipe production.

One example related to this thesis work describes how AR can be used to plan the trajectories of industrial robots. The authors describe how they could optimize the robot's movement path by using AR to visualize the path before actually moving the real robot. [8]

Learning and Training

Another area where AR can be really helpful is learning and training. When new workers are being trained it can be difficult to mediate all types of knowledge. Experienced workers' accumulated knowledge cannot always be found in books and manuals but are gained from, often, many years of experience and operation. Weibel et al. [15] writes about how AR can be used to pass on skills and knowledge from an experienced worker to a new. They have developed a training platform where the trainee uses a tablet as a training aid for learning to perform maintenance of large industrial machinery. The trainee uses the tablet to film the machinery and different aids are provided together with the video feed. These aids can be a media files such as pictures or movies providing the trainee with additional information or superimposed models showing how different tasks should be performed, e.g. rotating a lever in different directions. If the trainee has questions he can contact the remote trainer with the tablet.

Together with this tablet Weibel et al. produced a bracelet which can give the wearer haptic hints. The bracelet has six vibration actuators, which can be used to aid the trainee. Since these actuators can be controlled individually different sensations can be generated, e.g. "clockwise rotation". This way the trainee can receive hints when performing sub-tasks such as "rotate this switch in counter clockwise direction". It can also provide feedback if the trainee makes a mistake.

The authors discovered that one potential problem with this approach is that the user might become too dependent on the training tool and do not learn the tasks properly. A way of handling this is to gradually decrease the amount of help provided.

Regenbrecht, Baratoff and Wilke also explored how AR can be used for learning and

training. They stated that "One main advantage of using AR technology for training compared with VR or other multimedia technologies is the possibility of on-site experiences for the trainee." [12] They illustrated this with a scenario where the trainee were learning to drive a car. When driving a car there are many potentially dangerous situations that can occur and not all of them can be fully understood or mastered by just reading about them. Not all of them can be fully tested on the trainee either. In the described scenario the trainee wears an HMD while driving a car in an enclosed environment. Through the HMD the trainee can experience situations that otherwise could not have been tested, e.g. a child jumping out in front of the car.

In an area more closely related to IAR this set up could be used on, e.g., maintenance engineers to test error handling and problem solving that otherwise would be difficult to test.

Task Support and Collaboration

Task support and collaboration is an area where AR really could make a difference.

One early example of task support was Caudell and Mizell's use of an HMD when manufacturing a Boeing airplane. [6] The worker wears the HMD and get visual help when placing wires. It was Caudell and Mizell who coined the term "Augmented Reality" back in 1990.

Regenbrecht, Baratoff and Wilke describes how AR can be used to aid an industrial worker in various tasks such as fuse placement and item picking.[12] In one of their examples a worker manufacturing a truck cockpit is placing various types of fuses (of different color) on a board. There are many different models of trucks so the boards look very different. The schematics used are black and white prints with different numbers indicating different types of fuses. Filming the board and adding an overlay showing where the various types of fuses should be placed aid the workers in their task.

The real estate and building sector is another area where AR could be useful. One paper suggests that AR can be used to for interactive presentations at the constructions site, e.g. showing clients how a building could look after renovation. [16]

Stutzman et al. describes another way of how AR could improve collaboration. They suggest that if workers were equipped with a handheld computer with a camera and various sensors "plant efficiency can be improved by providing a richer method of communication". [13] Supervisors in a control room could easier direct the workers' actions and get a better overview of work status.

Another relatively early example of how AR could be used in a collaborative way is the augmented reality conferencing system presented by Kato and Billinghurst.[9] With their system the users get access to a virtual shared whiteboard where they can collaboratively interact with both 2D and 3D data.

4.1.2 Augmented Reality in the Future

Augmented reality has been around since the 1960s but it was not until the late 1990s when it became a distinct field of research.[14] Since then computers have become much more powerful and computer graphics a lot more realistic but apart from that the AR field has not had any mayor breakthroughs – until now that is.

Before, AR was a technology only explored by researchers – and to some extent the industry – but with the introduction of the smartphones the general public has for the first time a chance to use this technology in their daily life. For many AR applications the accuracy of the digital overlay is of great importance and with the possibility of a more

precise localization using GPS and other technologies the application areas has greatly improved. [7]

The evolution of AR technology has gone from stationary computers to bulky laptops to smaller hand-held devices to small wearable computers such as the Google Glass¹. The future of AR will most likely be these wearable computers where information can be accessed without the obstruction of one's hands. The phenomena of an HMD is nothing new but it's not until recently that computers has gotten small enough to be conveniently wearable, as well as being available to the general public. [7]

According to Alkhamisi and Muhammad the total revenues of mobile AR applications will increase from \$0.1 million 2012 to \$5.2 millions 2017. [2] In many ways AR is still in its infancy and we have yet to see its full potential.

4.2 Software

In this section the software used in the design and implementation processes are presented.

4.2.1 Unity3D and MonoDeveloper

Unity3D is a multi platform 3D game engine. It uses a "wysiwyg" (*what you see is what you get*) interface which means that the user is not only working with code but also models and objects which can be seen and manipulated in a 3D environment. In Unity3D it is possible to create simple models such as cubes and spheres or import more complex models – e.g. a robot – from 3D model software such as Maya, Rhino or, as in this case, RobotStudio. These objects, called gameobjects, are placed in the 3D world and can be given functions and behaviors by attaching scripts to them. The scripts can be written in C#, JavaScript and Boo². To add texture and color to the models different materials can be used.

Unity3D provides a "play mode" which can be used to test one's application without having to build it and export it to the targeted platform. This is very helpful and time saving but the play mode does not behave *exactly* as the real built application, which can lead to errors that are difficult to find. One example was that some objects were visible in play mode but not in the built version.

Usually Unity3D is used for developing games but it can be used to develop other types of applications too and CRC has used it in several projects before. It also has a large community and is easy to work with if one is not a very accustomed programmer. These were the main reasons why Unity3D was chosen as development tool.

There were some thoughts of using Visual Studio (VS) instead of Unity3D but it turned out that in order to use Metaio with VS one had to use C++. Since the author does not know C++ it would be too time consuming to learn it for this project. Therefore the idea was dropped.

MonoDeveloper is a text editor included in the Unity3D package. This application was used for code implementing and editing.

