Handheld AR for advertising

Semester: Medialogy semester 10

Title: "*Handheld Augmented Reality for advertising*"

Project Period: spring 2011

Semester Theme: Master thesis

Supervisor(s): Stefania Serafin

CPR no.: 241178xxxx

Members: Christian Gjelstrup

Copies: 3

Pages: 110

Finished: May 27th, 2011

Aalborg University Copenhagen
Lautrupvang 15, 2750 Ballerup,
Denmark

Semestercoordinator: Stefania Serafin
Secretary: Lisbeth Nykjær
Phone: +45 9940 2471
lny@create.aau.dk

Abstract:

Inspired from the massive hype there lately has been around Augmented Reality (AR), this thesis is an investigation into the area and the technology behind it. The catalyst for the hype has been identified as today's smartphones, which, because of their increasing hardware capabilities, provide all necessary means for enabling AR experiences. But with one significant difference: Due to the portable nature of smartphone devices, these experiences now can be transferred to mobile outdoor settings. Combined with recent development within AR tracking, such as tracking natural features (NFT) in images rather than tracking binary fiducials, this holds potential of pushing AR into a new era.

Having identified a need for the advertisement industry to adapt to new technologies in general, such target area was chosen, and a field study with a professional advertisement agency was initialized. This resulted in an iPhone App, utilizing NFT, which was directed towards augmenting an outdoor environment. With the perceptual effect of the experience in focus, theory on sensual and spatial presences was used for evaluation.

Preface

This report is the result of the work done during the project period of 10th semester at Medialogy, Aalborg University Copenhagen 2011. The following text will provide a guideline in how to read the report.

Chapters and sections

The chapters of the report are denoted with ongoing numbers, for each chapter and subchapter. The ongoing numbers for subchapters, referred to as sections, are starting from 1 in the beginning of every new chapter. The depth of the subchapter's denotation will not exceed 3 levels. Smaller subsections are left without numbering.

References, quotes and figures

References to authors will be written in square brackets, containing the surname of the author, the year of publication and the page number. If no specific page is referred to, page number will be omitted and if there are multiple authors, the first author will be noted by name, followed by "et al". Reference to websites will be written in square brackets, containing the phrase "web" followed by an ongoing numbering.

Quotes in the text are marked by citation marks and the entire quote will be written in *italic*. If there is preceding or subsequent text, which is not desired within the quote, these will be denoted using three dots within square brackets. The source for the quote will be defined immediately after the quote, in square brackets, i.e. *"…and this is how a quote is written."*

Pictures and figures are explained by a caption, which contains ongoing numbering, i.e. Picture 9 or Figure 9. The lists of pictures, figures, and web references will be listed in the end of the report immediately after the Bibliography.

The appended CD contains the report as PDF and Word document, together with an audio recording of a meeting with ADs from Neue Digitale / Razorfish.

Acknowledgements

I wish to thank the staff at Neue Digitale / Razorfish for collaboration during the project period.

Table of content

1. Preliminary analysis

1.1. Introduction

Augmented reality (AR) seems to have become a hot topic, which has attracted a lot of attention during the last couple of years. As seen on Google Trends [web 1], or in Appendix A, web searches on the term *Augmented Reality* has approximately tripled during 2009 without significant degrease throughout 2010. There are suggestions about the attraction being caused by the fact that AR does no longer require the use of stationary PCs but now can utilized on Smartphone devices. As an example ABI Research in late 2009 predicted handheld platform to transform the AR ecosystem [web 2]. Before going deeper into the area of handheld AR for the smarphones, it is appropriate to understand what AR means.

1.1.1. What is AR?

In short, AR is about augmenting the real environment with virtual information, which often, but not necessarily, is visual content in the form of text or 3D graphics [Carmigniani et al. 2010]. In this way AR can be viewed as a technology, which transfers computer-generated information into its users view of the physical world. A more specific and thorough explanation, of what should be understood by AR, will be carried out on a later point.

1.1.2. Why is AR interesting?

Why is combining real objects and virtual objects (VOs) in 3-D useful or interesting? The idea behind Augmented Reality is to enhance a user's perception of and interaction with the real world [Lalanne & Jürg 2009]. The VOs can display information, which the user could otherwise not detect with his senses and such information could for instance help in performing real-world tasks. Hereby AR is a specific example of what Fred Brooks calls *Intelligence Amplification* (IA): using the computer as a tool to make a task easier for a human to perform [Azuma 1997].

Before going deeper into how AR can be constructively applied, it is relevant to examine what smartphones, as a platform, can offer in context of AR.

1.1.3. What makes the Smartphone platform attractive for AR?

According to [Haller et al. 2007 *p.106*], mobile phones have in recent years developed into being an ideal platform for augmented reality (AR) due to the increasing capabilities of the integrated hardware and combination of build-in functionalities. Because of the advance in technology, the current generation of phones, smartphones, are having increasingly faster processers and the tradeoffs between device size and capability are equalling out with smartphones getting smaller and more powerful. Today's smartphones have integrated cameras to capture their surroundings, dedicated 3D graphics chips for rendering virtual graphics in real-time and full colour displays to present the virtual material in the context of the physical world. They have multi-touch screens as input device for interaction and are equipped inertial sensors, such as digital compasses, accelerometers and gyroscopes, for measuring device orientation as well as Wireless network (broadband) access, which can provide information such as GPS location. In combination, such features apparently constitute all necessary means for enabling AR experiences.

Besides for the build-in functionalities of the smartphones, other attributes making them attractive for AR could be related to fact that an increasing amount of people uses them. In 2010 more than 300 million smartphones where sold and The Coda Research Consultancy predicts a global sale of 2.5 billion during the 2010-15 period [web 3]. This means that an extensive amount of users is already out there, which number will grow into being a majority, of mobile users in the Western society, during the forthcoming years. As smartphones are becoming an integrated part of our everyday lives, there is a strong market for AR Smartphone applications. Furthermore the users are already having familiarity in using the platform, which is unusual for new hardware.

1.1.4. Why is handheld AR so special?

A very interesting aspect of bringing AR to mobile devices is their portable nature. Unique for smartphones, as compared with traditional stationary PCs, is the fact that these computing devices are available at all times and not attached to a specific location. They are with us wherever we go and this ubiquity opens up for a new series of AR applications, within the public domain, indoor as well as outdoor. In this context, AR can be experienced everywhere in theory. As Vernor Vinge has said to the phenomenon: *"Cyberspace has leaked into the real world*" [web 4].

1.1.5. How does the future of handheld AR looks?

Visiongain, an independent business information provider, has conducted a market research report on mobile AR 2011-2016 [web 5]. Based on research and analysis of the global markets, they conclude that mobile AR, as a market sector, is on a verge of a commercial boom and predicts mobile AR to be one of the most significant growth markets in the communications and technology sectors over the five years to 2016. This is supported by the so-called *Hype Cycle* for Emerging Technologies, made by Gartner Research, which identifies mobile AR as one of the ten most disruptive technologies for 2008-2012 [web 6]. Other predictions point to AR applications to become pre-loaded on smartphones in future as well as an estimated 150 million to 200 million users in 2012 [web 7]. In the same way Juniper Research has recently issued a report, which predicts global revenues from mobile AR applications and services to approach $1.5 billion by 2015 [web 8].

Whether or not these predictions are exact is not as important as the fact that AR currently is in the spotlight and assumable on the vast of a boom with discussable dimension. The tendencies shows that AR applications for smartphones is in the process of changing, from merely being prototypes, related to research etc., towards becoming commercial consumer products for a mass market.

1.1.6. Motivation

All in all these various factors make AR, for the handheld platform, a very relevant and actual research topic anno 2011. Well suiting the general approach of investigating how new technologies can be utilized in new ways and for new purposes, handheld AR has been chosen as overall scope for this Medialogy Master thesis. From a Medialogy perspective, there is a range of interesting topics and problems connected to such technology and the usage of it. Having chosen the technical direction for my Medialogy education, I personally have an interest in how cutting edge handheld AR systems can be utilized, and considering my profile, which is *Interaction*; I find it relevant to work with an interactive field such as AR experience. The focus for this thesis will therefore be on new opportunities provided by the ubiquity of smartphones, i.e. the qualities that can be enabled by bringing AR to the public domain. This leads to the following initial problem formulation:

1.2. Initial problem formulation

"What is handheld AR and what can it be used for?"

1.3. Defining AR

As mentioned earlier, AR is about augmenting the reality with virtual information. To define AR, it is therefore appropriate to define what should be understood by reality. According to Poslad, reality refers to the state of actual existence of things in the physical world, which means that these things exist in time and space, as experienced by a conscious sense of presence of human beings, and are situated and embodied in the physical world [Poslad 2009 p.38]. So when AR is augmenting the reality, it is in fact augmenting the conscious experience of the reality and thereby it is the human perception of reality that can be altered or enhanced by AR. To measure and evaluate AR, the human perception thereby has to be considered and taken into account.

The Oxford dictionary defines AR as a technology that superimposes a computer-generated image on a user's view of the real world, thus providing a composite view [web 9]. Besides for only considering virtual content in visual form, such definition can seem imprecise and weak as for instance a TV broadcast then could be thought of as AR. [Bimber & Raskar 2005] point to the fact that the augmented information has to have a strong link to the real environment where the AR is situated, which explains why TV not would be AR. Besides from merely integrate synthetic information into the real environment, a noticeable characteristic of AR therefore is the need of linking the content of the information to the physical surroundings in which the AR is situated.

Milgram et al. suggest that there is relation between AR and Virtual Reality (VR) and argue that it is valid to consider the two concepts together [Milgram et al. 1994 p.2]. For doing so, Milgram et al. have provided a reality-virtuality continuum, which is described below.

1.3.1. Virtuality continuum

Milgram et al. uses the phrase *virtuality continuum* to describe the concept of a continuous scale ranging between complete virtuality (VR environments) to complete reality (real physical surroundings). Everything that lies within the two-dimensional plane, between these two extreme poles, is what Milgram in 1994 defined as *Mixed Reality* (MR) [Milgram & Kishino 1994] and any example of mixing virtual and real will fall within this range. As depicted in figure 1 below, showing Milgram's Reality-Virtuality Continuum, AR lies closer to *Real Environment* (RE) than to *Virtual Environment*. This is because of the fact that in AR the virtual augments the real as opposed to *Augmented Virtuality* (AV) where the real augments the Virtual, thus bringing AV closer towards *Virtual Environment*.

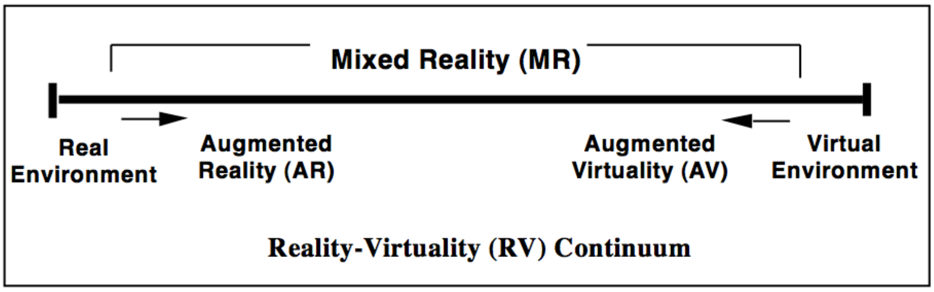
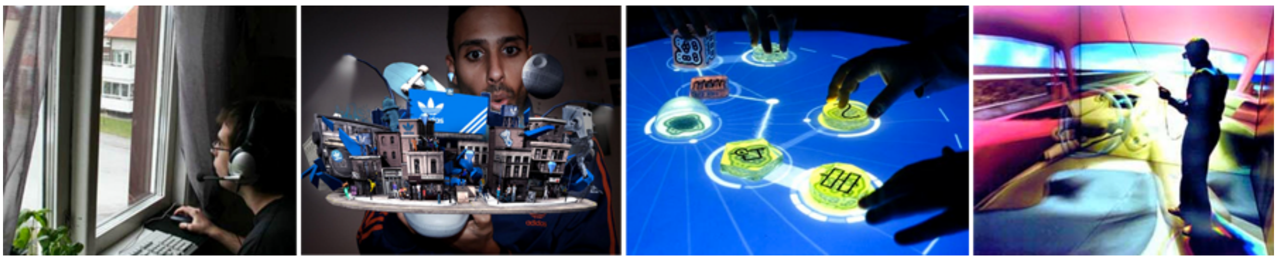


Figure 1: The virtuality continuum

The following concrete examples is a visualization of the Virtuality continuum's various elements, i.e. RE, AR, AV, and VR:



Picture 1: Real Environment. Picture 2: Augmented Reality. Picture 3: Augmented Virtuality. Picture 4: Virtual Reality

In this way AR can be thought of as a "middle ground" between VE (completely synthetic) and telepresence (completely real), situated closer to telepresence than VE on the Virtuality continuum's scale.

1.3.2. AR in relation to VR

The idea of comparing AR and VR, in order to define AR, has been adopted by a range of researchers and publishers in the field of AR. As an example, [Bimber & Raskar 2005 p.17] identifies the main contrast between the two to be that in AR, the real environment is not completely suppressed but in fact plays a dominant role, which supports the previously described need of a link between the virtual content and the physical surroundings.

[Höllerer & Feiner 2004] argue that AR and VR have in common that they provide an experience, which is to be explored interactively, but differ in AR's aim of supplementing the real world, rather than creating an entirely artificial environment [Höllerer & Feiner 2004]. Here it is interesting that interaction is introduced as a property of AR. [Poslad 2009] also acknowledges the necessity of interactive elements in both AR and VR, and elaborate that a more natural interaction is afforded in both situations, e.g. by using voice or gestures instead of the keyboard mouse interface of the PC. Furthermore [Poslad 2009] suggest that AR is making interaction in the physical world more virtual by digitally enabling relevant objects in the real world. From this suggestion it follows that AR in fact can be considered a user interface, which further has the potential of affording natural interaction. To fully clarify what can be understood by interface in this context, it is necessary to examine what it is interfacing. This will be done later by taking a deeper look into how AR concretely can be applied in various ways.

1.3.3. General definition

Azuma provides a commonly accepted definition:

"An Augmented Reality (AR) system supplements the real world with virtual (computer-generated) objects that appear to coexist in the same space as the real world. [...] We define an AR system to have the following properties: combines real and virtual objects in a real environment; runs interactively, and in real time; and registers (aligns) real and virtual objects with each other." [Azuma et al. 2001]

From this definition, three characteristics of an AR system can be identified:

1) It combines both the real and virtual content (within a real-world environment),
2) The system is interactive and performs in real-time, and
3) The virtual content is registered with the real world.

In elaborating on these points, it can be noticed that they contains the main aspects from the previous definition descriptions. To point one, it can be added that the virtual content must have strong link to the real environment. Throughout an AR experience, the user should be able to interact with either or both virtual and physical objects, but even though interaction obviously can have different characteristics no special requirement has yet been identified. To clarify what Azuma means by registration of the virtual content, such registration can be considered as the virtual content being aligned according to a real-world coordinate position [Azuma et al. 2001].

1.3.4. Thesis definition

Azuma's definition will be used as a general AR definition for this thesis. As the scope of the thesis is related to AR for the Smartphone platform, the term handheld AR will be used for describing any AR perceived through the display a Smartphone. Furthermore handheld AR will be thought of as an experience, which is enabled through the hard- and software of the Smartphone platform itself and not relying on other computational devices, situated in the context environment. In this way handheld AR falls within the broader category, mobile AR (MAR), with the distinct difference that MAR systems (MARS) also include AR setups, which uses external sensors and computational devices, and furthermore not necessarily has run on an handheld platform [Höllerer & Feiner 2004].

1.4. Requirements for AR systems

Having obtained a general theoretical understanding of what is meant by AR, a subsequent logical next step, in conducting any research related to AR experiences, is to clarify which elements and aspects are constituting systems that can provide such experiences, i.e. what the building blocks of AR are. Such knowledge should provide basis for understanding how AR are carried out from a technical perspective but also aid in delimiting the problem area, thus leading towards a concrete problem definition for this thesis.

1.4.1. Criteria for AR

ISMAR (International Symposium on Mixed and Augmented Reality) is an international event for the Mixed and Augmented Reality research community where carefully peer-reviewed conference papers are presented to reveal the latest developments within the field [web 10]. ISMAR is being held in North America, Asia and Europe on a rotating basis and Feng Zhou et al. have identified the following requirements for any AR system, based on experience gained through ten years of ISMAR, as well as by studying existing technology surveys [Zhou et al. 2008 p.2]. The factors that must be developed in order to provide an effective AR experience, includes:

1) Graphics rendering hardware and software that can create the virtual content for overlaying the real world.
2) Tracking techniques so that the changes in the users position can be properly reflected in the rendered graphics.
3) Tracker calibration and registration tools for precisely aligning the real and virtual views when the user view is fixed.
4) Display hardware for merging virtual images with views of the real world.
5) Computer processing hardware for running AR simulation code and supporting input and output devices.
6) Interaction techniques specifying how the user can manipulate the AR virtual content.