4.2.2 Vuforia and Metaio

Vuforia and Metaio are two different frameworks for working with AR. Vuforia was used in the concept and Metaio in the prototype. They can be used together with a variety of different programming languages and they also each have a plugin for Unity3D. With both

¹<http://www.google.com/glass>, accessed 2013-01-10

²<http://boo.codehaus.org>, accessed 2013-12-15

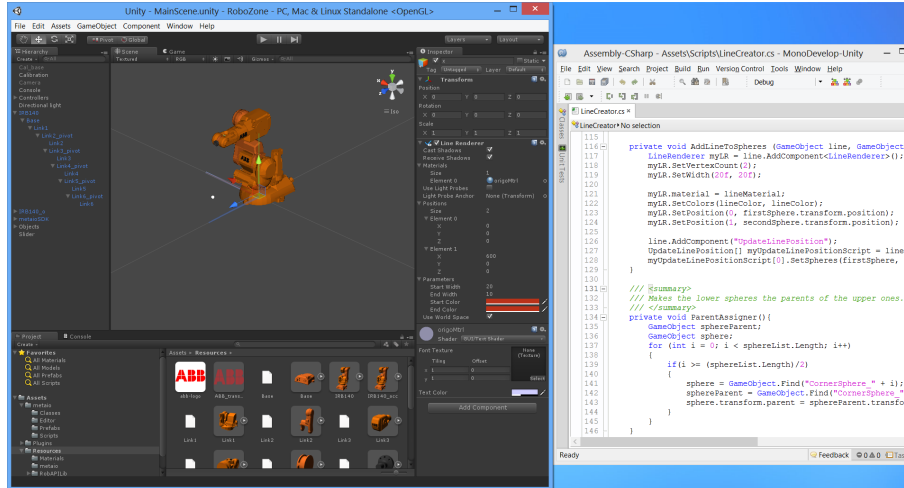


Figure 4.1: Game engine Unity3D together with text editor MonoDeveloper.

frameworks it is possible to add AR functionality to applications. They offer various form of image and marker recognition and can be used to superimpose things like virtual models, text and images on video feeds of real world objects.

4.2.3 RobotStudio and SafeMove

ABB's RobotStudio is used for robot simulation and offline programming. It can be connected to a robot controller to control a robot but it can also act as a virtual controller and simulate a robot. In the thesis project it was used to export the robot model used in the application and to test the connection between the application and the robot controller.

An industrial robot can be dangerous to handle since it is heavy and can move at a high speed. This is why there are a lot of regulations and safety measures to consider when working with robots. Often the robots are standing within a fenced area so that humans cannot get injured.

SafeMove is a safety add-on for ABB's industrial robots. In the SafeMove Application Manual it says: "SafeMove is a safety controller in the robot system. The purpose of the safety controller is to ensure a high safety level in the robot system using supervision functions that can stop the robot and monitoring functions that can set safe digital output signals"[1]. SafeMove restricts robot movement so that it is safe for humans to work next to them.

Chapter 5

Development

This chapter explains the different development phases of the project.

5.1 Concept

The first part of the project was used for gathering information about AR and how AR is and can be used in industrial environments. A list of possible ways of using AR together with robots was produced and this list was presented on a meeting with ABB Robotics. During this meeting there were discussions on what type of functions the project should focus on as well as who the end-user of the prototype would be. We decided to design an application for internal marketing purposes. This application would not need any real (and useful) functionality but instead have a lot of *cool* features to show AR's technical potential.

After this meeting a proof of concept was produced which implemented some of the different possible functions discussed at the meeting. The concept was an iPad application produced with the game engine Unity3D. A 3D model of an industrial robot was exported from ABB's RobotStudio and imported to Unity3D. To make the application visualize the model onto the real world the AR framework Vuforia was used.

The three concept functions implemented (see Figure 5.1) were:

- Displaying a safety zone around the robot.
- Displaying different colors on different parts of the robot to illustrate age and/or condition.
- Visualizing the position and path of the TCP¹

The concept was presented to ABB Robotics in order to decide which functions should be included in the prototype. At this point we decided that the end-user should be maintenance engineers and instead of *cool* features go for real functionality. We also decided to start implementing the safety zone and see if time allowed for implementation of more functions.

¹Tool center point – the tip of the robot

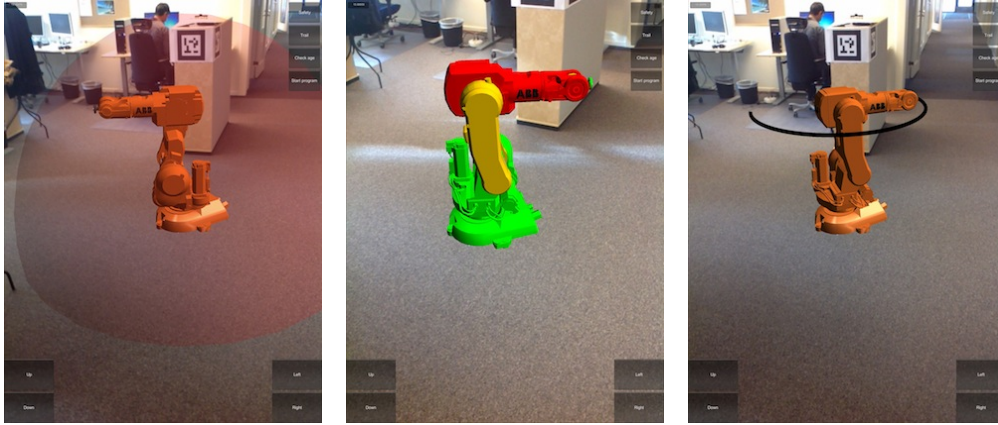


Figure 5.1: The concept functions: safety zone, color condition and TCP trail.

5.2 The Prototype

When the concept had been presented to ABB Robotics and the end-user had been decided, the work with the prototype began.

Since the ABB Robotics uses Windows 8, which Vuforia does not support, a new framework had to be found. Eventually the Metaio framework was chosen.² Unfortunately Metaio had a high learning curve and had (and still has) quite many bugs.³ Because of this it took some time to get even a simple project set up.

5.2.1 Safety Zone

The main function of the application would be to visualize how close to the robot one could get without being worried one might get hit by it – i.e., displaying a safety zone. ABB Robotics had the idea that the coordinates from the SafeMove – a safety system design by ABB – could be used. This system is used to make sure that robots only moves within certain areas. In SafeMove the user enters a set of coordinates, which together creates a zone (or several). These zones are only showed as numbers and dots on a two dimensional coordinate system and can be rather difficult to grasp (see Figure 5.2). To have a way of visualizing them could therefore be really useful.

The next step was trying to create a virtual model of the zone. In Unity3D it is easy to create objects with simple forms such as cubes and spheres but more complex objects are more difficult to produce. Since the zone can have between three and eight coordinates it was not possible to take a cube and simply adjust it's size. Instead a function which took the numbers of coordinates given and calculated how the zone should look was written. This function took the coordinates and draws a line between each coordinate and it's neighbors, thus creating a wireframe. Figure 5.3 shows examples of different possible safety zones.

5.2.2 Tracking

In order for the application to know where to put the virtual model the robot needed to be tracked.

²See *Development Challenges and Solutions: Switched from iOS to Windows* on page 21

³See *Development Challenges and Solutions: Errors while building the application* on page 21

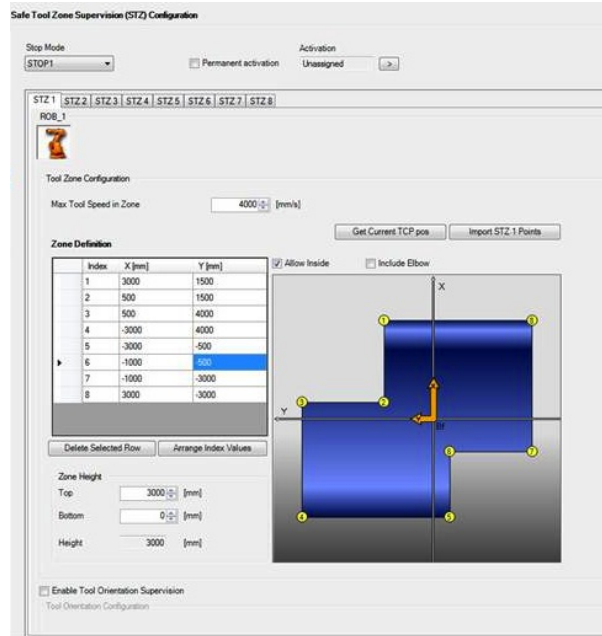


Figure 5.2: SafeMove coordinate input interface.

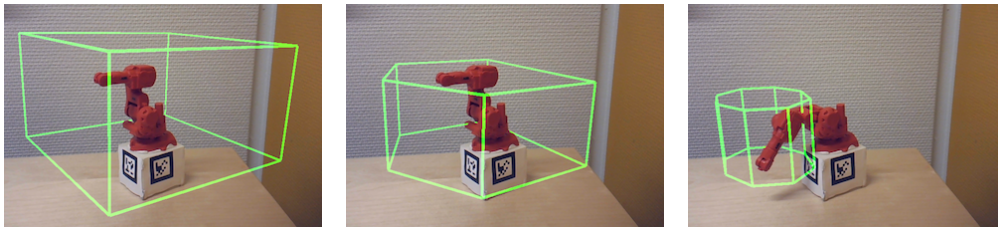


Figure 5.3: Different forms of safety zones.

Metaio can use both optical and non-optical ways of tracking. The non-optical way uses GPS and the phone's inertial sensors. Since the robot is situated indoors it was not possible to use this type of tracking hence an optical way was used instead.

There are different types of markers that can be used when working with optical tracking. These markers can be either ordinary pictures, barcodes, QR codes or a number of other different types. There are also Metaio's own ID markers. These types of markers are called fiducial markers⁴ and look like a bigger version of QR codes and eventually these were the markers chosen.

The markers work as reference points and make it possible for the Metaio software to know where to place the model. Because of this one of the markers always has to be visible in order for the model to be aligned correctly.

At first a picture marker, which was included in a Metaio tutorial, was used. It was difficult to get the tablet to track this marker in a satisfactory way so after a lot of time working with different pictures the ID markers were chosen instead.

⁴http://en.wikipedia.org/wiki/Fiducial_marker, accessed 2014-01-20

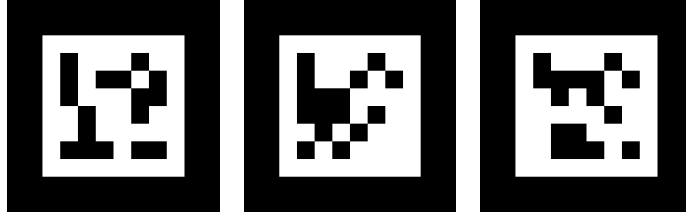


Figure 5.4: Three of Metaio’s ID markers (fiducial markers).

It was rather difficult to get good tracking with this marker as well but after some time tweaking different parameters and reading questions and answers in the Metaio forum the tracking improved. Metaio also released a new version of their SDK, which provided a more stable tracking.

The next questions were where to put the marker and how many markers that should be used. To make it possible to move around the robot several markers would be needed. It was also important to place the markers so that they would not be obscured by the different parts of the robot when the robot moves.

One possible solution would be to place the marker on the floor next to the robot. This would work but be problematic in several ways. The floors next to the robots are often dirty so the marker would probably get soiled rather quickly. On bigger robots it would be impractical because of the size. The user would want to look up to see the zone around the robot but would have to point the camera downwards in order to see the marker. The angle between the marker and the camera would also be large and the larger the angle the more difficult it is for the software to track the marker. An angle larger than 45° can cause a flickering result or no result at all.⁵

The size of the marker was also an important aspect to consider. If the marker was too small then it would be difficult for the camera to focus on it and the software might not recognize it. The safety zone of the robot can be several meters wide so the marker had to be quite large too.

The robot provided for testing was an IRB140 – a one meter high robot mounted on a small table. Because it was situated on a table it was possible (and practical) to place the markers on each side of the table. This way the robot would not obscure the markers (for the most time) and the horizontal angle between the markers and the camera would be fairly small. The size of the table was 50 x 42 x 41 cm so printing on A3 papers⁶ was ideal. Since most office printers can print A3 there would also be no need to order custom made prints from a printshop.

Taking margins into account the result was a marker with the size of 27.2 x 27.2 cm. Tests showed that the user could stand at a distance of 3 meters and still have good tracking.

To be able to look at the robot from more than one side additional markers would have to be used. As a proof of concept another marker was added on one side but it would be easy to add markers to the two remaining sides as well.

5.2.3 Zone Adjustment

A function that was added was the option to manually change the size of the safety zone.

⁵See <http://dev.metaio.com/sdk/tracking-config/create-image-trackable/> (accessed 2013-10-23) for more information.

⁶A3 paper size: 29.7 x 42 cm

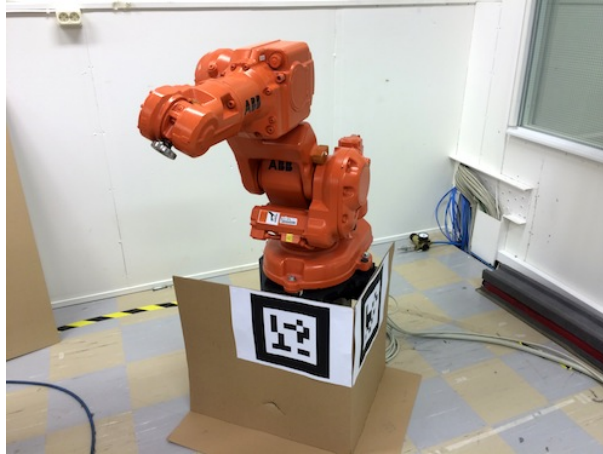


Figure 5.5: The ABB IRB140 robot.

First, gameobjects⁷ in form of spheres were added to the positions of the coordinates and the wireframes were then connected to the spheres. For the sake of convenience only the bottom spheres were made visible and each one had a different color to distinguish them from each other.

Second, a script was added to the spheres which made it possible to drag them in the x and z dimensions.⁸ Since the lines of the wireframe were connected to the spheres they moved as well when the spheres were moved.