As described in the introduction, today's smartphones includes all means for enabling AR experiences. To account for the first of the above-mentioned factors, these devices have integrated graphics rendering hardware (GPU) as well as software for the synthetic content, e.g. the iPhone developer SDK's OpenGL ES for graphics [web 11] or OpenAL for audio [web 12]. In the same way smartphones have processing hardware (CPU) for computations related to the fifth factor.

Factor two and three are both interesting aspects of AR development and seems further to be related. The virtual content must be properly aligned within the users real-world view but when the user's position and view changes, also the virtual content must be re-aligned in order to appear fixed to real-world coordinates. Therefore the users location and orientation must be tracked in real-time and the virtual content registered into the world-view accordingly. Tracking and registration is according to [Wagner et al. 2010] a complex matter without any general single best solution as this depends on the character and location of application. For these reasons, the various tools and techniques for this area will be wholly analyzed when the context for the AR application, related to this thesis, has been settled.

The last factor concerns how the user can influence the virtual content of an AR application. Such interaction should necessarily also be context relevant and an analysis of possible interaction approaches and techniques will be carried out when the context and purpose of the application has been made.

1.4.2. AR building blocks

Bimber and Raskar take the approach of explaining the requirements for AR through general building blocks, which is structured in bottom-up hierarchical levels as illustrated in figure 2 below [Bimber & Raskar 2005 p.21]:

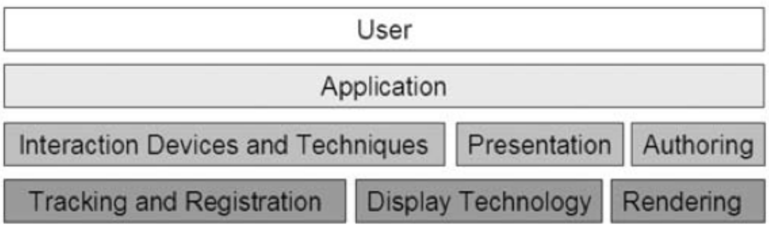


Figure 2: Building blocks for AR

According to the figure, the lowest and most fundamental level consists of the building blocks of *tracking and registration*, *display technology* and *rendering* respectively, without either of which AR would not be possible. This seems very logic, as the virtual content naturally must be rendered, overlaid to the user view via a display, and be positioned relative to the users current location and direction of view.

The second level is made by the components of *interaction devices and techniques*, *presentation*, and *authoring*. This level is argued to be more advanced because ideas and early implementations of presentation techniques, authoring tools, and interaction devices/techniques for AR systems just are emerging and still non-standardized.

By *application*, which fills the third level, Bimber and Raskar refer to the interface to the user as well as application context. Having composed the AR content, i.e. authored the virtual information, provided an input/output stream etc., it is at this level the user actually is having an AR experience and here AR gets applied to a real-world scenario. It thereby here the users, positioned at the top of the hierarchy, can take advances of what AR can offer.

By comparing the two different approaches in describing the required elements of AR, there are no contradicting elements to be noticed. Though, it is evident that the one, related to ISMAR, is concentrating more on the technical factors of AR system development whereas the latter puts more focus on context and usage of AR. The hierarchical illustration of building blocks, constituting AR and AR systems, reflects a focus on viewing AR as a human-computer interface, which can be applied to various scenarios. For such an interface to make sense, it further implies importance in how AR can benefit and be of use to the users.

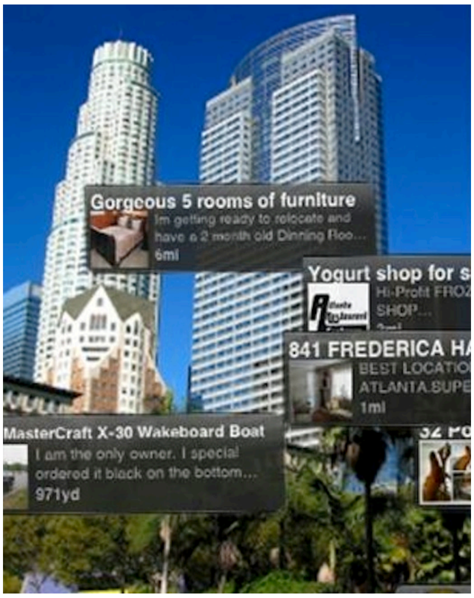
Inspired from the concept of viewing AR in terms of hierarchical structured levels, it seems beneficial to start by considering the user context and application purpose and subsequently proceed downwards through the levels when developing an AR application. Therefore questions like "*how can AR benefit the user?*" and "*which situations does this apply to?*" should be asked and considered. Before settling of a specific research topic, to be investigated with this thesis, a deeper look on how AR can provide such beneficial qualities for its users will be taken and described following sections. Furthermore this will be done through an attempt to categorize the different ways in which AR can and has been applied for various purposes.

1.5. AR application areas

According to Poslad, AR generally seeks to make interaction in the physical world more virtual by digitally enabling relevant objects in the real world [Poslad 2009]. The following sections will be an attempt to investigate and cover the various ways in which AR can make such interaction more virtual. The purpose is not to cover all possible applications but merely to obtain a general understanding of how AR can be applied. This will be done by dividing overall AR application fields into categories, which will aid in settling on a specific application area for the thesis to address, thus delimiting the thesis' problem scope.

1.5.1. Information systems

In the digital world, the Internet provides an increasing amount of information, which can be ubiquitous accessed via wireless networks. Even though such information can be related to the physical reality we live in, it is to a certain degree still external to our physical world. According to [Lalanne & Jürg 2009], AR holds the potential to bridge these two worlds whereby any kind of imaginable virtual information can be pulled into a given environment. AR information systems can theoretically provide users with the right information in the right context at the right time. The portability of Smartphone devices makes handheld AR especially suitable for such purposes. There are already various AR browsers applications for smartphones available, such as Metaio's Junaio [web 13] and Layers' Layar [web 14], which enables users to gather information about nearest restaurants, cinemas etc. simple by holding the devices towards their location. The virtual information is geo-tagged to specific locations' GPS position and users can use build-in search functions to specify which kind information, related to the immediate surroundings, to get displayed. In the same way Mobilizy's Wikitude [web 15] app lets its users see Wikipedia information about places around them. Another example is the Google Goggles visual search app for Android, which attempts to bridge the real and digital world by letting the users use camera snapshots for browsing their search engine [web 16]. The process, of adding extra virtual information to an object, can be referred to as *annotation*.

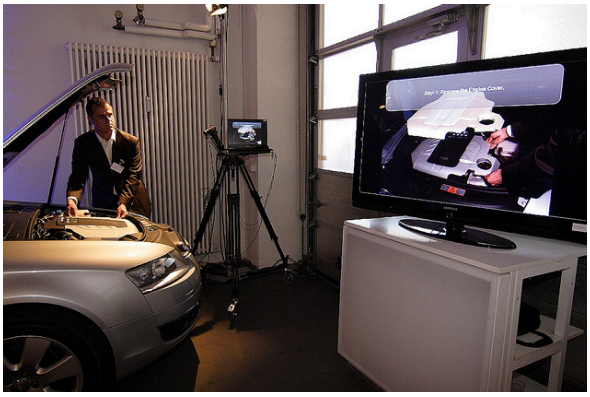


Picture 5: Junaio AR browser (left). Picture 6: Wikitude World Browser (right).

1.5.2. Assembly, maintenance and repair

Augmented Reality can also be used within the area of assembly, maintenance and repair of machinery etc. By displaying instructions directly on top of the associated equipment or machinery, the users attention is not divided between the guiding instructions and the task itself as it would be to for instance use a printed manual for guidance. Besides for enabling full attention on the task at hand, augmented reality holds the potential of displaying the relevant information perfectly aligned with the associated components thus making the guidance more intuitive. Furthermore step-by-step instruction can be shown as animations, which can make task-solving a lot easier to perform, but also make otherwise too complicated tasks possible to solve. As technology repeatedly advances, products in general are becoming increasingly more complex, causing assembly, maintenance and repair task of them to be accordingly more demanding and difficult. Therefore AR technology is likely to be used as new generation of instruction manuals in the future, both for private use as well as within industries. In relation to handheld AR, a significant aspect to consider is the fact that one hand naturally will be reserved for holding the actual device, leading only one hand available to work with. In cases where two hands are needed for solving a task, handheld AR is not appropriate and an AR solution, which is displayed with for instance HMDs, would be to prefer.

Examples of AR applications within this category could be the company Metaio's prototypes for car engine repair and maintenance such as the once depicted below:

Picture 7: BS off-Screen Library (left). Picture 8: Car Service of the future (right).

1.5.3. Military

For many years, military aircraft pilots have used Head-Up Displays (HUDs) and Helmet-Mounted Sights (HMS) to augment the view of the real world [Azuma 1997]. Hereby basic navigation and flight information can be provided as well as graphics, which aids in aiming by providing location information on targets. Mobile AR technology is also being used among land troops to provide information on enemy targets, dangerous areas to avoid and strategically plans, thus potentially reducing the amount of casualties.

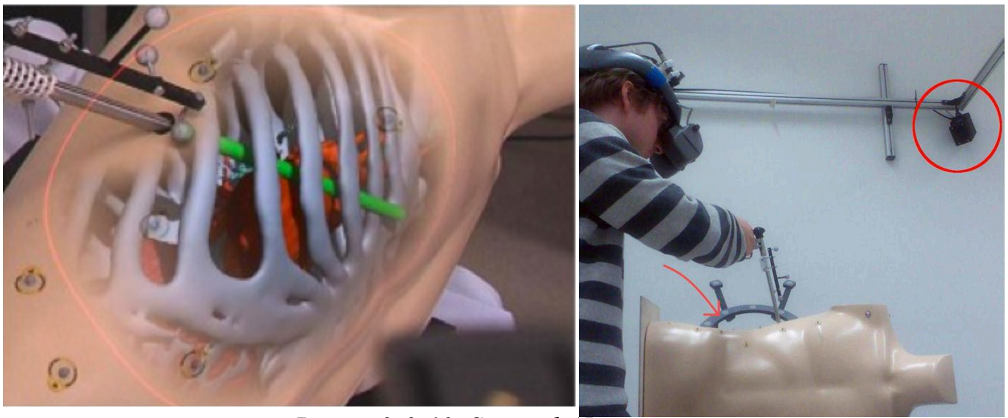
Augmented reality can also be applied to military training situations such as simulating large-scale combat scenarios and real-time enemy action with what is called Battlefield Augmented Reality System (BARS) [Van Krevelen & Poelman 2010]. As soldiers obviously do not use smartphones while in battle situations, there is no direct comparison for handheld AR in relation to military usage, but it is assumable that the concept of video-see-through on an external display could be applicable in for instance augmenting the sight of a weapon.

1.5.4. Medical

Within the medical field, Augmented Reality could serve as a visualization and training aid for surgery. Using the datasets of a patient, collected with non-invasive sensors, as augmented content over the patient, could provide surgeons with „x-ray" vision in real-time [Azuma 1997]. This could reduce the amount of incisions to a minimum thus making operations safer and less time consuming.
As with AR for assembly, maintenance and repair, doctors can use AR for step-by-step guidance without drawing the visual attention away from the patient. This could for instance depict the exact path for a forthcoming incision and could be beneficial both in real and training situations as well as enabling remote surgery.

General for AR within the medical field is the crucial need for a very accurate AR implementation where the augmenting content is perfectly aligned with the real-world objects, e.g. the patient. Otherwise guidance could become misguidance with possible lethal consequences. As it was the case with military related AR, there is no logic in using handheld AR for medical purposes.

An example of a concrete AR application, which addresses the medical field, is a surgical AR navigation system made by a research group from UNC Chapel Hill [Azuma 1997]. For the purpose of image-guided surgery, this project was based on transformed ultrasonic scans of women's wombs into visual 3D representations on real-time, which enabled doctors to see the fetus, through HMD's, as virtual overlay on top of the patient. The system has been used regularly at Brigham and Women's hospital and findings has shown that the system has been able to shorten the average length of surgery from eight to five hours.



Picture 9 & 10: Surgical AR navigation system

1.5.5. Education

It is well documented that multi-modal stimuli can be attractive in learning situations [Rowe 2991]. The virtual overlaying content, provided by AR systems, can have both visual and auditory character and further such systems can afford interaction, which is kinaesthetic, e.g. haptic or physical. It is therefore assumable that the multi-sensorial experiences, obtained through AR, can be beneficial in an educational context.

[Azuma et al 2001] argues one of the basic goals of AR to lie in enhancing the users perception and it logically follows that an enhanced perception can lead to a broader understanding. Overlying a physical object with relevant information about it can for instance be a tool to give a better cognitive understanding of the object. As apparent in the description of AR for military and medical purposes, AR furthermore is applicable to various training scenarios.

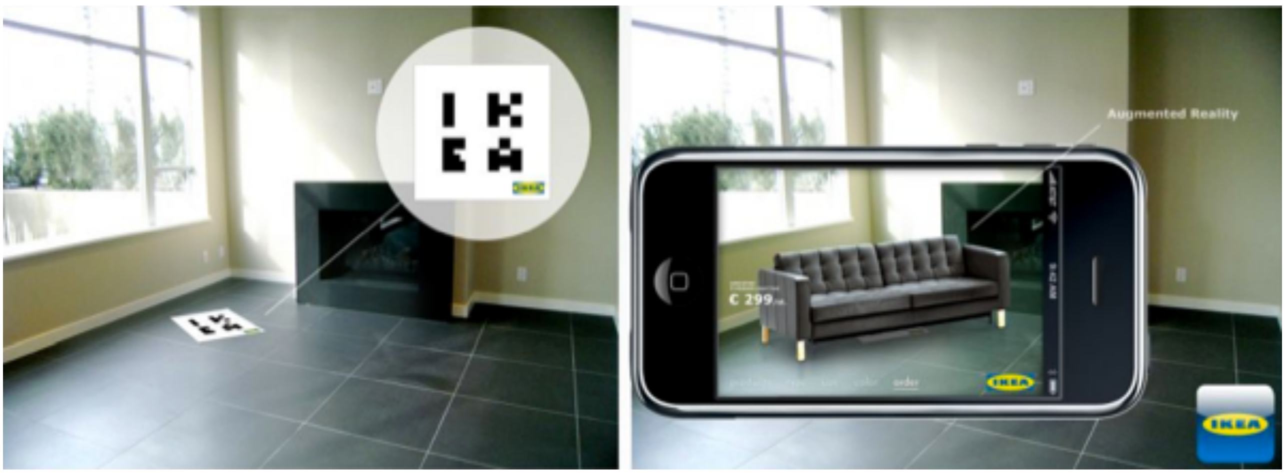
Another aspect of educational AR is its potential for including more fun, interesting or entertaining element in the process. Virtual multi-media output from AR systems, together with the variety in interaction possibilities, seems ideal for enabling edutainment applications.

Handheld AR might have distractive influence in a traditional classroom setting but in relation to mobile learning, AR might enable new possibility. Having a virtual teacher with you anywhere at anytime could open new doors for obtaining knowledge.

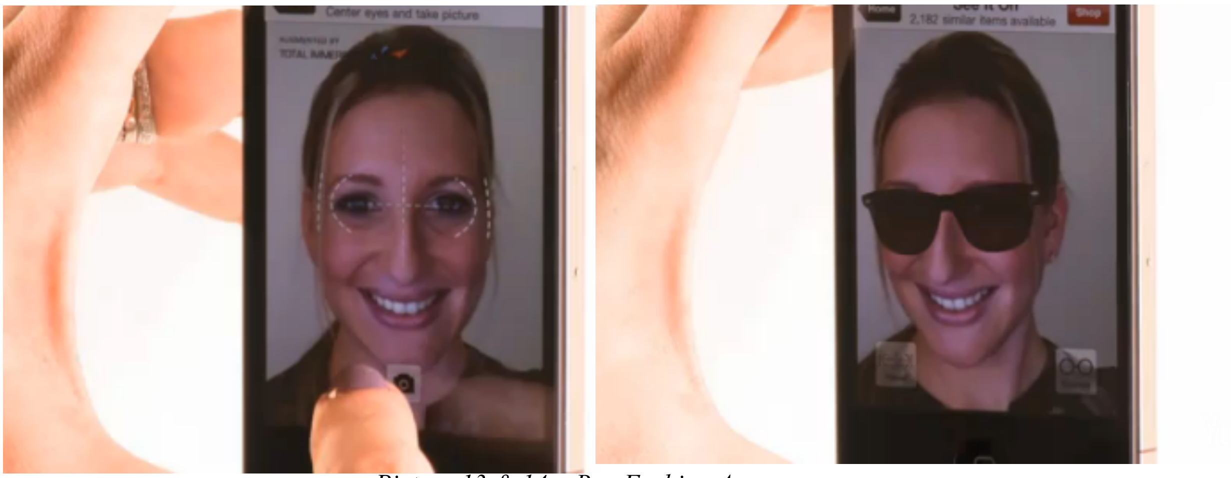
1.5.6. Marketing and Sale

AR technology can also be used in relation to the advertising and selling of consumer products. At Point of purchase (P-o-p), AR can depict products in 3D without the need of unwrapping the product from its packaging or assembling it. A nice example of such an application is the kiosk-based AR stand from LEGO, which enables possible consumer kids to see a visual 3D model of the associated LEGO product, directly on the wrapping, when pointing it towards the kiosk [17]. Besides from the see-what-you-buy aspect, this kind of kiosk-based applications also can attract attention towards products due to its novel and entertaining presence, which affords interaction and activity. For the same reason, AR also can be used for advertisement purposes related to event marketing, such as fairs, festivals etc. Examples of other applications of consumer related AR could be online web implementations or interactive outdoor billboards, posters, banners etc.