Only the option to change size horizontally was implemented, not vertically.

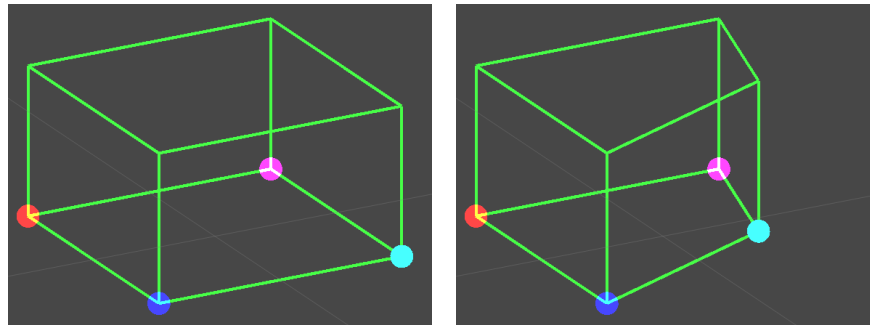


Figure 5.6: Wireframe with colored spheres for size adjustment.

5.2.4 Occlusion Model

In AR the virtual model is added on top of the real world, between the world and the camera.⁹ This can make the experience less realistic since the model often should be situated behind a real world object.

⁷See *Software: Unity3D and MonoDeveloper* on page 10

⁸In Unity3D the y dimension is the vertical dimension.

⁹At least when using ordinary cameras – with a depth sensitive camera, e.g. Microsoft Kinect, or special filming techniques this can be avoided.

To tackle this problem an occlusion model was used. An occlusion model is an invisible model that blocks – occludes – all the other models behind it. In Unity3D this is achieved by using a special material provided by Metaio.

Occluding a static object like a table or computer screen is not so difficult. One only have to create a virtual model and use it as an occlusion model. With moving objects, such as a robot, it is much trickier.

This was done by establishing a connection to the robot and extracting the current location of all the different links. The rotation value of each axis was used to calculate the robot's position and the occlusion model was moved to match the real robot.

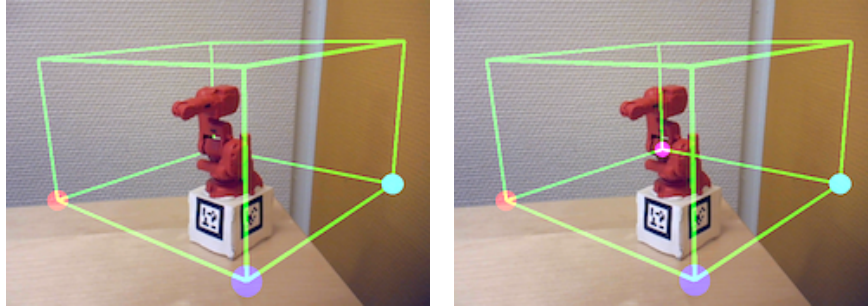


Figure 5.7: The occlusion model turned on and off.

5.2.5 Calibrating the Markers

By default the model is placed right on top of the marker. It is possible to change this by changing two sets of parameters when loading the Metaio plugin. One set of parameters controls the translation offset and the other one the rotation offset.

The calibration is very important for integrating the virtual model into the real-world scene. A small calibration offset at the marker leads to a large offset further away.

Two ways of calibration were tested where the first one was calculating the offsets. The rotation offsets were easy: on one side the model had to be rotated 90° along the x axis and on the other side it had to be rotated 90° on both the y and z axes.

Calculating the translation was a bit trickier. By measuring the size of the marker, dividing it in half and adding the whitespace around it a translation offset were calculated. However, this was not as accurate as hoped and was not usable.

The second calibration method was manual calibration. A model of the robot's base was placed in the scene and by pushing buttons the user could align it and place it over the real robot. This had to be done individually for each marker and was not very easy. It was difficult to see if the model was placed exactly in the right place.

In both these calibration methods it was really important that the marker was placed completely horizontal otherwise the rotation offset would be wrong.

A third way of calibrating was prepared for but never fully implemented (due to lack of time). By using the robot itself to do the calibration it would be possible to get very exact results.

5.2.6 Connecting to the Robot

The main reason ABB Robotics wanted the application to be developed on the Windows platform was to make it able to connect to the robot.

A connection between the application and the robot was needed for a number of reasons: To get the occlusion model to follow the robot, to draw the TCP trail and to get the coordinates of the safety zone. The last function, the coordinate extraction, was not fully implemented due to lack of time. These coordinates are stored in a XML file in the robot controller and instead of downloading this file from the robot controller the file had already been saved on the tablet. The function of extracting the coordinates from the XML file was however implemented.

To show the position of the TCP, a sphere with a trail renderer was created. A script, which got the position of the real TCP was written and attached to the sphere. The trail renderer leaves a trail behind it showing the path the TCP has taken (see Figure 5.8).

In order for the model to be able to follow the robot it needed a constant connection to the robot controller. Every frame the application extracted the rotation value of each of the robot's links. The model was then updated and the corresponding link was rotated to the correct position.

There were some problems with the connection to the robot but eventually they were solved. See *Development Challenges and Solutions: Connection problems* on page 21 for more information.

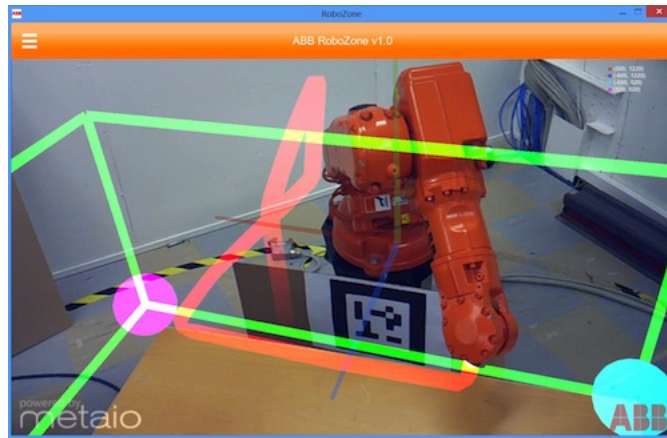


Figure 5.8: TCP trail.

5.2.7 Interface

Since the main focus of this thesis work was to create a functional application the interface design came second. The interface was designed to have simple layout and focus on functionality.

The application would be used on a tablet using touch-based inputs and therefore the interface had to be designed for this. Here the difference between Unity3D's play mode and the real application became obvious since the resolutions were not the same. This made it more difficult to design the interface and initially the buttons and the user controls were often too small.

Since it can be tiresome to hold the tablet with just one hand all the buttons were placed along the left side of the application, thus making it easier to interact (see Figure 5.9).