From the consumers' perspective, AR can be used to assist in choosing which product to buy. By displaying detailed product descriptions, etc., decisions about selections can be made easier and better judged. Due to the mobile properties of smartphones, handheld AR is especially good for this and can furthermore be exploited for showing directions towards locations of relevant products. Even though this kind of information might be very useful, it is more interesting that AR holds the innovative potential of enabling novel display of consumer product within their right context and natural environment. The Swedish company, Ikea, has for instance launched an AR application, which allows consumers to view virtual representations of their furniture as overlay within the natural settings of the household [web 18]. Using AR together with the concept, known as the magic mirror paradigm, can provide another example of displaying virtual content within its right context. More specific, this is the concept of letting the camera and display device act like a mirror, which lets the user see a reflection of him together with the VOs [Fiala 2007]. A concrete example of such concept could be eBay's Fashion app, called "See It On", which lets users virtually try on various glasses and sunglasses by letting the iPhone function as magic mirror. The company behind the app, Total Immersion, implemented it by combining AR with face tracking and the app had its public release in February 2011 [web 19].



Picture 11 & 12: Ikea AR App



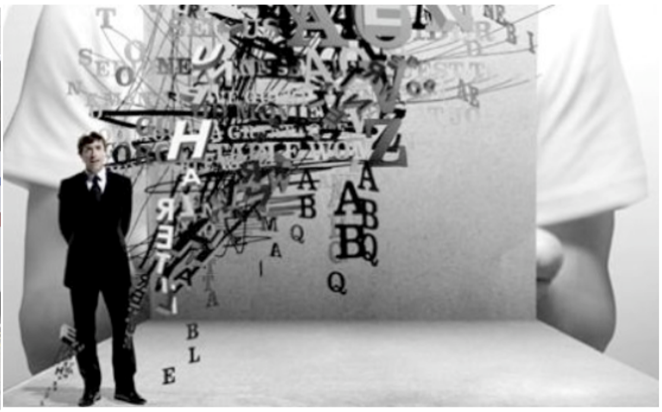
Picture 13 & 14: eBay Fashion App

1.5.7. Entertainment

As AR and especially handheld AR is an emerging technology, the novelty of engaging in such applications can cause a degree of excitement in the users, which is sometimes referred to as the *Wow* effect [web 20]. When users are momentarily fascinated, surprised, etc. they are also enrolled in an action where they are being entertained.
Besides from the entertaining factor of the *Wow* effect, there is a range of ways AR can be applied for entertainment purposes. It could be for the fields such as contemporary theatre, dance performances or interactive art.

AR can be used for augmented the space within museums as well as virtually bringing history back to life in its previous locations, which Museum of London's Augmented Reality app, *Streetmuseum*, is an example of. With this app, users are guided to various sites around London where historical photos, from the exact same places, can be displayed over the present view of the site [web 21]. Another example of such a touring application is the *Archeoguide* project, which allows visitors of certain cultural heritage sites in Greece to see complete visual 3D reconstructions of the ancient Greek buildings that now lies in ruins [web 22]. In the same way the synthetic overlay can obviously have fictional character and be composed as for instance narratives. The possibilities within using AR to alter the users' perception of their immediate surroundings for entertainment, seems endless.

AR can also be used in combination with traditional mediums such as magazines/books/papers, TV broadcasting or movies. As example, the 2D nature of print could appear in 3D, which where the case with a magazine cover from Esquire in 2009 [web 23]. In relation to sports broadcasting, information about athletes could augment sport events in real-time during. In the same way, AR could be applied to movies and thereby possible open up for new interaction possibilities and experiences for the spectators.

Picture 15: Streetmuseum (left). Picture 16: Esquire AR cover (right).

An area of AR for entertainment, which seems very promising, is gaming. Using the physical surroundings as stage for computer games invites to novel forms of gaming experiences as compared to the ones obtained from traditional PC games. [Haller et al. 2007 p.383] argues that AR gaming might very well succeed, where Virtual Reality gaming only had limited success, because of the following three aspects:

1) In VR the player is always left with the feeling of existing in a synthetic world as opposed to AR gaming, where the vast majority of the player's view is the physical world and only the introduced game pieces require computer-generated graphics.
2) Being able to see the physical world provide the player with a natural orientation sense and an ability to move freely in the combined physical and virtual world while avoiding obstacles such as chairs, trees, and other people.
3) The physicality of moving in open spaces is generally appealing to players and allows them players to understand and experience the game more intuitively, as when compared to sitting in front of a PC.

In relation to gameplay, it is imaginable that AR could enable real-world objects to be included as element for progressing the plot of the game. An example of an AR gaming system, which can be played both as single- and multi-user, could be the software and hardware connected to the AR Drone from the French company called Parot. By controlling the drone with an iPod or iPad, the player can navigate the drone around open space meanwhile having virtual content superimposed on top of the view as seen from the drone through cameras attached to it. Dependent on the software being run on the controlling device, the player can for instance battle virtual combat-flights or other drones in a first-person-shooter (FPS) manner [web 24].


Picture 17 & 18: AR Drone.

1.5.8. Collaboration and social interaction

AR can also be applicable within a multi-user context such as collaborate task solving or co-located collaboration. It can be used for shared object viewing where the VOs as well as real-world objects with augmented content attached to them. In this way AR can be used to enhance a shared physical workspace. Besides for face-to-face interaction, AR can also support remote collaboration, where AR potentially could aid in task solving as well as in enhancing the feeling of copresence in an augmented video chat or conference. An example of such conferencing system was an AR application, which allowed virtual images of remote collaborators to be overlaid on multiple users' real environments [Zhou et al. 2008].

In relation to handheld AR in a multi-user context, the interesting possibilities are those provided by the mobile nature of the Smartphone devices. This ubiquity enables AR's qualities in aiding collaboration to be applicable anywhere within the public domain where PC's might not be available. Besides for assisting in collaborative task solving, AR holds potential to support mobile social interaction. The previous mentioned AR browsers already offer functionality for providing and sharing location specific virtual user-generated information, such as personal experiences and recommendation, i.e. Layar's *BuildAR* [web 25], Wikitude's *me* [web 26] or Junaio's *My Channel* [web 27].

1.5.9. Hybrid applications

Comparing these various AR applications areas, it is apparent that many of the described categories are overlapping each other, e.g. the LEGO kiosk stand, which was a mixture of marketing and entertainment. In the same way it is assumable that the qualities of AR could be strengthened through such combinations. Using educational AR, together with entertaining AR, could provide basis for rich edutainment experiences. Including AR supported collaboration into learning situations might further enhance the learning outcome. Accordingly, such collaboration could open for new gaming experiences when included into the gameplay of AR games where for instance face-to-face interaction or social interaction could be the driving force. Based on such reflections it seems beneficial to use a broader view and understanding of the variety in qualities, offered by AR, when integrating AR into a specific application. The described application areas will therefore not be used as narrow "boxes", in which an AR application should fit into, but merely be used as inspiration and knowledge base for understanding what it is AR can do in various contexts.

1.6. Choosing application area

As apparent through the investigation of various application areas, AR can be applied for great variety of different purposes and benefits. The intended goal with this thesis is not to review or judge, which application areas are more useful than others, and the broader understanding of possible application areas should merely serve to understand the strength and potential of AR. As seen, for instance with hybrid application areas, this strength and potential is not necessarily restricted to a specific area but rather applies to multiple areas in different ways. To focus and concretizes the research direction, it is however found appropriate to narrow down the scope of the thesis to target a specific application area. What is found interesting, is to engage in an investigation into how AR technology can be applied to a new but specific context / scenario and furthermore to evaluate the outcome and impact of applying new technology in such a novel way.

For personal reasons, based on a wish to engage in a career within the advertisement industry, it was decided to invest the outcome of applying AR technology to this field, i.e. choose marketing/sale as application area. Furthermore, the author shares the opinion that it is important for the advertisement industry to adapt to new technology, such as described below, thus finding it both to be an interesting, as well as a highly relevant, research direction for the thesis.

1.6.1. The need for the advertising industry to adapt to new technologies

Close to the end of the 20th century, as well as the very beginning of the 21st century, there was an extensive amount of research done within an area, defined as Interactive Advertising in 2000, due to the fact that internet was becoming widely available to the public [Lombard & Duch 2001]. As compared to traditional mediums such as TV, radio or books, this opened up for new possibilities of making the consumers play a more active role in relation to advertisement. Consequently this raised new challenges for the advertisers, who had to renew their strategies in order to benefit from the possibilities, which was offered by the new technology. As Lombard and Duch stated:

"*New communication technologies are creating new challenges for the advertising industry. […] As communication technologies evolve, becoming more interactive, personal, and sophisticated, advertising is being forced to evolve as well*" [Lombard & Duch 2001]

In the same way as the Internet was it at the time, AR can be argued to be a new communication technology of our present time, and another relevant parallel can be drawn in the fact that they both afford human computer interaction (HCI). Another interesting point in Lombard and Duch's work, concerning interactive advertising, is that they claim that early research and theory, regarding the concept of presence, is providing a valuable framework for developing effective advertising techniques and messages in the new media world. As inspiration for this thesis, it raises the question on whether the key in evaluating an AR application could be found in evaluating the user-perceived presence? However, there are various forms of presence, e.g. spatial, social etc., and it is too early in the research process to answer such a question. Furthermore, AR related presence seems more complex than the one related to complete virtual experience, as an AR experience per definition is constituted by a combination of virtual and real. Additionally, a prerequisite for presence, according to Lombard and Duch, is that the technology providing the experience should be transparent, and whether this is possible for AR technology should therefore also be further investigated.

1.7. Problem formulation

"How can cutting edge handheld Augmented Reality technology, utilized within the public domain, affect the user experience when applied to an advertisement context?"

2. Conceptual analysis

The following chapter serves the purpose of developing a concept for the thesis application, which corresponds to thesis problem formulation.

2.1. Defining cutting edge within handheld AR technology

To understand the current state of the art within cutting edge AR technology for smartphones, it is relevant to take a brief look into how the field has developed over the years.

2.1.1. Brief historical overview of handheld AR

AR can be dated back to the late 1960s when computer graphics pioneer Ivan Sutherland constructed the first computer-based head mounted display (HMD) and Bell Helicopter experimented with augmenting pilots' view for navigating in the dark [Ekengren 2009 p.11]. Throughout the 1970s and 1980s various academic computing research and commercial technological development contributed to the evolution of augmented reality but it was not until 1992, the scientist Tom Caudell coined the term, *Augmented Reality*, for describing research related to airplane manufacturing by the Boeing company. It was also during the early 1990s that the first AR products appeared for addressed the needs of the medicine and engineering fields [web 28]. Towards the end of 1990s, AR had become a distinct field of research and it was also here Feiner et al. developed the first prototype of a mobile AR system. This was a touring system, which provided 3D information about buildings and artefacts in a mobile setting [Feiner et al. 1997], and the prototype is depicted below:

Picture 19 & 20: A Touring Machine.

In relation to handheld AR, Möring et al. developed the first AR system to run on a solely on a consumer cell phone in 2004 [Möring et al. 2005]. This was I video see-through system, which could track fiducial markers, like the QR code depicted below, and render 3D graphics accordingly. In 2004, also the first handheld AR application for Personal Digital Assistant (PDA) devices appeared, in the form of the *Invisible Train*, which was a collaborative multi-user game based on tracking QR codes [web 29]. Around this period, developers toolkits, such as the ARToolkit [web 30] and the Studierstube ES [web 31], where released to enable rapid and more straightforward prototyping of handheld AR applications. These tools where all based on using QR codes (also sometimes referred to as LLA markers [web 32]), and became a standardized approach for developing handheld AR application in the years to follow.

Picture 21: A QR code

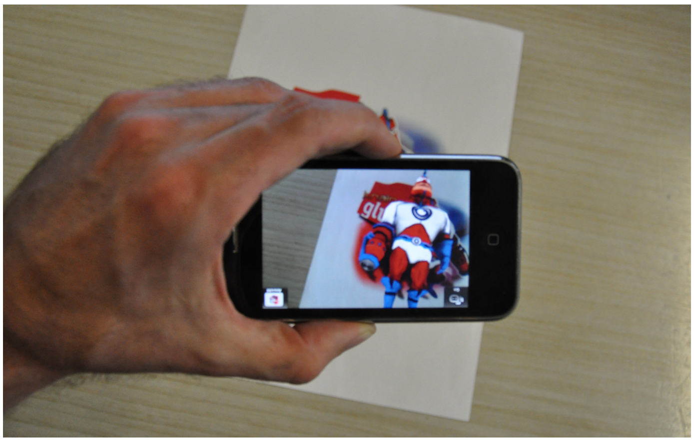
More specifically, QR codes are two-dimensional binary pattern in black and white, which are placed on a planar surface. Once is identified, the position and orientation of the pattern, such as the one depicted above, is followed for each frame This technique requires only low computational costs and are thereby attractive for platforms with limited computing power, e.g. mobile phones [wagner].

It was not up until recently it became possible to enable tracking for handheld AR applications without being dependant on the artificial QR markers. In 2010, the company called Total Immersion announced the first commercial AR implementation for smartphones, which used markerless tracking [web 33]. Instead of relying fiducials, the markerless tracking approach is using computer vision algorithms to track natural features of for example a photograph or poster. A huge benefit with avoiding QR codes is that they can be considered intrusive for the physical surroundings, e.g. these fiducials could invoke aversion if seen all over the cityscape. From a visual perspective, QR codes do furthermore not have any link or connection to the actual physical reality, which is to be augmented. Besides, markers can occlude the user's vision, e.g. hide important parts of equipment in an AR repair scenario. In this way there is many benefits of with markerless tracking, which can be can be considered cutting edge, within handheld AR technology, of today.

"The holy grail of mobile Augmented Reality is to be able to track natural features in a robust way without the need for fiducials" [Ekengren 2009 p.42]

2.1.2. Hands-on experience with markerless Tracking

Metaio's previously mentioned Junaio AR browser also provides features for natural feature tracking (NFT), which they, based on the metaphor of gluing virtual content to an image, refer to as Junaio GLUE [web 34]. To obtain a broader understanding of the state-of-the-art, within handheld AR technology, it was decided to take a deeper look on Junaio GLUE, i.e. acquire hands-on experience with markerless tracking. Their actual tracking principles and algorithms are patented, thus unavailable for the public, but by signing up as a Junaio developer it is possible to create a functional markerless tracking application, which can apply virtual content, in the form of text or 3D models, on top of a chosen pictures. As Junaio functions by sending server requests from smartphones, running Junaio software, all frontend development is therefore done in PHP scripting language. More specifically, the Junaio developer registers a Junaio GLUE channel and uploads reference image, 3D models, 2D graphics, audio, text information, etc., to the Junaio server, together with a general PHP file [web 35]. Anybody, who is logged into the developer's channel online, can subsequently use this exact image reference for registering and displaying the virtual content, provided by the developer, as handheld AR. The following pictures are personal visual documentations of the testing experiences of Junaio GLUE on a 3GS iPhone:



Picture 22: Testing Junaio Glue

In evaluating Junaio GLUE, the general impression was, without having too much to compare with, that the markerless tracking worked vey well and seemed both stable and flexible. The provided 3D model was perfectly aligned with the reference image and the update rate (camera framerate) appeared sufficient fast, for the model to stay aligned, when the Smartphone was moved at a slow pace. Another noticeable characteristic, was the ability for the tracking to work even when the camera where tilted at certain degrees, in respect to the reference image, i.e. when the image where captured as being "skew". Furthermore, the implementation allowed for certain freedom in distance between camera and image. A downside, with using Junaio GLUE for markerless tracking, is that the developer has limited freedom in terms of being restricted to use the various functionality provided by the Junaio developing environment. If additional functionality should be required, Junaio GLUE would thereby not be an optional implementation tool. Whether this is the case for the thesis' implementation will be considered at a later point. Furthermore various implementation methods, for utilizing markerless tracking, will be examined through a technical analysis.

2.1.3. Cutting edge definition

Cutting edge, within handheld AR technology, will for the scope of this thesis be defined as technology, which utilized markerless tracking for enabling AR on smarphones. In respect to the problem formulation, it is consequently a criterion for the AR application not to build upon a marker-based tracking technique.

2.2. Authoring tool for AR development

2.2.1. DART

The Designer's Augmented Reality Toolkit (DART) is a framework and developer's tool made at Georgia Institute of Technology, Atlanta, which aims to support development of augmented, virtual, and mixed reality applications [Haller et al. 2007 p.178]. The implementation tool, provided with DART, dates back to 2007 and was a C++ written plug-in for extending the now non-existing company Macromedia's Director developing environment in order to enable fast prototyping of AR applications. Hereby this implementation tool, based primarily on the use of QR markers, is not cutting edge of today. However, the DART framework also provides a set of excellent guidelines and suggestion to the phases of making concept, design and evaluation of AR applications. These are relevant and useful for this thesis and the following will be an outline of the guidelines, which consist of four stages:

Stage 1 - Exploring the Ideas:

Due to the fact that an AR application generally is meant to augment the surroundings of a specific environment and thereby is *in situ*, DART points to the importance of considering the contextual environment already from the very beginning of the development. Incorporating the context of space to the process of generating ideas for a concept makes sense and will be adopted for the development related to this thesis. Gathering data about the location where the experience will be situation can according to DART be done by for instance shooting video footage of the location. Further this video can be used as background for sketching various ideas as overlay on. Pictures can also be used for this purpose but an advance with video lies in the possibility to sketch storyboards etc. over time.

Stage 2 – Populating the Virtual World:

This stage is entered when the concept has been completed and is concerned with the development of the virtual content. According to the DART guidelines, putting too great effort into making the virtual content without proper testing can be a mistake, which can result in waist of time if the content does not convey the application intent and must be changed or disregarded. Instead of creating detailed and compelling virtual content at an early point, they suggest evaluating the user experience before too many resources are spent on the final content development, thus making the design phase into an iterative process. Inspired from the film industry, DART argues for using animatics and storyboard in pre-production, which for instance can be applied on video footage in the same manner as in stage one. Here they introduce the concept of using *sketch actors*, which is illustrations, e.g. hand drawings, of virtual and real-world objects that are placed in the storyboard to depict how these objects will be displayed at a given times.

In reflecting on the guidelines for this stage, there is a logic in the fact that testing the user experience, prior to spending an extensive amount on time on creating the virtual content, would reduce the risk of having to re-do or discard it. However it is uncertain how close to the real AR experience such film inspired approach can give the user. The dilemma seems to be that the video approach might not be sufficient for inducing the intended user experience, as for instance the real-time interaction is diverse as well as the representation of the virtual expression is poor. Even though DART does not provides other means for measuring the user experience at this stage, it suggest the use the Wizard of Oz (WOz) simulation method for such evaluation. In this method, a "wizard" operator plays a role in a work-in-progress computer system by manually simulating sensor data, contextual information, or system intelligence [Dow at al. 2005].

Instead of merely making the design phase iterative, as recommended by DART, another option would be to make the whole application development process iterative. A possible advance of this could be that the user could be able to have an actual AR experience even though the application concept or design still not is fully completed. In this way the user experience will be closer to the one intended, and thereby more directly testable without having fully completed the virtual content in details, i.e. testing an AR application with a for instance a functional tracking and registration implementation but only 3D models with low polygon count (low-poly models) as virtual content. Another relevant aspect to be noticed from stage 2 is that the video approach could to be an attracting way of communicating design ideas in situations where numerous designers are involved in creating the application design together.

Stage 3 – Developing the Application:

As suggested by the description, this is the stage where the application is implemented, i.e. the virtual content is combined with the implementation of the actual AR technology such as tracking and registration. Here, the DART offers a set tool for implementation but does not provide any general guidelines, which seems useful for this thesis.

Stage 4 – Evaluation and Deployment:

More interesting would be how the DART framework approaches application evaluation but once again the offerings mainly are related to the implementation toolkit, e.g. debugging tool for common implementation problems related to AR as well as data visualization tools.

2.2.1. APRIL

The Augmented Reality Presentation and Interaction Language (APRIL), is another authoring tool, made at Vienna and Graz Universities, Austria, in 2004, which provides concepts and techniques for AR application development [Haller et al. 2007 p.138-159]. In the same way as with DART, the APRIL implementation tool itself builds on an existing developer environment, in this case the Studierstube runtime system [web 36] and has not been updated to match the state of the art within AR technology anno 2011. But interesting and relevant about APRIL is that it also provides a language with root in theatrical terminology, as well as a corresponding component model, which can support the general application development process. These working tools has been carried out by basing them on matching to uniquely identified properties of the AR paradigm, which are as Real-world interfaces, Hardware abstractions, Runtime engine, and Authoring workflow, as summarized below:

Real-world interfaces:

A unique property of AR is identified in that the real world is included in the users perception of the application space. The real world context for the application should according to APRIL be taken into account during development, which is very much in accordance with DART's guidelines for *exploring the ideas*. To do this, APRIL borrows from theatrical concept of a *stage*, which in it new context is a spatial container that can be linked real world places, real-world object, or to the user's display. More specifically, the stage is supposed used as a frame for considering what should be included in the implementation. In relation to the real-world context property, APRIL points to the need of considering real-world objects as an integrated application objects during implementation, e.g. to model a real-world object that is involved in interaction even though it does not have to be rendered as 3D graphics. The stage concept will be further explained later together with the APRIL component Model.

Hardware abstractions:

Hardware abstractions build upon the idea that AR authoring, in order to be flexible, the application content must be considered separately from all elements related to the actual system that the application will run on. Hereby elements, related to the virtual content, interaction or the system, can be changed or replaced without influencing the general application, e.g. the tracking method can be changed without the need also for changing the structure of the virtual content. Hereby is it possible to divide the development of various application aspects between multiple developers and designers. By considering hardware and hardware setup as an abstraction, AR application concepts can furthermore easier be transferred to other contexts and environments with different conditions. Another benefit is that early prototypes can be carried out desktop computers and later ported to other computational platforms, which can make early prototypes in controlled environments possible.

Runtime engine:

Runtime engine relates to the fact that an AR application obviously requires a combination of hard- and software in order to be utilized. The proposed implementation method, based on the Studierstube environment, is due to the criteria for this thesis not relevant and will intentionally be left out.

Authoring workflow:

APRIL provides a set of tools to structure the workflow for authoring the application concept with respect to the idea of separating virtual content and system. As with DART, APRIL point to the necessity of making the application development into a incremental and iterative prototyping process due to the complexity and experimental nature of AR as a medium. Even though these tools, on a source code level, are irrelevant for this thesis, it is interesting and inspirational that APRIL embraces the use of UML statecharts to describe the general application functionality. This seems to be an attractive way to get a clear overview of the application, as a system, prior to the actual code implementation. Furthermore it could make the actual coding implementation more straightforward and this approach will therefore be adopted for the thesis' application. For more information on UML statecharts, the reader is referred to Appendix A.

The APRIL language:

As mentioned earlier, APRIL has adopted terminology from the theatre world to explain the application. The term, *story*, is used for representing the temporal structure of the application. The *story* is composed from various individual *scenes*, in which the *actor* (the user) can *interact* in order to executer a sequence of predefined *behaviours,* whereby the *story* progresses. These terms are the main components of the APRIL language terminology and their relation can be depicted in APRIL's main component model below.

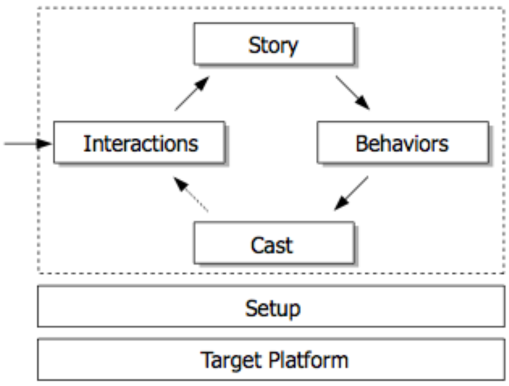


Figure3: The APRIL component model

As seen in the model, the *story* is thought of as a separate layer on top of physical soft- and hardware setup and the chosen platform so each layer can be changed without influencing the other. The arrows on the model represent the direction and cause of change in events, i.e. new *behaviours* make the *actor* perform new *interactions* and so on. Whether or not it could be beneficial to apply the APRIL's theatrical to the application development process, related to this thesis, will be decided when an advertisement purpose and message has been made.

2.2.1. Applying the authoring tools

Through the descriptions of the DART and APRIL authoring tools for AR development, a variety of interesting aspects has been identified and related to the thesis context. It is apparent that both DART and APRIL point to the benefits of considering the contextual environment already from the very beginning of the development process, and such an approach will be adopted for the thesis methodology. But how suiting these tools generally are, is however believed to be application dependent. As an example, APRIL's theatrical language might be less appropriate for explaining the development design of simple Apps with minimal user interaction. It is therefore too early to tell how to use these tools, but after the application concept has been concretises, it will be continuously described how and to which extent they has been used for inspiration.

2.3. Defining the application

2.3.1. Field collaboration

As advertising is beyond the scope of the author's education and competences, an attempt to provide a realistic advertisement concept for the handheld AR application would be hard, if not impossible. Building the concept upon research and theory within the field might prove to be an acceptable approach, which could be sufficient for the research purposes of this thesis. But to fully answer the problem formulation's question on how AR can be used for advertisement, it is preferable to seek expertise from experts within the advertisement industry when developing the application concept. Within this industry, Art Directors (ADs) are specifically employed for the purpose of generating application ideas and concepts for advertisement campaigns and strategies. Besides from being able to provide a realistic advertisement concept, ADs might also have general or specific suggestions on how handheld AR technology could benefit advertisement, if considering AR during the brainstorm process. Such collaboration could thereby contribute to a better thesis, and furthermore time and resources could be released for the author to focus on utilizing the actual technology as well as examining the result of applying novel technology to the field of advertisement.

Through an Internship, conducted during Medialogy 9th semester at Neue Digitale / Razorfish in Berlin, the author has gained access to resources from within a professional advertisement agency. An obvious choice was to enrol in collaboration with this company and such agreements where therefore established through a meeting with Technical Director, Paul Schmidt, and Senior Technical Architect, Mathis Moder, at Neue Digitale / Razorfish. The wanted outcome from the collaboration was from the author's side mainly related to specifying a concrete advertisement message and concept with relevance to AR technology. Another desirable outcome of such collaboration could lay en the company's ability to provide pre-designed graphics for the virtual content. Frome the agency's side, the wanted outcomes was the possibility of showing the finished AR application to potential clients or adapting it for an actual company advertisement campaign.

2.3.2. Expert knowledge exchange (with ADs)

As an initial step in clarifying the advertisement message and intended user experience, it was decided to make an unspecific enquiry to the agency's team of ADs, consisting of five employees. Having been informed that sending concept request to multiple ADs and subsequently choosing the most attractive outcome idea was the common procedure for concept development, it was found appropriate to follow this approach in obtaining an advertisement concept. Therefore an e-mail, summering the problem area and scope as well as a general introduction to AR, was send to all members of the AD team together with an enquiry for a suiting advertising concept for such an application. The intention was subsequently to study the returned concepts, i.e. comparing them for tendencies and determining how to narrow the data down to a single concept. Unfortunately the outcome from the enquiries was not fruitful, and after five days of waiting, having sent a reminding e-mail but received neither concept ideas nor feedback, it was decided to change strategy in pursuing a concept. Afterwards it was chosen to address two ADs, in person, to ask if they would be willing to be interviewed for the purpose of creating a concept. Later, when the enquiry got accepted, a date for the meeting was arranged.

As the ADs are trained in the process of generating such ideas, a consideration, prior to the meeting, was not to interfere with their usual working routines and let the process run as if it was a usual company project. For this reason a structured qualitative interview did not seems to be an appropriate method for conducting the meeting. However, there was a series of research questions, which had to be considered in order to use the application for answering the thesis' problem formulation. So completely renouncing control over the outcome of the meeting could on the other hand not be an option. Merely having an concrete advertising concept provided would not necessarily reveal the intended message and user experience for it to convey, so in planning and preparing for the meeting, it was decided to specify these research questions, i.e. clarify the necessary outcome of the meeting, and use semi-structured qualitative interview, with non-leading questions, as method for the meeting. The following sections will describe the reflections, related to the sought after meeting outcome, as well as the chronology for meeting to be carried out.

2.3.3. Structuring the meeting

Before engaging in the actual brainstorm process for generating a concept idea, a necessity would be to brief the ADs about the AR technology and general possibilities provided by it. Together with a general introduction to the thesis direction, this should create basis for identifying the most suiting client for such an application.

By researching previous conducted projects, at Neue Digitale / Razorfish's website [web 37], it was discovered that the agency previously had made a couple of AR application. To get closer to finding an appropriate way testing how handheld AR system can be used for advertisement purposes, it would be interesting to know if these previous application had been evaluated, and how. Therefore the initial questions was prepared:

1) *Has the previously conducted AR applications been evaluated?*
2) *How?*

Making the AR application concrete, necessarily means addressing it to a chosen client but to match the criteria of situating the advertisement in the public domain, also the context needs to be taken into consideration. As apparent from both the DART and APRIL, it is beneficial already to consider the actual environment from the very beginning of the AR application development. Therefore it would be appropriate to determine a location for the application prior to the brainstorm process, thus leading to the following questions:

3) *Could you imagine one of your clients whom it could make sense to use AR to advertise for?*
4) Could it be relevant to advertise for this client within the public domain?
5) Anywhere in particular?

Having clarified the addresser of the advertising message (the client) as well as an application context (within the public domain), the fundament for starting the brainstorming process would have been laid. Thus the following questions should invite for such process:

6) How can a handheld AR application (for smartphones) be used in advertising for this client?
7) Which message?

As mentioned it is not sufficient merely to have a concrete concept provided without understanding underlying reflections and intentions. To be able to test if the application is conveying the intended message, the following question were prepared:

8) Is there an overall idea behind the message/application?

This question obviously addresses the general intentions to be conveyed with the application, and such an open question should aid in obtaining a broader understanding of the ADs' reflections about the concept. Considering the APRIL approach of explaining an AR application according to theatrical terminology, this could be related to understanding the idea behind the *story* as a whole.

Having understood what to tell, the subsequent steps lies in clarifying how to do it. For an AR application, virtual and real word information is the tools and delivering such message and experience to the user, leading to the following question:

9) Which information is needed to deliver the message?

Getting closer to specifying how the virtual information can convey the message, an additional question could be ask:

10) Which kind of virtual content (media) is involved, e.g. text, graphics, animations, sound?
11) Is there any specific aesthetic related to this content?

As apparent from the AR analysis, described in the previous definition of AR, a fundamental part about AR is that it lets the user interact with the information. The user's role in relation to the message should therefore be considered, which yields a question about interaction:

12) How should the user be able to influence the information flow (e.g. interact with the information or choose which information to see at given times)?

To be apple to understand the application structure and visualize it through an UML State Machine Diagram, formerly called an UML Statechart, it is necessary to define the steps, or *scenes* according to the APRIL language, by which the message is constituted. Furthermore it is appropriate to define the users role in each of the steps, i.e. which *interactions* and *behaviours* are present in each *scene*. Thus the following questions:

13) Which steps is the message told in (e.g. according to a timeline, storyboard, user input)?
14) Are there specific user tasks involved in these steps?

Per definition, the physical surroundings (the context) play an important role in AR. The context and its role in relation to the application concept should therefore be considered, i.e. answering:

15) What is the physical surroundings (the location) role in relation to the message?
16) How does it relate to the virtual content and the user?

It is assumable that there is great variety in possible ways a conveying a specific advertising message. This thesis has set out to examine how handheld AR, as technology, can be used as catalyst. To be able to evaluate how handheld AR can be used for conveying the intended message, it is crucial to examine the message in relation to AR as an experience. Towards doing so, the following question was prepared:

17) What is AR's role (as technology and experience) in relation to the message?

Having defined and examined the advertisement message for the application, a series of answers related to the user experience should be addressed. As the intended advertising message is delivering through a user experience, it is appropriate to examine which user experience is intended for the user. Evaluating the AR application, in relation to the problem formulation, will necessarily mean to test it on the end users. Therefore it is important to clarify the correspondence between the intended message and the intended user experience, which relates to the users' perception while being engaged with the application. This leads to the following series of question:

18) Which impression or associations is the advertising message supposed to leave?
19) Is there specific emotions or feelings related to this?
20) Are there other things that can be said about the intended user experience?

To be able to test the application, on the intended end user, it is also necessary to define the target group for the application, i.e. determine their age group, gender etc. Besides it is relevant to determine whether it is a single or multi-user application, i.e. whether the application for instance should afford collaboration or social interaction. The last questions to be answered during the meeting with the ADs can thereby be described as:

21) Who is the Target group for the message?
22) Should the application support multi-user engagement?

2.3.4. Summarising the meeting

The meeting was conducted on 30th of Marts 2011 within relaxed and informal settings of a quiet café. An audio recording of the entire interview can be found at appended CD-ROM under the title: AD meeting.mp3. The following sections will summarize the important aspects and findings from the interview.