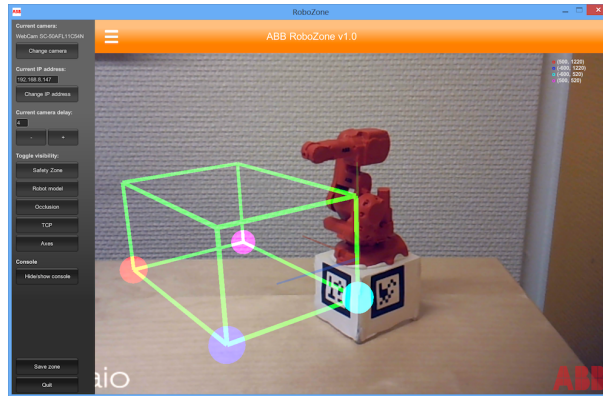


Figure 5.9: The user interface (in opened state).

5.2.8 AR Testing

Before using the application on the real IRB140 robot provided for testing (see Figure 5.5) a number of different testing equipment were used (see Figure 5.10).

In the beginning a picture marker placed on the wall was used as tracking reference. This worked fine when no advanced interaction was necessary, e.g. the development of the zone function.

Later a mini version of the test robot set was constructed. ABB Robotics provided a robot model of scale 1:10 and a small platform was crafted with 3D printer. Since the real robots were located in other buildings this model was very convenient and allowed for easy testing.

To test user movement and for getting a feeling of the size of the zone two markers were placed on a shelf. With this set it was possible to walk around and to see how far away one could stand with the tablet and still get an acceptable experience.

The occlusion tests were made with an unplugged robot. Since this robot could not move it could only be used for static testing but it gave a good feeling of how the occlusion function worked.

Finally testing with the fully functioning robot took place at ABB Robotics. Here the connection and the movement of the model were tested.



Figure 5.10: AR test flow.

5.2.9 Development Challenges and Solutions

Switched from iOS to Windows

The concept was made in Unity3D with the Vuforia AR framework and was developed for an iPad. Since the ABB Robotics use the Windows platform for most of their products they wanted the prototype to be made for Windows as well. The problem was that the Vuforia framework does not have support for Windows. The quest for a new framework began and eventually the Metaio framework was chosen. Metaio seemed to be a good choice – it was used in many commercial applications, it was free (although with a watermark in one of the corners of the screen) and it had support for Windows as well as Unity3D. This was actually the only framework that had support for both Windows and Unity3D.

Errors while building the application

One of the first major problems discovered was that after the application was built it did not work (most of the time). While using Unity3D's play mode¹⁰ the application worked without problems but after it was built it did not recognize the markers. This happened most of the time but not always. There was, however, no consistency when it did or did not work.

Eventually it was discovered that not all of the necessary dll files were copied when the project was built. If they were manually copied the application would start to work.

In mid October Metaio released a new version of their framework and after upgrading to this version this problem disappeared.

No examples, bad documentation, small community

One of the best things with the Vuforia framework was the documentation and the tutorials. It was really easy to get started and everything worked right from the start.

It was much tougher to start working with Metaio. There was no good tutorial on how to develop for Windows in Unity3D, only for iOS and Android development.

Eventually with the help from a couple of "webinars" – a series of recorded web seminars – a first test application was produced.

Another problem was that the Metaio community was much smaller than that of Vuforia. The number of developers for Windows were also much smaller than for the rest of the platforms. Often when posting questions in the official forum no answers were given.

Connection problems

It took some time to get the connection between the application and the robot working. See Figure 5.11 for an overview of different approaches.

Unity3D uses Mono¹¹ which is a cross-platform, open source .NET implementation. Mono only supports managed code and .NET 2.0, which proved to complicate things.

To access the robot controller the easiest way is use ABB's PC SDK¹² which provides easy access to the controller. After some time and a lot of testing it turned out that the PC SDK needs .NET 4.5 to work and thus could not be used.

¹⁰See *Software: Unity3D and MonoDeveloper* on page 10

¹¹<http://www.mono-project.com>, accessed 2014-01-15

¹²<http://new.abb.com/products/robotics/robotstudio>, accessed 2014-01-15

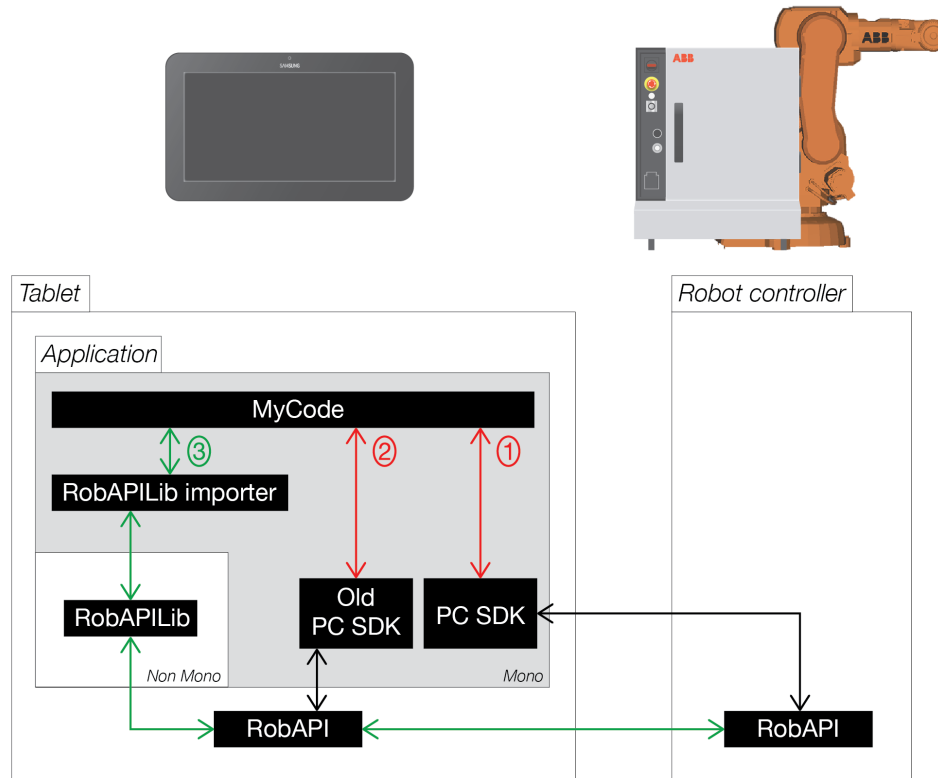


Figure 5.11: The application-to-robot connection.

When this did not work an older version of the PC SDK was tested. This version only needed .NET 2.0 but it used a mix of managed and unmanaged code¹³ so this did not work either.

If the PC SDKs would have worked they would have had a connection to the RobAPI – an interface application which works as a link between the robot controller and other computers. The new PC SDK can connect directly to the RobAPI in the robot controller whereas the old PC SDK connects to a RobAPI in the same computer, which in turn has a connection to the RobAPI in the robot controller.

On the third try a solution was found and instead of using the PC SDKs ABB Robotics created a plugin for Unity3D, the RobAPILib, which is used by the application to access the RobAPI framework in the computer. Lastly a class was created in Mono which imports the functions in the RobAPILib making it possible to access data in the robot controller.