Regarding question 1 and 2, the replies stated that none of agency's previous AR applications had been evaluated by the agency itself. However there had been a significant amount of media attention and positive media critique, which had been used for measuring the advertisement campaigns as being successful. As an example, it was explained that one of the AR apps, an AR calendar for the German car company Audi, had received design awards and encouraging reviews. So in relation to the evaluation of the thesis application, there was not concrete methods to adapt, as a the timeframe for the thesis would not allow for obtaining and measuring external media response. In this way there is not results from previous user evaluation of the company's AR application to analyse or obtain inspiration from.

In discussing possible clients to address the application to (question 3), the ADs argued that a coherence between the client's product and AR as technology would be preferable. Using AR to advertise for cereal products would not have as strong impact as when using AR in a context where the technology and experience of AR can be associated to the product itself. Therefore cars in general would be a good choice, and when asked to elaborate, it was explained that cars nowadays are getting more and more equal, which results in the need for the manufactures to compete over technical differences. This applies both to technical differences of the cars themselves but also in relation to how they are being advertised and communicated. It was told that the agency's client, Audi, where using the German phrase *Vorsprung durch Technik* for advertising their brand. This phrase can be translated as *Advance through Technology*, which represents this exact advertisement strategy. Therefore Audi was argued to fit perfectly for the thesis application, as communicating the brand through novel AR technology would prove the actuality of the message, claimed by the phrase. Furthermore this client has tradition for combining classic and digital, which might be symbolized by combining real and virtual.

In answering question 4, it was told that it would make sense to advertise for this client within the public domain because of the fact that advertisement is about getting attention. AR could have the potential of attracting people's attention, surprising as well as stunning them. Opposed to other media forms, like television or radio, an AR application in public space is not limited to specific time periods and will be available to the users at any time. In this way it would meets today's expectations of being able to access information at anytime and anywhere, which has been formed by for instance mobile Internet access. In narrowing down the application context to a more specific location, it was told that the typical big posters format, as seen on billboards all over the Berlin cityscape, would be a good choice due to their visibility and quality of attracting attention.

Through a brief brainstorming process related to question 6, 7 and 8, it was settled on an advertisement concept of creating an AR car release for a new car model. Often car premiers are local events, which only take part on a single location such as a car fair or motor show. With an AR application it could be possible to expand such a premier to multiple locations, e.g. various capital cities simultaneous like a global multi-premier. Developing a non-geographical bound premier, through AR, would be something previously unseen, which could reveal the power of new technology.

In specifying which information would be needed for such a release (question 9) it was explained that there generally are two kinds of car releases, which either are a release of an upgraded but already existing model or a release of a brand new car model. If the release concerns an upgrade, the *communication fire* leading up to the release is often limited to just a couple of weeks, as the companies generally prefers to sell as many of the old models as possible and make a rapid switch. But when car manufactures releases a brand new car for the market, there is commonly three years of *communication fire* before the actual release, for the purpose of building up expectations. The communication of information related to releasing a new car model is thereby a long process, which can be compared to composing a long piece of music. This means that most information is conveyed already before the actual release, and the release is mainly about revealing for the first time how the car model actually looks. This important moment is also called *End of secrecy* and this is a very aesthetic and optic moment. As most information is conveyed prior to the release, the content of the AR application should therefore mainly be the car itself and maybe links to where to obtain further information. At a release, people might be talking about the car's motorisation or capabilities but the main thing of interests, is the shape and aesthetic of the car. So the visual presentation of the car itself is the most important aspect, and an accurate and nice-looking virtual representation of the car would be crucial for an augmented release. Hereby the answers for question 10 and 11, about the characteristics of the virtual content, have also been specified.

In relation to how should the user be able to influence the information flow (question 12), it was mentioned that it could be interesting if the user had possibility to customize how the car was displayed, e.g. make it possible to choose which colour the car is displayed in. This could afford interaction among the spectators who could share their customizations and express their personal taste, while pointing their smartphone on the billboard, thus also answers question 22. Furthermore it was explained that silver generally emphasizes the details of the shapes and underlines the curves because the way it for instance reflects the light. This colour is therefore often chosen for releases, and would make an appropriate initial colour prior to customizations.

How the context, for the virtual content (the car) to augment, should appear (question 15), was argued to be very dependant on which type of car model is to be released. Advertising an urban car would normally be displayed within an urban city environment, a convertible would be shown by the sea, whereas a sports utility vehicle (SUV) both could be depicted in an urban or off-road staging. But for a release the important thing is to guide the focus towards the car, and the background should therefore be discrete and designed for not attracting attention away from the car. However car manufactures tries to make their releases different from competitors, e.g. by having a release situated at an opera stage or inside an illuminated motor tunnel, so it should be noted that the context of the release deliberately can be chosen for creating associations to the car model. In replying to a additional question about whether the actual location of the billboard should play a role in relation to the poster design and AR application in general, it was argued that such coherence could enhance the overall impact of the campaign, e.g. by letting the visual content of the poster correspond to the actual environment around the billboard. An idea for doing so was, according to one of the ADs, to let the poster depict the inside of the building at which the billboard is mounted onto.

Besides from being exclusive, the general wished-for impressions and associations (question 18) are that Audi are ahead of their competitors, both conceptually and product wise, due to advance through technology, i.e. are the first in all aspects, are in constant progress, are the state-of-the-art. Furthermore the intention is to actually show that this is the case instead of merely just claiming it. Buying a car is a very personal thing, and it is the actual design of the car together with economical factors, which will have the greatest impact on the choice when selecting a new car. However brand associations do influence choices and, as mentioned earlier, the technology of AR could possibly be used, as medium for expressing that Audi is innovative and ahead on all fronts. The target group for Audi's cars where described as affluent people with professional careers and general curiosity about technology, which interestingly also were including younger people.

2.4. Outlining the application

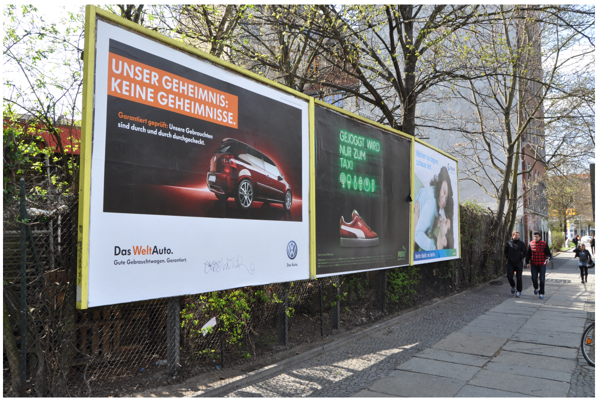
Based on the meeting outcome, the application context and content can be outlined as follows:

2.4.1. Client

The car manufacturer, Audi, has been chosen as addresser for the AR application because of the fact that AR technology, as communication tool, both suits and manifests the very nature of their advertising strategy "advance through technology". Furthermore it is attractive for this client to advertise within the public domain where the attention of a broad target group could be attracted, as well as using a mobile media form, which is available on demand at any time. In this way the quality of mobility within advertisement also matches the previous argued qualities within mobility of handheld AR.

2.4.2. Context

The reality, or situation, which is to be augmented with the AR application, was chosen as being the traditional channels for public advertisement, which are billboards such as the ones depicted below.



Picture 23 - 26: Advertisement billboards within the Berlin cityscape

2.4.3. Concept

The overall application concept idea is to create a novel and alternative form of a release for a brand new car model, i.e. break *the end of secrecy*, though AR. Hereby the application is an event-based campaign, which conceptually could be staged in multiple cities simultaneous. More specifically the idea is for the user to point the smartphone towards the billboard in order to display the virtual 3D car on top of the poster environment. By viewing the billboards from different perspectives the user can thereby reveal the shapes and curves of the car, i.e. explore the AR application interactively.

2.4.4. Poster content

The poster is serving as stage for the augmentation. A criterion for the poster design is that it should be visual neutral in order not to distract attention from the virtual car, which in turn should be in focus. Furthermore the poster design should resemble a geographical landscape, which matches the intended nature and spirit of the car, e.g. a landscape scenery or city environment. Ideally, a visual connection between the poster and the surroundings of the billboards could provide an enhanced visual coherence. As an example, each poster could be specifically designed for each billboard in each city.

2.4.5. Virtual content

The virtual content of the application should be limited to a 3D model of the actual car being released. It is important that the model appears aesthetically appealing, and resembles the real car as the curves and shapes constituting the car form should be in focus for the user. Additionally, a grey colour for the car assumable will accentuate its forms but a possibility for the users to customize colour could enable adjustments to the users personal taste and might furthermore enable social interaction among them.

Further virtual information should be omitted as the car model should reserve full attention, and such information moreover typically would have been delivered during a longer *communication fire* prior to the release. Eventually, web-links to where to find further information about the car could be provided.

3. Technical analysis

The following chapter will be a technical analysis of how the application concept can be utilized, with respect to the problem formulation.

3.1. Analysing the application context

3.1.1. Unpredictable conditions

One of the most significant characteristics for the application context, being an outdoor environment, is that the lighting conditions are continuously changing with factors such as weather conditions, time of day, time of year etc., due to such factors' impact on the illumination from the sun. In comparison with a controlled environment, where the lighting conditions can be controlled, this raises the challenge of utilizing a tracking system, which is somewhat are resistible to such changing lighting conditions. Ultimately the tracking implementation should work under, or be able to adapt to, all possible meteorological conditions, including rain etc., but it is however believed that this not is a criterion for being able to evaluate the application.

Other characteristics, present in an outdoor city environment, are related to external distractive elements, which could influence how the AR application will be perceived. Due to the dynamic nature of cityscapes, unpredictable external visual and auditory stimuli are bound to occur, thus possibly divide the attention by stealing focus from the application. Bypassing people in the space between user and poster could occlude the visual application field, as well as causing issues for camera tracking, and a sudden sound from for instance a car horn might bring reflective responses in the user. General auditory noise, from traffic etc., could furthermore be a more overall and permanent distraction but as the application does not contain sound element, the influence from such distraction is less crucial.

To consider the influences from the changes of conditions within the environment, it is found appropriate to pilot test the application in an outdoor environment during the iterative design and implementation phase, and more importantly evaluate it under the actual circumstances, i.e. test it in the street.

3.1.2. Conflict between visual cues

The application concept dictated that a poster should be augmented rather than an actual real-world 3D space. As the poster should depict a real environment, i.e. should consist of a photograph of a real environment that matches the nature of the car model, the poster can be considered a 2D representation of a 3D space. In this way the "reality" that will be augmented is in fact an abstract version of a "real" reality where only two-dimensional visual cues are given. For a moving user this means that dept cues, which in the real world are given by distances between user and objects in the scene while viewing from a continuously changing perspective, not are provided by the 2D representation. The only dept cue available, with the poster, are only related to the users spatial cognition of the depicted space, as well as cognitive logic relationship between the size of the (3D) car and the space it is situated in. If assuming the car is correctly registered according to the user view, the car itself will however provide 3D depth cues, which correspond to visual information about object geometry as perceived in the real world. In other words, there could be mismatch in depth cues between the 2D poster and the 3D car, which might influence how the AR application will be experienced. When evaluating the experience, provided by how AR technology have been applied to the given application concept, such a possible issue should therefore be taken into account.

3.2. Analysing the platform

Throughout the thesis, it has been known that the handheld AR application should be utilized on a smartphone. Before the implementation can be initialized, it is however necessary to further delimit the target platform, i.e. choose whether to develop the application for iPhone or Android. Based on prior experience with iOS development, it has been chosen to make the AR application into an iPhone App. The results, found by evaluating the application, are however believed to be platform independent and unaffected from such choice.

3.2.1. Hardware limitations

Developing an iPhone application is obviously different than developing a PC application, as such a device offers less computational capabilities etc. Even though the iPhone hardware has improved significantly recently, it should still be considered that only limited CPU and RAM are available for running the applications. Instead of building a theoretical foundation for understanding the exact requirements for iOS development, it has been chosen to perform the experimental implementation by compiling the source code directly on an actual device rather than the iPhone simulator, which is part of the Apple iOS developers' tools. Hereby the danger that the source code would run only in the simulator but not on the device is eliminated, and moreover such an approach should give a consistent impression about loading times and general performance. As the application is intended for research purposes only, also the considerations on fulfilling the source code requirements for submitting applications to Apple's AppStore are intentionally left out.

3.2.2. Hands-up display

In contrast to stereoscopic displays, a benefit with smartphones' monoscopic display is that a series of perceptual depth cues issues, caused by miscalibration between the two displays, are avoided altogether. As agued earlier, smartphones' hands-up displays are furthermore minimally intrusive, socially acceptable, readily available, and highly mobile. A drawback with hands-up displays can however be that only one hand is left for interaction. But for a great variety of AR applications in which the user is intended to interact with the virtual content, smartphones' multi-touch features, present at the same plane (the display surface) where the virtual content is depicted, to some extent counts for this.

For AR applications that provide an extensive amount of information, e.g. by descriptive text information about the surroundings, the limited screen space of smartphones should logically be considered during the design phase, i.e. reducing the information load to a minimum. Usability tests and design standards could aid in approximating a right balance for this, but in case of the thesis application, where the virtual content is limited to a car model, such issues are avoided. In an imaginable future, where the physical space we live in could be overloaded by augmentations, view management however becomes a broader issue, which would call for a debate about related ethics and on how to enable the users to filter the information flow.

In returning to the limited field of view provided by the display size of smartphones, it is however noticeable that the environment of target for the augmentations not necessarily must be restricted to a size, which can be contained within the display view. Moving the display to navigate through an information space that is essentially larger than the field of view, for instance supports a visual perception phenomenon that is known as the *Parks effect*, which allow the user to experience a larger AR space due to the persistence of the image on the user's retina [Bimber & Raskar 2005].

3.3. Tracking and registration methodology

As discovered, in the examination of requirements for AR systems, a criterion for AR was the need for the virtual and real world object to be aligned in respect to the user's current view. This is known as the process of registering the virtual content, and this can, according to [Bimber & Raskar 2005 p.109], be explained as achieving a geometric relationship between real and virtual coordinates. A premise for being able to do so, while a user is moving, is the necessity of knowing the user's current view at all times and in real-time, which is referred to as tracking. In this sense, an AR system is a context aware system, where tracking is a prerequisite for enabling registration, which in turn is a prerequisite for sustaining the illusion that the VOs coexists in the same world as the real-world objects. According to [Reitmayr et al. 2010], tracking and registration are the two most fundamental technical challenges in developing AR systems. To be able to successfully overcome these two challenges, they deserve a deeper analysis. The following sections will be used for carrying out such an analysis, which should build a base for choosing the appropriate tracking and registration methods for the thesis' application.

3.3.1. Tracking

„The precise, fast, and robust tracking of the observer, as well as the real and virtual objects within the environment, is critical for convincing AR applications" [Bimber & Raskar 2005 p.19]

Izkara et al. divides the tracking process into the following two main tasks [Izkara et al. 2007 p. 5]:

1) The initial pose estimation where the system must recognise the scene and compute the camera pose for that frame.
2) Once the initial pose has been computed, the system must update the camera pose according to the

movements of the real camera. This is the tracking phase.

To be able to elaborate on these two tasks, the first step is to understand what is meant by the term *pose*. In short, pose refers to the user's position and orientation with respect to the user' view [Ekengren 2004]. To give the user freedom to move around within the AR environment, the AR system must continuously allow for a pose with 6 degrees of freedom (6DOF), which is constituted by three angles for orientation (the Euler angles of yaw, pitch and roll) and three variables for position (x, y and z coordinates) [Van Krevelen & Poelman 2010]. In the context, of a handheld video-see-through AR system for smartphones, tracking can be reduced to considering the pose of the actual device, as the device display serves as the view to the world. Allowing for 6DOF can in this way can be seen as giving the user complete freedom for moving the device around in 3D space.

In returning to Izkara et al.'s division of the tracking process, the first task can therefore be explained as determining the initial pose of the user at the point of entering the application space. The subsequent task is thereafter to continuously update the system to be aware of the change in the user's pose, i.e. keep track of how the build-in camera's direction in pointing is changing, as well as its absolute location within the environment, over time.

Having gained a theoretical understanding of what tracking consists of, the remaining question is: How to actually create a system that can count for pose estimation? [Van Krevelen & Poelman 2010] argues that tracking is as a complex problem with no single best solution. Such solution would be dependent on application context and character, and these factors should therefore be taking into account when choosing tracking method. In 2007, [Haller et al. 2007 p.384] postulated that there, at the time, still not where technology to perform accurate real-time tracking in outdoor environments due to the dynamic change of conditions, such as light (e.g. change in intensity or absent of sunlight) or moving objects (e.g. bypassing persons or vehicles). The following sections will be an investigation contemporary tracking methodology.

3.3.2. Tracking methods

In general, tracking methods can be differentiated as being either egocentric *inside-out* or exocentric *outside-in*. To explain those terms in the context of handheld AR systems, an egocentric inside-out tracking system is relying solely on the equipment of the Smartphone, as opposed to exocentric outside-in tracking, where an instrumented environment can establish the device pose and return the tracking data for registration, e.g. attaching sensors into the environment [Henrysson 2007]. In this way, an egocentric inside-out tracking system can thereby be regarded a self-contained system. Even though this tracking classification often is employed for describing camera-based approaches, they are also well suited for describing other tracking technologies as well [Bimber & Raskar 2005, p.19]. Furthermore, *tracking-by-detection* can be used as an overall term for describing AR tracking in general.

Besides for the above-mentioned differentiation, tracking can be divided into the categories of either being sensor-based, vision-based, or a combination of the two (hybrid). Having cutting edge within AR technology for smartphones, the following section will provide an overview of various tracking methods according to the categories they belong to. The aim is to obtain a broad theoretical understanding about strengths and weaknesses in relation to the application context, without going into too many unnecessary technical details about the underlying mathematical algorithms etc. This should lead to a substantiated choice of tracking method for the thesis' application when taking its context into consideration.

3.3.3. Sensor-based tracking techniques

As the name suggests, sensor-based tracking techniques, also referred to as non-visual tracking techniques, are methods that depends on using sensors for determining the pose.

Such sensors can for instance be ultrasonic sensors, which measures distance according to how long it takes for acoustic signals to propagate through space. By sending out auditory signals in ultrasonic frequencies, above human hearing range, from emitters, the time of flight (TOF) can be estimated according to the time duration it takes for the receivers to sense the signal [Ekengren 2009 p.25]. An example of an ultrasonic sensor is the PING))) sensor, which can send out a signals and sense when it returns in for of an echo [web 38]. A drawback with ultrasonic sensors is the limitation in possible distance range due to the loss of energy, which sound have with travelled distance [Ekengren 2009]. The PING))) sensor, for instance, only has a range around three meters, which could be insufficient for AR systems that target large public spaces.

Electromagnetic sensors use electromagnetic signals, instead of sound, for determining the TOF. A system that is build upon this approach is the global positioning system (GPS), which uses 24 satellites and 12 ground stations spread around the world [Ekengren 2009]. With a precision of around 3-10 meters, GPS is too inaccuracy for most AR application but the differential GPS (DGPS) system comes closer to usable measurements with its precision around a meter. Besides for the poor accuracy, another downside is related to fact that occlusion within lies of sight to the satellites can course failure in measurements. In another context than estimating the pose, it is noticeable that GPS can be used for AR application to determine the position of physical objects within the surroundings, e.g. as it is the case with the previously described AR browsers of Junaio, Layer and Wikitude. In general, a handheld AR system that partly relies on information from a server can be termed a *distributed system* as contrast to a *standalone* system [João 2009].

As opposed to TOF, the principle of inertial sensing is based on sensing actual change in position or orientation. In contrast to TOF measurements, which typically implies to exocentric *outside-in* AR system, inertial sensors implies a use of an egocentric *inside-out* tracking approach. In case of a video-see-through AR system for smartphones, the sensing is thereby typically done through inertial sensors from within the device itself. Android and iPhone smartphones both have a build-in accelerometer, which is an inertial sensor that can measure linear or angular acceleration of the object attached to. A problematic part of estimating position with accelerometers is that they only measures change in force applied to the object. While it is not an issue as long as the smartphones position is unchanged, the problem appears when a constant force (steady motion) is applied without being sensed.


Picture 27: Orientation axis of a Gyroscope (left). Picture 28: Corresponding Euler angles (right).

Today's smartphones are also equipped with compasses, which are sensors that are using the two magnetic poles of the earth to determine horizontal orientation. Another very interesting thing, related to inertial orientation sensing with smartphones, is that the latest G4 iPhone also contains a build-in mechanical gyroscope. In contrast to compasses, sensing with gyroscopes allows for 3DOF (pitch, yaw and roll) and thereby estimate the total orientation. A general issue with gyroscopes can be that they are prone to drift but a periodical recalibration can to a certain extend count for this [Ekengren 2009]. The first iPhone games with gyroscope control has already seen their light, such as first person shooter game, N.O.V.A – Near Orbit Vanguard Alliance, which renders the virtual game world according to the players orientation, as depicted below.

Picture 29: N.O.V.A

According to [Ekengren p.20] there are no standalone sensors that afford general reliable 6DOF tracking in unprepared outdoor environments, and AR system utilizing sensors are therefore typically doing so in combination with other tracking techniques, such as vision-based approaches, in order to estimate the pose.

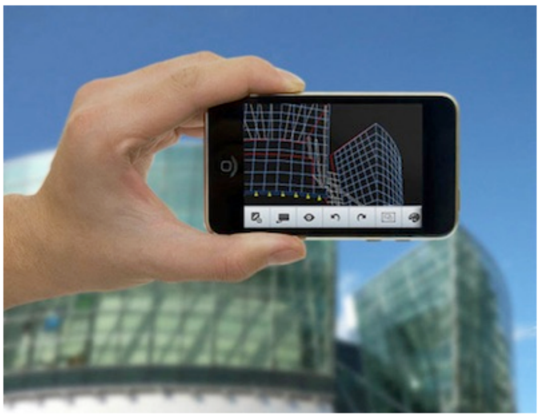
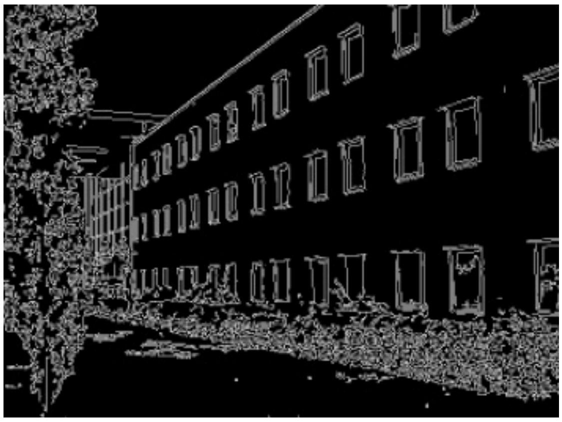
3.3.4. Vision-based tracking techniques

Vision-based tracking techniques, are techniques that uses image processing methods for calculating the camera pose, thus applies to video-see-through AR systems. According to Zhou et al., vision-based techniques have been the most active research area within tracking, covering more than 80% of the presented approaches on ISMAR over the years [Zhou et al. 2008]. Most of available vision based techniques can be divided into three classes: *Marker-based, Model-based* and *Structure from Motion* (SfM), where both Model-based and SfM are markerless approaches. As markerless tracking have been define as a criterion for the thesis implementation, the focus will, besides for a brief introduction to Marker-based tracking, be on obtaining an overview and understanding of the various vision-based markerless methods. Analyzing such methods, while having smartphones as computational platform in mind, should provide basis for determining which tracking methods would be the most suiting for the thesis' application.

Tracking with markers is a simple, easy and robust, method, due to the simplicity in the binary nature of these black and white markers. The technical steps in this approach generally are to binarize the captured images, process them to connect the potential connected components, and match the QR pattern against the image to find its position within the image. This position can subsequently be used for calculating the camera pose in respect to the marker's position [Yu 2009]. As marker-based tracking is beyond scope of this thesis, further details will be omitted.

The most recent trend in computer vision tracking techniques, presented at ISMAR, is the *Model-based,* which, opposed to marker-based tracking, relies the previously described NFT [Zhou et al. 2008]. *Model-based* methods require priori knowledge about the physical environment, such as a 3D model or an image template that will appear in the scene, and these methods can be further subdivided into the following three categories: *Edge-based*, *texture-based* and *Optical flow-based*. These categories will be further explained below.

With Edge-based methods, computer vision algorithms are used for identifying and extracting edges from the captured images, and subsequently matching them against the edges of a 2D or 3D model of the scene [Yu 2009]. Such a model can for instance be a Computer-Aided Design (CAD) model, which is a representation of a physical world objects that can have similar characteristics as the blueprints of a building [web 39].

Picture 30: A smartphone depicting a CAD model of the background building (left). Picture 31: Edges from building extracted through canny edge detection algorithms in OpenCV (right).

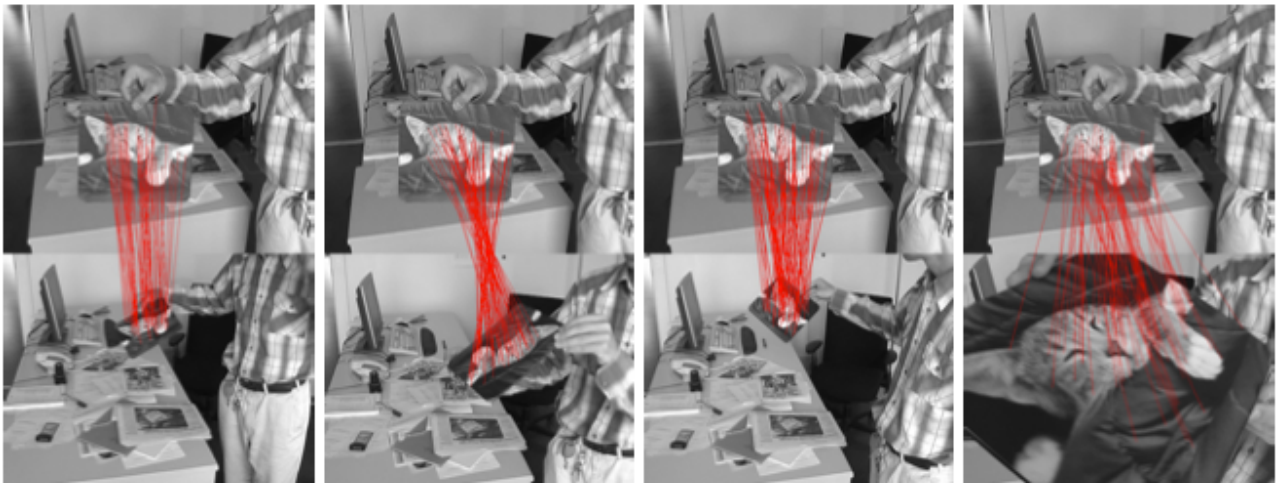
It is noticeable that an *edge-based* method is well suited for augmenting spaces, which contains a variation of distinct geometrical features, e.g. lines and corners, as apparent in for instance wide area outdoor public spaces with numerous buildings. In relation to outdoor environments, where lighting conditions is bound to be unpredictable, another benefit, with this method, is that the extracted edges generally remains the same even though the lighting situation changes [Zhou et al. 2008].

As the name suggest, *texture-based* methods takes texture information, from the captured images, into account. In this context, texture can be understood according to [Yilmaz et al. 2006]'s definition of a texture as being a measure of the intensity variation of a surface which quantifies properties such as smoothness or regularity.

One branch of *texture-based* methods uses global texture information. By matching known templates, in form of a selected region of interest (ROI) from for instance a picture of the surroundings, against the corresponding ROI of the captured images. The Open Source Computer Vision (OpenCV) programming library provides various functions for rapidly implementing template matching but a downside with template matching is that it is computationally heavy [web 40]. Furthermore there logically is a trade-off between ROI size and required computational effort, as well as a trade-off between how small the ROI is and how big the chances for it to be occluded, within the environment, are. But the most significant downsides with template matching is probably that is does not allow for skew camera angles as well as the fact that it is sensitive to lighting conditions.

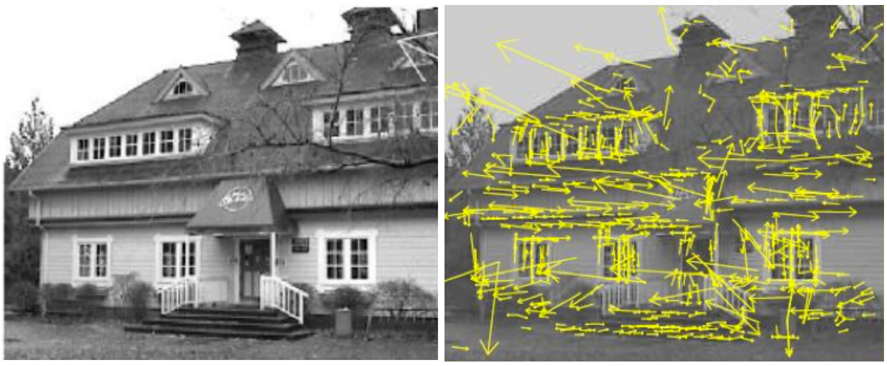
The other branch of *texture-based* methods is based on using local texture information for extracting feature points, which are also referred to as key points or interest point [Zhou et al. 2008]. As with template matching, a common denominator for extracting interest point is a need of having prior knowledge about the scene. Instead of matching whole segment of images against each other, it is sufficient to match interest points from the captured images against the same interest points of a reference image. This results in significant decrease in necessary data to compare, and these methods are consequently far less computational demanding than template matching methods. There are a variety of computer techniques for determining which feature characteristic the selection of points should be based upon, and this phenomenon is known as the method's *descriptor*. Furthermore the terminology for the techniques related to the actual matching can be referred to as the method's *detector*. Without going too much into technical details, the following sections will describe the most important and relevant of this branch of *texture-based* methods.

An example of a method using interest points is the FERN detector, which was made by CVLab in 2007 [web 41]. Significant for the FERN method is that it requires an offline training phase where the texture of a reference image template is analyzed and converted into non-hierarchical structures, called *ferns.* Hereby the most identifiable interest points are recognized and these can subsequently be matched against the incoming camera images by pair-wise pixel based intensity comparisons [Özuysal et al. 2010]. In this way, FERN is a two-step approach with a slow initialization but a subsequent fast run-time performance. According to [Wagner 2009], FERN is the state-of-the-art for fast pose tracking, and the FERN source code, intended for being compiled on PCs, is available under GNU General Public License [web 42]. The memory required for storing the *ferns* are apparently extensive but [Wagner et al. 2010] has successfully modified FERN to run on smartphones in 2010, which they refer to as PhonyFern. Below four image sequences are visualising the interest point that has been detected with FERN. Each picture is divided into an upper image, which shows the static reference image, used for generating *ferns*, and a lower image that is a sample from the camera sequence. The corresponding interest points are furthermore connected pair-wise with a red line for illustrative purposes.



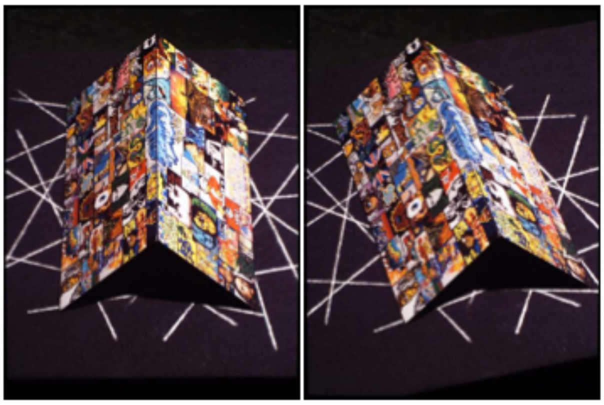
Picture 32-35: FERN

Scale Invariant Feature Transformation (SIFT) is a computer vision algorithm for describing and detecting interest points in an image, which was made, by David Lowe, back in 1999 [Meng 2004]. The algorithm transforms the image into a collection of local features in the form of vectors, as depicted on picture 37 below, and each of these vectors are supposed to be distinctive and invariant to image scaling, rotation or translation, which means that it allow for skew camera angles [Meng 2004]. Calculating vector data for each point makes the performance of SIFT robust but also slow. In comparison with FERN, SIFT does not require a longer training phase but are on the other hand performing slower during run-time [Wagner 2010]. By exploiting the new capabilities of smartphones, and modifying the algorithm, a research group, from Christian Doppler's Laboratory for Handheld Augmented Reality, have successfully transferred SIFT into PhonySIFT for smartphones in 2008 [web 43].



Picture 36 & 37: SIFT

Inspired by SIFT, Bay et al. made a similar but simplified algorithm in 2008, called Speeded Up Robust Features (SURF), which they claimed to be both faster and more robust than the original SIFT [Bay et al. 2008]. The descriptor of the SURF method is build for finding distinctive locations in the image, such as corners, blobs, and T-junctions [Bay et al. 2008], which makes SURF attractive for tracking simple geometric shapes such as the ones seen on picture 38 below. In the same way as with SIFT, a vector is also calculated for each interest point, and the detector is based on estimating a match by comparing the distance between corresponding vectors in the captured image and reference image respectively.



Picture 38: SURF

Optical flow-based methods have the significant characteristic that they do not require any previously defined information about the environment, e.g. an image reference template. Solely by using computer vision algorithm, change in camera motion can be calculated by comparing the actual camera frame to the previous frame. This means that more than one image is necessary, and where the current image is denoted *N*, the calculation of Optical Flow are based on *N* with respect to *N*-1 in the sequence of captured camera images. The Optic Flow methods in general use variety of the images brightness, and computes velocity vectors from which the motion can be estimated. The following pictures illustrate two sequential images as well as the corresponding velocity vectors displayed on top of an image, which is equal to the second of the sequence:

Picture 39, 40, 41: Previous image N-1, actual image N, and the Optical Flow between the two.