Rotation problems

The robot consists of a base module and six links which each can rotate around a separate axis. The links have a relationship where a movement in the lowest link affects the position of all the upper links but a movement in the top link does not affect the position of any of the lower links and so forth.

¹³http://en.wikipedia.org/wiki/Managed_code, accessed 2014-01-15

In Unity3D the robot model was constructed in the same way. The base module was the parent object and each link was the parent of the next one in line. However, when getting the rotation angles of the robot and trying to rotate the corresponding link of the model all the upper links took no regard of this bottom-up relationship and started rotating around the same axis. This error was caused by using a wrong rotation function. The solution was using a different function, which rotated the object relative to the parent.

Problem with delay

There were some delay between the input and output of the real world video stream where the output was delayed around 30 to 50 milliseconds. This was not a big problem in itself since this delay did not affect the experience much. The output of the movement of the models did, however, not suffer from this delay, which resulted in the models always moving ahead of the real robot. This way there was always some offset between the occlusion model and the robot.

To solve this problem a queue was constructed which took the current position of the robot and stored it. When the queue had been filled with a few values the first value was then dequeued and used to update the position of the model.

Pivot point

The robot model was imported from RobotStudio and consisted of seven parts; the base module and the six moving parts. There were no problems with aligning them because Unity3D did this automatically. The pivot point¹⁴ of the joints were, however, not in the right position.

In Unity3D it is not possible to change the pivot point of objects directly. To solve this problem empty gameobjects were added to serve as placeholders. The empty gameobject were placed at the coordinates of where the pivot point should have been and the robot part were made a child objects of this (see Figure 5.12).

Instead of rotating the robot part itself one could then rotate the parent object which had the right pivot point.

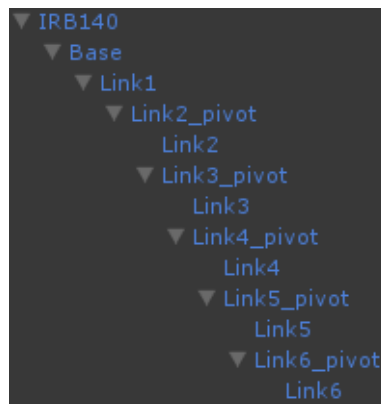


Figure 5.12: Child-parent relationship of the robot model's parts.

¹⁴The point of which the link rotates around

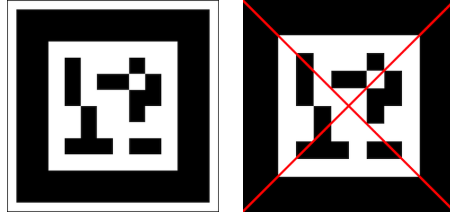


Figure 5.13: The markers should have a white border to improve tracking.

Tracking problem

There was a lot of trouble to get the marker tracking to work in a satisfactory manner when using the Metaio framework. Even when placing the camera close to the marker the virtual model would often shake and flicker. The tracking stability increased after Metaio released a new version of its framework but there were still big problems.

After asking questions on the Metaio forum and showing a short video of the flickering model a few suggestions were given. It is important that there is white border around the marker since this helps with the tracking (see Figure 5.13). The resolution, both of the camera and the application, could also be experimented with. If the camera resolution were too low it would not make out the different parts of the marker when standing a bit further away. But if the resolution were too high the application would become very slow.

Chapter 6

Results

In this chapter the final results of the thesis work are presented.

6.1 The Prototype

The result of the prototyping phase was a Windows tablet application that visualizes an ABB industrial robot's SafeMove coordinates using AR. The application is designed to be a tool for maintenance engineers, helping them verify that the SafeMove settings are correctly adjusted.

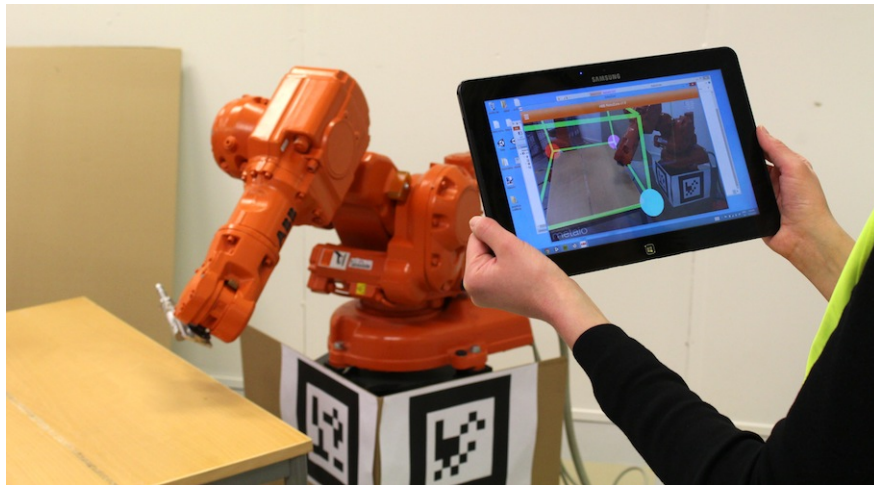


Figure 6.1: User holding the tablet.

6.1.1 User guidance

To use the application the maintenance engineer holds the tablet and points it towards a robot (see Figure 6.1). The recorded camera stream is displayed on the screen and the picture is augmented with a model of the safety zone showing in which area the robot is allowed to move.

The safety zone is displayed as a wireframe with colored spheres in the bottom corners. By dragging these spheres the user can adjust the size of the zone. This new layout can then be saved.

The connection to the robot is established automatically with a predefined IP address. It is possible to change this address in the settings menu.

Fiducial markers¹ are used to track the position of the robot and correctly align the model of the safety zone to the real world. Before using the application one or more of these markers must be attached to one of the sides of the robot. At least one marker must always be visible, and close enough for the tablet to be able to focus on it, for the tracking to work. When the markers have been attached they must be calibrated.

Calibration

Each marker must be calibrated² in order for the application to work properly. The markers are used as reference points and their relative location to the robot must be obtained to properly align the safety zone.

The calibration is done by aligning a transparent model of the robot's base part on top of the real base part (see Figure 6.2). Using buttons the model can be moved in all three dimensions and aligned correctly. This procedure must be done once for each marker.

Each time the application is started the user gets the choice to do a new calibration or use the calibration from last time.