By mapping these velocity vectors to a 2D array, corresponding to the size the images pixel array, the Optical Flow for each frame in the image sequence can be represented in a vector field, where each vector visualize the direction and amount of motion in respect to position in the image. Below, an example of such a vector field is depicted but it should be noted that the example does not relate to the image sequence from above.

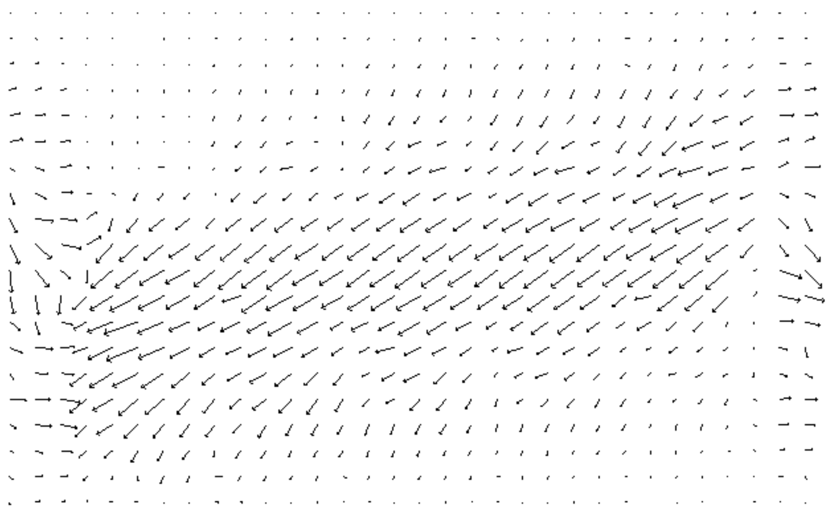


Figure 4: Vector field representing Optical Flow

Optical Flow has its root in research on robotics and was originally developed for the purpose of estimating the egomotion of moving robots, i.e. for path planning and navigation in unknown terrains [Roberts 2009]. The process of determining the position and orientation of a robot by analysing the sequence of images, captured by the robot, is known as Visual odometry, which, besides from Optical flow, also includes methods for getting distances by using for instance capturing images of marked robot wheels. Obtaining a structure of the scene or environment through Optical Flow is furthermore referred to as Structure from Motion (SfM) [Pupilli & Calway 2002]. Simultaneous Localization and Mapping (SLAM) is an example of such a method, which uses SfM to build a map of unknown environments meanwhile using the map to navigate a robot through it [Riisgaard 2004].

In relation to registration of virtual content, it is important to notice that Optical Flow does not provide initial pose estimation with respect to the environment, i.e. does not count for Izkara et al.'s previously mentioned first step of the tracking process. An additional tracking method for obtaining the initial pose is therefore needed to supplement Optical Flow in order to provide a complete tracking solution for an AR system.

The Parallel Tracking and Mapping (PTAM) is a nice example of a SLAM method, which both supplies means for estimating initial pose as well as uses Optical Flow for calculating the subsequent change in pose. For estimating the initial pose, PTAM requires the user to initialize the system while pointing the camera towards a planar surface [Klein & Murray 2009]. The surface will then be identified by the system, and form ground-plane for future augmentations. As a consequence there is certain requirements for the system: Either the user needs to be instructed about the initial step, or the application design or context must secured that a planar plane will be present during initialization phase of the system.

PTAM was developed by Klein and Murray at the Active Vision Laboratory, University of Oxford, in 2007, and was modified, to suit the capabilities of handheld mobile devices, in 2008 [web 44]. A PC version of the PTAM Source Code is available for non-commercial and research usage [web 45] but this C++ code would need to be heavily modified in order for it to compile on a smartphone device.

3.3.1. Errors in tracking

In general errors in tracking can be divided into two categories: static or dynamic errors [Ekengren 2008]. Static errors are the ones, which causes registration errors when the user and the environment are still, whereas dynamic errors are the ones that can appear when the user and environment is in movement. In this way the static errors are related to fault in the tracking system or measurement of the associated sensors etc., opposed to the dynamic errors, which occurs when image processing, graphics renderings or other computations causes system delays, thus creating a time-gap from when the user's real-time pose is performed to the point where the virtual content are actually registered and rendered according to it.

As learned from the preliminary analysis, the tracking of the users pose was a prerequisite for the subsequent process, called registration, and following sections will be a deeper analysis of this process.

3.3.2. Registration in theory

„The objects of the virtual and the real world must be perfectly aligned at all times or the illusion of coexistence will fail" [Ekengren 2004 p.24]

Registration is the accurate alignment of virtual and real images in respect to the user's view [Haller et al. 2007]. As previously described, a prerequisite for such alignment is to track the pose of the user and render the graphics accordingly. As a consequence, any inaccuracies or errors in tracking will cause the registration of the virtual content to be off from its intended position in relation the real world.

This problem also exists in VR but is less serious and critical here. Due to the total immersion within a complete virtual world, registration errors result in visual-kinesthetic and visual-proprioceptive conflicts, which means that the visual modality is in conflict with how positions and movements of body parts are perceived [Azuma 1997]. Such modality conflict can result in the feeling of motion sickness but as the kinesthetic and proprioceptive systems are much less sensitive than the visual system, the conflict is less noticeable than in AR where the conflict, caused by registration errors, is visual-visual. Furthermore, a phenomenon known as *visual capture* makes it even more difficult to detect registration errors in VR. This phenomenon is caused from the dominance of the human visual modality, which results in a tendency of the brain to believe what it sees rather than what it hears, feels, etc. A good example hereof is the *ventriloquist effect* where for instance the speech of an actor in a television show is perceived as actually coming from the actor's mouth rather than the actual speaker of the television [Choe et al. 1975]. The effect of visual capture increases the amount of registration error, which users can tolerate in Virtual Environment systems, and if such errors are systematic, users even might be able to adapt to the new environment if given a longer exposure time [Azuma 1997].

Generally the threshold, which determines the amount of acceptable registration errors, is dependent on context and purpose of the AR application, i.e. while some applications requires very accurate registration, less accurate registration might be sufficient for others. For AR applied to the medical field in order to aid in surgery for instance, a precise registration is of crucial importance while an AR application providing text information on surrounding restaurants demands less accuracy. In considering the thesis application, it is can be noticed that the application does not serve the purpose of guiding in task where high precision is required or in any task solving in general. Therefore there is not a crucial need for a perfect registration. However the covered research, e.g. [Ekengren 2004 p.19], point to a difficulty in convincing people that computer-generated VOs actually live in the same physical space as the real world objects, and shows that an accurate registration is a prerequisite for a sustained illusion of co-existence between the virtual and the real. A proper registration is thereby all-important nevertheless. To provide a more general understanding of the relationship between registration accuracies and AR application, it can furthermore be added that also distance plays a role. According to [Ekengren 2004 p.20], such relationship can be explained as follows:

"When registration errors are measured as the screen distance between the projected physical target point and the point where the annotation gets drawn, the following observation holds: The further away the object that is to be annotated, the less errors in position tracking impact registration accuracy" [Ekengren 2004 p.20]

From such statement it is apparent that failure in location tracking of the user causes higher registration errors the smaller the distance. However the opposite is the case with failure in orientation tracking, which will cause higher registration errors as the distance increases [Ekengren 2004 p.20]. As most targets for augmentation in outdoor environments tends to be some distance away from the user, errors in orientation tracking can be argued to have most impact on mis-registration for such cases.

3.3.3. Registration in practice

The following sections will outline the concept of aligning the virtual 3D object in respect to the real-world geography and the tracked pose of the user. As the 3D object is rendered on the actual display of the smartphone, from which the pose also has been tracked from, the process can be understood as a projection of the 3D object onto the 2D image plane, which resembles the display. In order to register the virtual content at a fixed position within the real-world environment, this content is defined within a general world coordinate system (wx, wy, wz) to obtain a description, which does not depend on the camera position [Lima et al. 2009]. The actual registration can thereby be done by rotating and translating the model into another camera-dependent coordinate system (cx, cy, cz) according to which the model is rendered. Hereby there is always a correspondence between the world and camera-dependent coordinate systems, which is defined as an affine transformation that are constituted by applying a rotation and a translation such as depicted on figure 5.

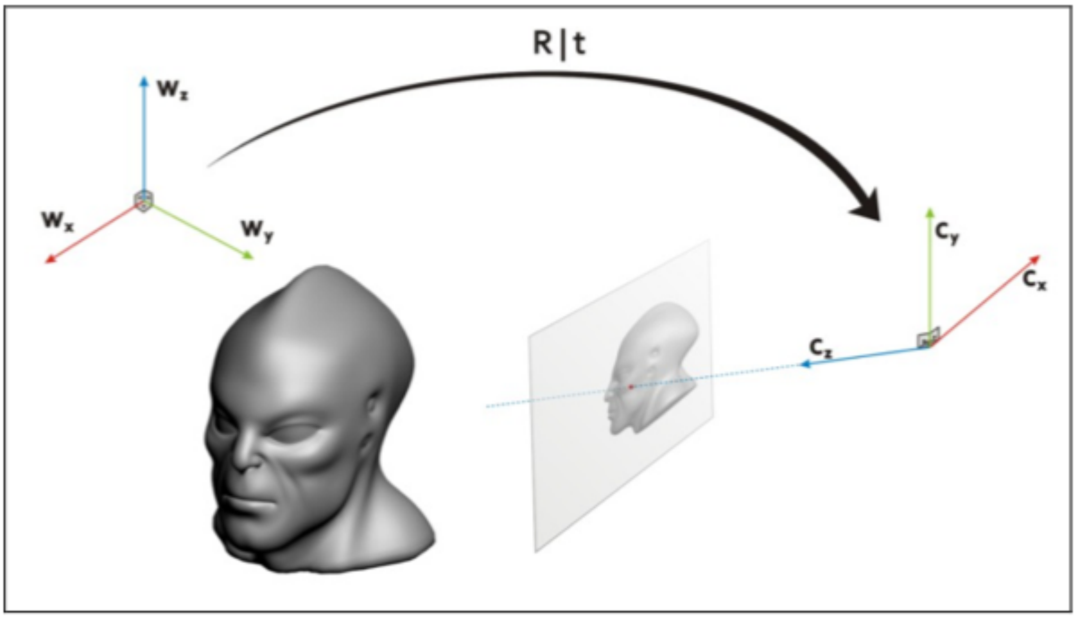


Figure 5: Virtual 3D model (the head), its projection onto the image plane (the display), and the relationship between world (wx, wy, wz) and the camera (cx, cy, cz) coordinate systems

According to Lima et al. there are various methods for calculating for such a projection, but the most simple and common approach is however to use a so-called *pinhole* camera model, where a camera is modelled by a set of intrinsic and extrinsic parameters. Here the intrinsic parameters are the ones that define the optical properties of the actual camera, such as the focal length, aspect ratio, and the location of the image centre, whereas the extrinsic parameters define the tracked pose with respect to some the world-coordinate system [Lima et al. 2009]. More in-depth descriptions of the mathematics concerning the registration of the virtual content, related to the thesis application, will be further described at the point where it is needed.

3.4. Virtual content

3.4.1. OpenGL ES

OpenGL (Open Graphics Library) is a cross-platform and cross-language API for developing applications, which uses 2D and 3D graphics. After the revolution of mobile phones took off, the Khronos Group made a trimmed-down version of OpenGL for embedded systems, called OpenGL ES. This version was released in 2003 and some present platforms that support OpenGL ES includes iPhone, Android, Symbian and PlayStation 3 [Rideout 2010]. An alternative to OpenGL ES could be the similar Direct3D API from Microsoft, but as Direct3D is Windows only, ES seems to be the appropriate choice for implementing the graphical side of the AR application.

There are two versions of OpenGL ES, which is ES 1 and 2 (from 2007) respectively, and they are quit different in their rendering pipeline. Whereas ES 1 builds upon a *fixed-function graphics pipeline,* ES 2 are said to have a programmable graphics pipeline, which makes ES 2 more flexible but also requiring more code writing from the developer. Furthermore ES 2 is only supported on newer smartphones such as iPhone

3GS and G4. A benefit with ES 2 is that it supports shaders, which are pieces of C-like code written in the OpenGL Shading Language (GLSL) and are compiled on the device's GPU at runtime. Hereby complex rendering can be processed on the GPU without affecting the performance of the CPU. As there assumable will be an extensive amount of computation required for the CPU to perform for tracking related image processing, it seem appropriate to exploit this advantage and implement the virtual content with ES 2. Besides, the application scope is to use cutting edge AR technology, from which it follows also to use the state-of-the-art within smartphone processing technology. It is neither the scope for this thesis to dig deeper into the complex structure of OpenGL API nor to cover all potential and possibilities provided by it. The following section will however provide a brief overview of various rendering aspects, which necessarily must be used in order to create an AR application with virtual 3D content with OpenGL ES 2.0.

3.4.2. Shaders

In general there are two types of shaders, vertex and fragment shader respectively, which typically are used in combination in ES 2.0 based iOS applications. Vertex shaders are used to transform a 3D model's vertices that are submitted with glDrawArrays function calls, while fragment shaders compute the colours for every pixel in every triangle of the model. Because of the highly parallel nature of graphics processors, thousands of shader instances execute simultaneously, and a significant characteristic is that shader are compiled at runtime on the iPhone itself [Rideout 2010].

3.4.3. Lighting

In any virtual 3D environment various algorithms apply the illumination of the virtual content during the rendering process, and for ES 2.0 application those algorithms are furthermore passed to the shaders. In this way, such illumination is an attempt to create realistic looking lighting that makes the 3D scene appear naturalistic. In case of an AR application the situation however gets more complicated as the illuminated virtual content is shown together with real-world objects, which are illuminated by real-world lighting sources such as the sun or electrical light sources. To avoid a mismatch between these two illuminations, the virtual illumination should imitate the illumination present within the real environment where the AR experience takes place, i.e. have same direction, intensity etc. Especially within an outdoor environment this becomes impossible due to the constant change in lighting conditions, which is present in such context. Perceptual issues related to this problem are thereby unavoidable, but becomes less noticeable the better the approximation of the virtual lighting is carried out. How to utilize an adaptive AR application that deals with this problem could be a study in itself, and for the thesis application it has been decided merely to create a virtual illumination, which to a certain extent appears naturalistic in it self and correspond to the illumination in the photography that constitutes the background poster.

To create illumination in OpenGL, at least one light source must be initialized by code. Such light can have different illumination characteristics that can be defined by light properties, e.g. directional, spot or ambient. Furthermore the source should be given a location within the OpenGL world coordinate space together with a direction. Additionally various material properties for the model's meshes can be passed to the shaders, and thereby it can be determined how the individual model materials should reflect the light source, e.g. the

GL_SHININESS parameter specifies the specular exponent, GL_SPECULAR specifies the specular colour of the surface, and GL_DIFFUSE specifies the diffuse colour of the surface [Rideout 2010].

3.4.4. 3D model

It is possible to use the iPhone SDK for drawing simple meshes, which can be rendered with OpenGL. But as a criterion for the car model was for it to appear naturalistic, it would be impossible to solely rely on such an approach. Fortunately various modelling software, such as Autodesk Maya and 3D Max [web 46] or the open source Blender [web 47] can be used as tool for creating complex model structures, which can be stored in a variety of model formats. Here material properties can be assign to each mesh in the model, and furthermore the model can be UV mapped, which makes it possible to bind textures to the UV coordinates of model's meshes [Rideout 2010].

4. Iterative design and implementation

This point in the development process would generally be where a UML statecharts over the dynamic flow of control from state to state within the AR system, as proposed by APRIL, could provide an illustrative overview of the system, which could aid throughout the iterative design and implementation phase. As the actual system is simple, in terms of not having an event-driven structure, it has however been chosen to omit using UML statecharts.

In relation to structuring the iterative design and implementation phase, it has been learned that the graphical content and the tracking functionally with advantage can be separated into two distinct processes. Hereby it is easier to adapt the AR application to other platforms, as well as possible to interchange tracking methodology without affecting the graphical implementation and visa versa. Because of the application concept and chosen tracking approach, yet another design process is introduced: the creation of a poster image, which natural features are used as reference for the tracking. The logic order, in which to proceed with these processes, therefore is as follows:

1) Create the virtual content (the 3D car model) and develop OpenGL functionality for it to be rendered in iOS.
2) Create a poster design that depicts an environment, which matches the chosen type of car's nature.
3) Create the markerless tracking functionality, which can be based on finding natural features in the texture of the poster image.
4) Connect the various components into a single iOS application, i.e. develop a general project structure, and register the 3D model according to the tracking parameters.

Any iPhone application must be developed with the XCode IDE programming environment, and XCode version 3.8, with iOS SDK 4.2, was used as developing platform for the implementation, together with an iPhone 3GS as target.