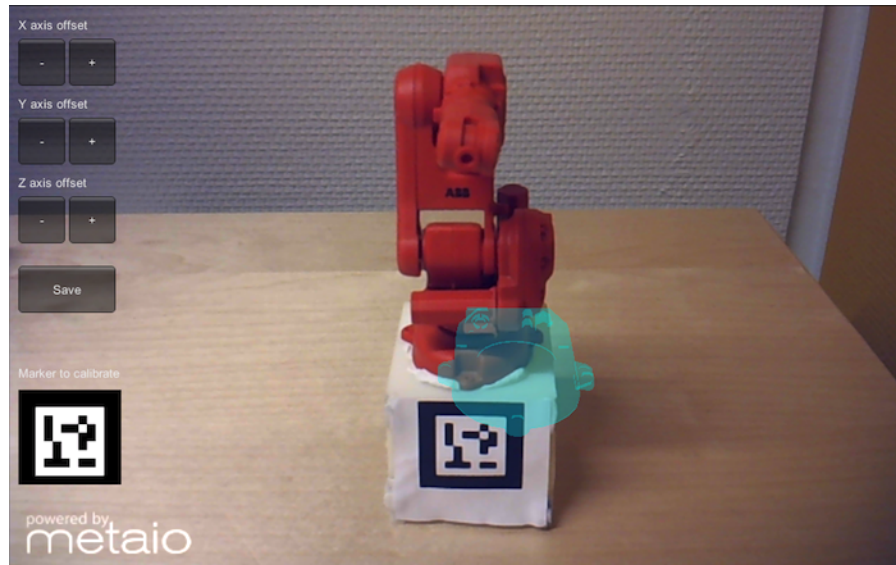


Figure 6.2: Calibration of one of the markers.

6.1.2 Interface

This section describes the application's interface.

¹See *Tracking* on page 14

²See *Calibrating the Markers* on page 18

Coordinates

In the upper right corner of the screen the coordinates of the zone are displayed. These are updated as soon as one of the spheres is moved. Next to the coordinates the matching sphere color is displayed.

Settings

The application contains a settings menu where the user can do different types of adjustments. This menu is opened and closed by pressing the \equiv icon. The functions of the different buttons in the settings menu are described below.

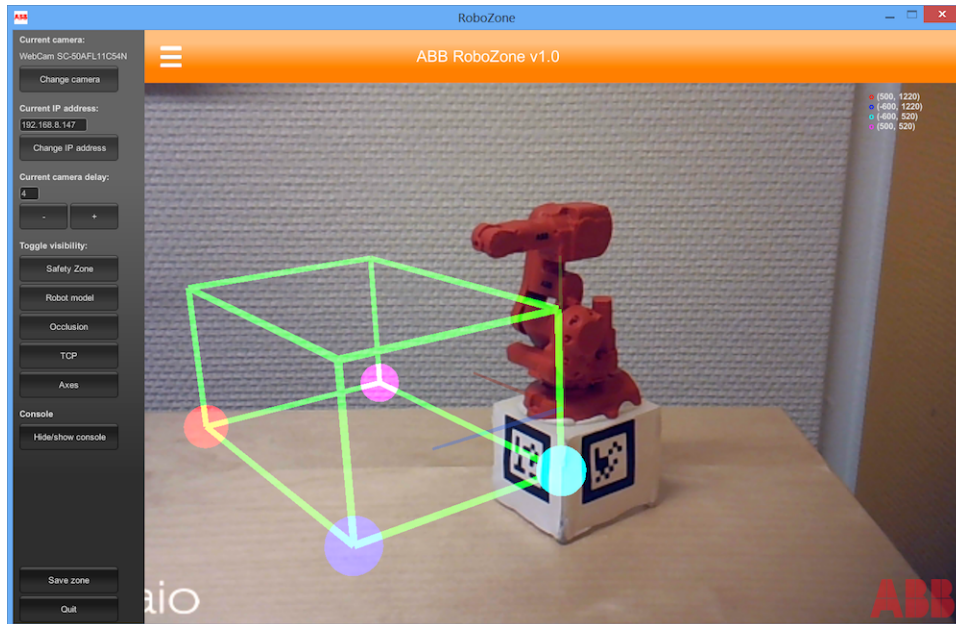


Figure 6.3: The application with settings menu open.

Change camera

Switches between the different cameras attached to the computer.

Change IP address

Allows the user to switch IP address and connect to another robot.

Current camera delay

Adjust the delay of the camera.

Toggle visibility of safety zone

Toggles the visibility of the safety zone on and off.

Toggle visibility of robot model

Toggles the visibility of the robot model on and off.

Toggle visibility of occlusion

Toggles the visibility of the occlusion model on and off.

Toggle visibility of TCP

Toggles the visibility of the TCP marker and the trail on and off.

Toggle visibility of axes

Toggles the visibility of the axes showing the coordinate system

Hide/show console

Open and closes the console.

Save zone

Saves the current zone layout.

Quit

Closes the application.

6.2 The Evaluation

Two evaluation sessions took place. Both sessions were formed as user studies where the test subject would first use the tablet and then be asked questions about the experience. The questions were open and the test subjects were encouraged to speak freely and express all their thoughts and feelings. Three test subjects were involved, all ABB employees working with robot development. One test subject had no knowledge of SafeMove and the other two had a lot of experience.

In the first session the application was tested on the employee with no experience of SafeMove. The test subject had mixed feelings, while he could see some use of the application he also thought the setup was rather unwieldy. It was useful to be able to see the zone but the test subject missed the functionality of being able to take a snapshot of the scene. Since it is tiresome to hold the tablet with stretched arms for a long time it would be easier to interact with the snapshot while holding the tablet in a more comfortable position.

The second evaluation session involved the two employees with SafeMove knowledge. They tested the application one at a time and afterwards discussed the experience. Both of these test subjects were very satisfied with the application and believed that it could be of good use. It would be especially useful for those companies who do not make virtual models of their cells³. Both test subjects thought that the occlusion model really improved the experience and one of them went as far as saying that he did not think it would work without it. They did not think that applying the reference markers and calibrating them would be a problem as long as it could be done easily with the help of the robot⁴. The only big problem the test subjects from the second evaluation saw was how one could get a good feeling of the zone when the zone was so big that one were standing in it.

³Short for workcell – a collection of robots and other machines working together at a factory

⁴See *Discussion* on page 29

Chapter 7

Conclusion

In this chapter the development phase and the result are discussed. In addition, two other sections describe the application's limitations and what future work could involve.

7.1 Discussion

As this being a very open project it was difficult knowing what the end result would be in the beginning. ABB wanted an application for a tablet that should implement AR and be used by maintenance engineers working with industrial robots. The final result was a tool for displaying the safety zone around a robot but there are of course a lot of other applications for AR in the industry.

The application is meant to be used by maintenance engineers to check if the safety zone has been configured correctly. This result did not really coincide with my early visions where the application would rather be used to inspect robots with unknown movement paths, giving the user an idea of how close one could stand to the robot (without being afraid one could get hit by it). When we decided to use the SafeMove coordinates the project automatically changed direction and a tool for verifying the zone configuration seemed like a more logical choice.

For me it was important the application provided some kind of added usability, not only showing off a (relatively) new kind of technology, AR. Therefore the interrogative "why" was always present: We could do it like this – but *why* should we? AR could be used to display that – but *why* should it? AR is a powerful technology and I am certain that we will see extensive use of it in the future but right now it is many times being used in ways that are inconvenient and not very user friendly. As an interaction designer I really believe that usability is important.

One big problem when using AR technology on a tablet or a phone is that the user has to hold the device with one or two hands, preventing the user from fully interacting with the task. It is also tiresome to hold a tablet in front of you with either stretched or bent arms for a longer period of time. Interacting with this application is no exception. The controls are placed closed to the side making it easier to access them without letting go of the tablet and instead moving the thumb. However, when moving the spheres is inevitable to let go of the tablet with one arm making it more difficult to adjust correctly.

To solve this problem it is possible to wear the screen instead of holding it. There are many examples of head mounted AR technology, one of the latest being the Google Glass. When using these types of devices one's hands are not obstructed from use, which greatly

improves the way one can interact with the environment. However, how this interaction is supposed to work – when there is no screen to touch – is not obvious. Google Glass uses voice commands but since industrial environments often are noisy it is not certain that this would work well.

7.2 Limitations

Since the application is a prototype, a proof-of-concept, it does not have full functionality. The only included robot model is the IRB140 so it will not work properly with other types of robots. The TCP marker will be positioned correctly but the occlusion model will be mismatching.

Due to lack of time the function for automatically obtain the SafeMove configuration file from the robot controller was not implemented. The file is instead stored on the tablet.

There are a few bugs in the application and there is very little error handling. It is possible to open a console to read error messages but much more information could be given.

The application was only tested on ABB personnel working with the development of ABB robots and software. They provided useful feedback and the result of their tests were important. However, it would also have been useful to test the application on non ABB employees, on people who work with ABB's products in various industries. There were a few reasons why this was never done. At the time of the test sessions it was not yet decided how secret the projects was, and if external people would be allowed to take part of the project. It would also be difficult to do test at external sites since the application was custom made for one special type of robot and also needed a network connection with the robot in order to work.

7.3 Future Work

Using Unity3D when implementing the prototype worked pretty well. However, if an application with full functionality is going to be implemented it would probably be better to create it with another, more flexible, program. Since ABB Robotics use the .NET framework in all their applications Visual Studio (VS) would be a good choice for this applications as well. It is possible to use the Metaio framework together with VS but only C++ is supported.

There are a few bugs in the application so there are some improvements to be done. Two things to change are the connection procedure and the structure of the occlusion model. In the current version the application checks the robot's position every frame which is not very efficient. There is also probably a better way algorithm to check each link's position. Right now each link's position is checked and updated even if that link has not moved.

The occlusion model is very detailed and therefore quite performance intensive. The model does not need to be this detailed and if a model with simpler form and shape is used the performance will be better.

At the moment it is rather difficult to get a precise calibration when calibrating the markers. Instead of doing the calibration by hand it is possible to use the robot. The robot has a built in calibration functionality which can be used to calibrate the markers as well. This would take a bit longer time than doing the calibration by hand but would provide a much more exact calibration.

Finally, tests at external sites would provide useful information of how the application could be used by non-ABB personnel.

Chapter 8

Acknowledgements

There are a number of people who I would like to thank:

Elina Vartiainen, my supervisor at ABB, for excellent guidance, interesting conversations, good advice for the future and for turning me into a movie star.

Kalle Prorok, my supervisor at Umeå University, for help with the report and all the encouraging words.

Per Willför for all the programming help and for showing me how cool industrial robots can be.

Karin Nilsson Helander and **Jonas Brönmark** for taking me under their wings and making my time at ABB both fun and interesting.

Susanne Timsjö and the rest of the **SARU group** for really making me feel part of the team.

And **Elin Sjöström** for all the love and support.

References

- [1] ABB. *Safemove Application Manual, Revision C*. <http://developercenter.robotstudio.com/Index.aspx?DevCenter=ManualsOrRobotStudio&OpenDocument&Url=../SafeMoveAppManual/doc4.html>, accessed 2013-12-04.
- [2] Abrar Omar Alkhamisi and Muhammad Mostafa Monowar. Rise of augmented reality: Current and future application areas. *International Journal of Internet and Distributed Systems*, 1:25, 2013.
- [3] Ronald T Azuma et al. A survey of augmented reality. *Presence*, 6(4):355–385, 1997.
- [4] Jan Berssenbrügge, Sascha Kahl, Helene Wassmann, and Jürgen Gausemeier. Design and VR/AR-based testing of advanced mechatronic systems. In *Virtual Reality & Augmented Reality in Industry*, pages 1–37. Springer, 2011.
- [5] Julie Carmigniani, Borko Furht, Marco Anisetti, Paolo Ceravolo, Ernesto Damiani, and Misa Ivkovic. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51(1):341–377, 2011.
- [6] Thomas P Caudell and David 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*, volume 2, pages 659–669. IEEE, 1992.
- [7] Hung-Lin Chi, Shih-Chung Kang, and Xiangyu Wang. Research trends and opportunities of augmented reality applications in architecture, engineering, and construction. *Automation in Construction*, 2013.
- [8] HC Fang, SK Ong, and AYC Nee. Interactive robot trajectory planning and simulation using augmented reality. *Robotics and Computer-Integrated Manufacturing*, 28(2):227–237, 2012.
- [9] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 85–94. IEEE, 1999.
- [10] Stefan Nolle and Gudrun Klinker. Augmented reality as a comparison tool in automotive industry. In *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*, pages 249–250. IEEE, 2006.
- [11] Manuel Olbrich, Harald Wuest, Patrick Riess, and Ulrich Bockholt. Augmented reality pipe layout planning in the shipbuilding industry. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 269–270. IEEE, 2011.

-
- [12] Holger Regenbrecht, Gregory Baratoff, and Wilhelm Wilke. Augmented reality projects in the automotive and aerospace industries. *Computer Graphics and Applications, IEEE*, 25(6):48–56, 2005.
 - [13] Brandon Stutzman, David Nilsen, Tanner Broderick, and Jeremiah Neubert. Marti: Mobile augmented reality tool for industry. In *Computer Science and Information Engineering, 2009 WRI World Congress on*, volume 5, pages 425–429. IEEE, 2009.
 - [14] DWF Van Krevelen and R Poelman. A survey of augmented reality technologies, applications and limitations. *International Journal of Virtual Reality*, 9(2):1, 2010.
 - [15] Sabine Webel, Uli Bockholt, Timo Engelke, Nirit Gavish, Manuel Olbrich, and Carsten Preusche. Augmented reality training platform for assembly and maintenance skills. *Robotics and Autonomous Systems*, 2012.
 - [16] Charles Woodward, M Hakkarainen, O Korkalo, T Kantonen, M Aittala, K Rainio, and K Kähkönen. Mixed reality for mobile construction site visualization and communication. In *Proc. 10th International Conference on Construction Applications of Virtual Reality (CONVR2010), Sendai, Japan*, pages 35–44, 2010.