4.1. Creating the virtual content

From the conceptual analysis it was found out that the virtual content of the AR application should consist of a 3D model of an Audi car. As the focus for the thesis is related to how to utilize certain technology, and not directed towards 3D modelling, it was prioritised not to spend extensive amount of time into model a realistic looking Audi car. Instead it was chosen to find and use a royalty free 3D model of an Audi car from the Internet.

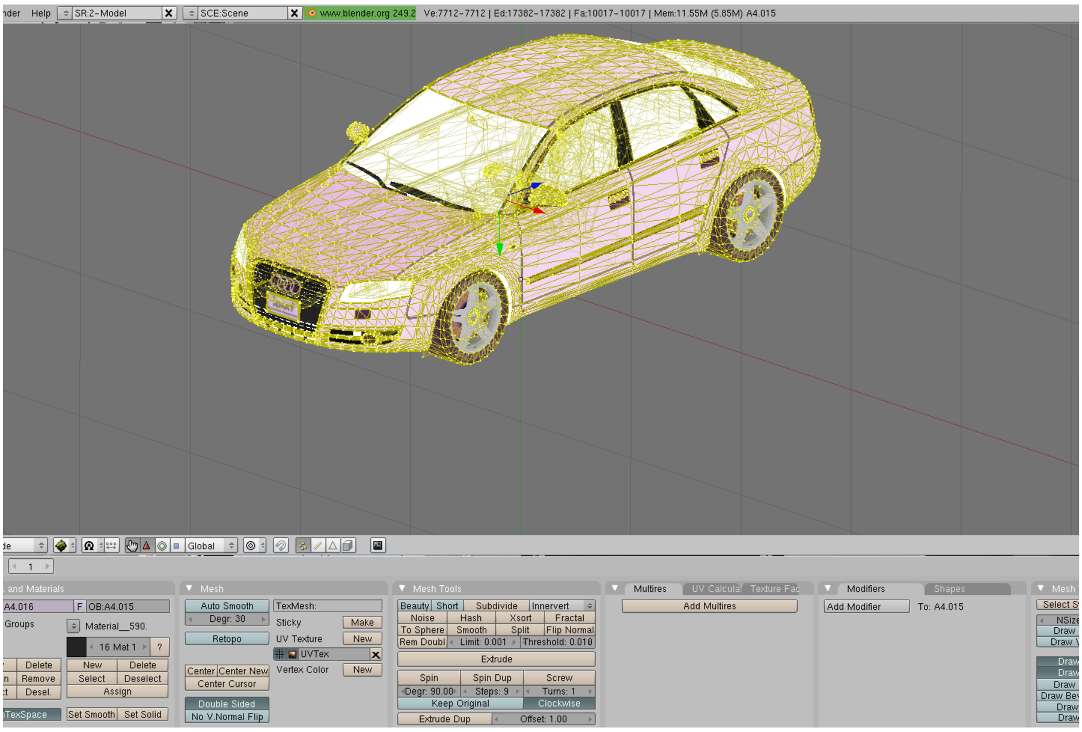
As described in the technical analysis, the iPhone iOS uses OpenGL ES for rendering 3D graphics. However, the iPhone SDK does not provide functionality for importing 3D models, and a model loader or parser therefore has to be either created by the developers themselves or borrowed from external resources. The makers behind the iPhone graphics hardware, Imagination Technologies, have also developed formats and tools, which are optimized for OpenGL ES and the POWERVR graphics chip, which is in the iPhone [web 48]. This 3D technology is called POWERVR, and in relation to 3D model format, the provided format is POWERVR Object data (POD files). Together with POWERVR developers SDK, Imagination Technologies includes a set of utility tools for converting other formats into theirs. For PODs, a Plug-in for the Blender software, called PVRGeoPOD, can be used to export other formats, such as Wavefront OBJ, 3DS, Collada, etc., into POD. In the same way, textures can be converted into the optimised and faster loadable PVR texture format by using the PVRTexTool utility tool.

After several iterations of attempts to successfully render a POD model on the iPhone, the following sections describes the procedure used for the implementation, together with explanatory reasoning.

4.1.1. Final design and implementation of the model

In general, 3D models are made from meshes, which is constituted by an array of polygons that again are made from vertices. In describing the complexity or level of details of a model, the number of polygons can be used. When dealing with a platform, which has limited processing capabilities as well as a limited screen size available, it is reasonable to consider the platform when balancing the complexity of the 3D content. When displayed on a small screen, the details of a complex model are less visible, and would require unnecessary amount of loading time or simply not being able to compile on the device. It is therefore appropriate to consider the trade-off between complexity and loading time as well as judge if the actual complexity can be reduced without loosing quality of appearance. With such considerations in mind, the right balanced royalty free model was found at [web 49]. This model was representing an Audi A4 Quattro, containing material properties and textures, which has been pre-assign to the modal via UV coordinates, and contained a total of 12769 vertices.

For converting the model into POD format, it was chosen to use the open-source 3D modelling software, Blender. As there only can be one texture and set of material properties for each mesh in the model, it was found necessary to activate the “split by material” import property in Blender, when importing the Wavefront model. Having installed the PVRGeoPOD, which consists of a Python script, to Blender, the model could subsequently be exported as a POD, which in this case contained a total of 15 meshes.



Picture 42: 3D model imported to Blender

The POWERVR utility tools provide a POD file viewer and editor, called PVRShaman. After converting each texture of the model from tga to PVR with PVRTexTool, the new textures could therefore be re-assigned to the POD in PVRShaman.

Having obtained a POD model, the next step was actually to have it imported to iOS. For the purpose, the POWERVR SDK version 2.08 was used, and the as suggested by Imagination Technologies it was decided to use one of their project templates, which follows the SDK, as initial template. Hereby the proper POWERVR project and rendering structure is pre-composed and ready to compile on iOS. After studying the various templates, it was settled on using the template called *complex lighting*, as this project was the one that included a vertex and fragment shader, which was found to resemble the illumination needs of the thesis application the most. In this way the OpenGL content was partly created by modifying a project template rather than building a rendering pipeline from scratch, thus saving time for improving the graphical splendour and overall results. As per POWERVR default, all source code for providing the OpenGL functionality is contained within a single cpp source file, which is interfacing the POWERVR SDK. The final cpp file, containing the thesis application's graphical development, can be found in Appendix B, at p. II, as the OGLES2ComplexLighting Class. Whenever there is a function call, which includes glUniform, such code is directed to the vertex and fragment shaders.

As mentioned earlier the iOS SDK does not provide functionality to import 3D models itself but the POWERVR SDK goes about this by loading the model's meshes into vertex buffer objects (VBOs), which is done via the LoadVbos() function. To be able to actually have the 3D model displayed with OpenGL ES, a POWERVR camera has to be initialized by providing a view matrix.

Picture 43: POD as rendered after initial import. Picture 44: POD as rendered after applying a global illumination

As seen on picture 43 and 44 above, the 3D geometry of the POD is not getting visualized merely by importing the model. In order to for the geometry to do so, an illumination source must be set up in OpenGL ES 2.0, i.e. implementing a light source with certain properties as well as a vertex and fragment shader that support them. The shaders in the OGLES2ComplexLighting template supports various lighting setups, such as directional, spot or point lighting, and the chosen lighting type became a global point light with individual diffuse and specular lighting properties for each mesh in the model. On picture 45 below, the model is depicted after having assigned the lighting properties for how the meshes' materials should react to the global light by providing colour parameters via the pMaterial->pfMatSpecular and pMaterial->pfMatDiffuse functions in LoadMaterial(). Picture 46 shows the model after having mapped textures to the meshes' UV coordinates in the LoadTextures() function.


Picture 45: POD as rendered with added colour properties. Picture 46: POD as rendered with added textures

Based on the opinion that such rendering was sufficient for meeting the requirement of providing a naturalistic resemblance of a real Audi car, the work on the virtual content was concluded at this point. The last step was to make the white background transparent in order to be able to have the image stream from the build-in iPhone camera to be displayed underneath the model. This was achieved by making the OpenGL layer include an alpha channel by changing the CAEAGLayer's color format from kEAGLColorFormatRGB565 to kEAGLColorFormatRGBA8, as well as providing an glClearColor() function, with alpha set to zero, followed by a glClear() in the OGLES2ComplexLighting Class' RenderScene () function.

Further implementation, which could improve the appearance of the VO, could be to also enable functionality for the shaders to support transparency when rendering the meshes. Passing the meshes's material property, named *Opacity*, to the shaders could do this, thus making the windscreen glass-like. Another material property that could be passed to the shaders is shininess, which determines how reflective a material should appear. This could for instance make the wheel rims of the car appear more chrome. By digging deeper into the potential, provided by the GLSL, significant improvements towards creating a more naturalistic looking 3D rendering could be facilitated, e.g. smoothing the transitions between the model's faces, etc.

4.2. Creating the poster content

As found out in the preliminary analysis, the poster content should resemble outdoor scenery that suits the nature of the car brand, which is believed to be an city environment in the case of the actual car model. It was furthermore a design requirement that the poster content was somewhat neutral in order not to steal focus from the car, which raises the dilemma that cityscapes usually are bustling and lively locations with various distractive visual elements. Moreover it was reasoned that conflicts in perceptual depth cues between the 2D poster and the 3D VO would become greater the more real-world object there is in the foreground. To count for such issues, it was decided to merely depict the city as a background and use a plane open space, without real-world objects, as foreground, and merge the two by photo editing in Photoshop. Such two pictures was found and bought from the website, called iStockphoto [web 50], and the chosen royalty free photos are displayed below:

Picture 47: Foreground photo. Picture 48: Background photo

After merging the two photos, as well as applying colour gradient and adjusting contrast/brightness in order to create a night-like appearance, the finished result where the following:

Picture 49: Final Poster design

4.3. Creating the tracking functionality

As apparent from the analysis, there is no single best tracking method for AR, and in order to choose the most appropriate method the individual applications should be considered in relation to it.

Because the overall means for estimating the pose either are egocentric *inside-out* or exocentric *outside-in* the natural first step, is to determine which of these directions to take. As learned from the preliminary analysis, today's smartphones contains all necessary hardware for creating AR. Even though a possible solution could be to provide location tracking by attaching cameras and sensor to the environment itself, it seems more obvious to exploit the capabilities of the functionality of the smartphones. When considering the concept idea of making it possible to show the car release on multiple locations simultaneous, such a solution would furthermore be more practical than having to integrate a sensor setup on each location. Therefore it has been chosen design a to egocentric inside-out tracking system for the application.

From analyzing the possibilities of using internal sensors as tool for tracking, it was learned that it would be required to perform a hybrid tracking solution by combing it with a vision-based method. Orientation in 3DOF could be achieved by relying on internal sensor such as for instance the gyroscope, but the array of build-in sensors are not sufficient for establishing the initial and subsequent location of the user, if considering GPS too inaccurate, and accelerometers too imprecise and drifting.

A characteristic of the concept and application design is that the virtual model of the car will be registered within a fixed position on the poster, which serve as an environmental background. Hereby it can be derived that the user are pointing the device towards the poster in order to have the model displayed. It therefore seems logical to use a *texture-based* method and rely on tracking natural features from the poster layout for estimating the initial pose. But as the model is static, in the sense that its registration position is fixed, there is no point in applying neither orientation sensors nor optical flow model for subsequent pose estimations as NFT methods also can count for this. An alternative approach could be to use an *edge-based* method but as this either would require a poster layout with distinct geometrical features or a distance threshold that allows the camera capture distinct geometrical features in the surrounding space, which could be modeled as for instance a CAD model, NFT opens up for more flexibility.

Having settled on utilizing a vision-based NFT method, the subsequent step is to determine a specific NFT approach. Considering that the iPhone has limited amount of available RAM and CPU, a prerequisite for the NFT method is that it is capable of running on the device. Through a comparison between SIFT and FERN it was learned that they both have been successfully ported to smartphone platforms. Furthermore it was apparent that SIFT does not require a longer training phase but on the other hand have slower run-time performance. As it is known that the graphical virtual 3d content should be both high quality and rendered during runtime, it therefore seems more optimal to use FERN or the faster SIFT alternative, SURF. Having delimited the various tracking methods down to two candidates, it was chosen to engage in experiments, which had the purpose of enabling one of these methods to compile on the iPhone device.

4.3.1. Experiments with FERN tracking

As mentioned earlier, the source code for FERN is available for research purposes or otherwise under GNU public license. Generally the FERN software is a set of C++ files, which depends on using the OpenCV library that can be downloaded from [web 51].

As this code originally was intended for being compiled on a PC, the first test run was conducted in Xcode on a PC, together with an external webcam, after having linked the OpenCV library to the project. The test result showed that the tracking of a print out of the default image, which where attached with the source code, indeed was very accurate and reliable.

As earlier described, the FERN library requires a longer training phase. This training phase serves the purpose of generating a tracker file and a detector file that contain information about the natural features in the image [Özuysal et al. 2010]. More specifically, an array of keypoints is found and stored in these files by conducting an image analysis of the reference image, according to the fern algorithms, and it is those keypoint that are used for future pattern recognition. Furthermore these keypoints are structured in a tree-structured hierarchy as ferns, and the amount of both keypoints and ferns are specified by various function parameters, which are passed to the library's planar_pattern_detector_builder class. Such a training phase is therefore only required once, and subsequently only the files are read into memory. Having examined the generated tracker and detector files is where apparent that they, with the size of approximately 20Mb, where too large to be read into the memory of an iPhone. The function parameters would therefore have to be tweaked downwards in order to successfully use the FERN recognition on iOS.

In a paper, conducted by inventors of FERN from the French CVLab, a figure depicts the relationship between the number of used ferns, and the average percentage of correctly recognized image patches over many trials (see figure below) [Özuysal et al. 2010]. Only the upper graph in the figure is of relevance to FERN, and the x-axis shows the number of ferns whereas the y-axis shows the average classification rate in percent, which in other words describes how fast and precise the reference image is being recognised.

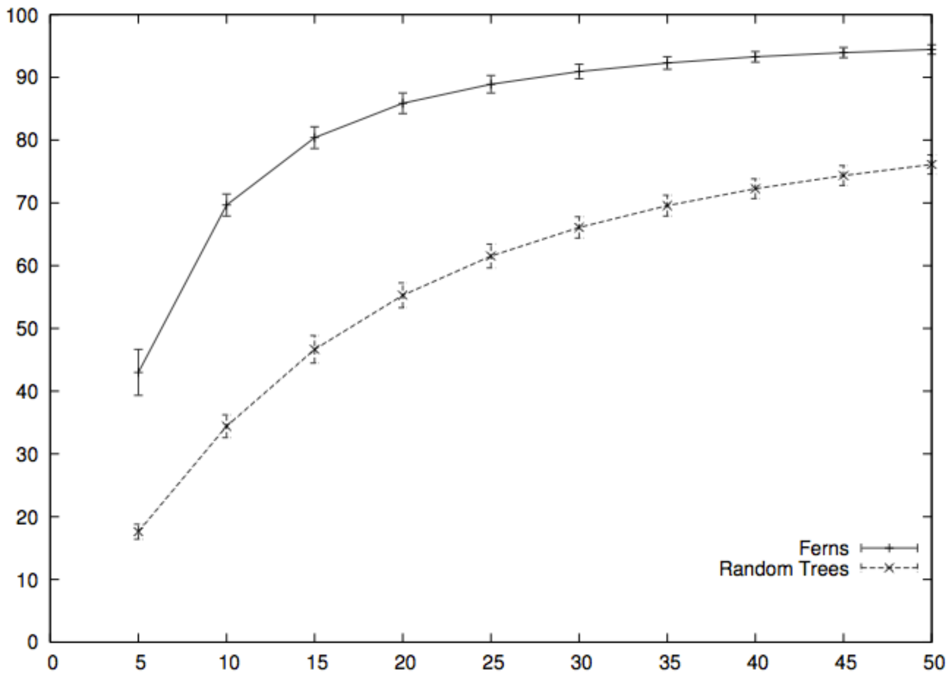


Figure 6: Number of ferns vs. tracking accuracy

By examining the curve of the graph, it is apparent that the number of ferns not is linear proportional to how correctly the image is recognized. On the contrary, the figure shows that the graph more or less is an inverse exponential function, which means that a small increase in ferns improves the recognition more, the fewer the amount of total ferns, i.e. an increase from 10-15 ferns gives greater increase in performance that 40-45. Moreover it can be seen that from an amount around 50 ferns, additionally ferns not are improving the recognition. Due to the tree-structure of keypoints in each single fern, experimental result in reducing the amount of total fern in the project showed that the file size of the tracker file was growing exponential with an increased number of ferns. From such observation it became clear that it was crucial to reduce the amount of ferns as much as possible while sustaining a high percentage of accuracy. Based on the values from the graph, it was chosen to experiment with various numbers of ferns within the range from 10-25 ferns for generating the tracker file, whereas the default number of ferns was 30. As the training phase requires a significant amount of computations, which would be too extensive to perform on the actual iPhone CPU, the PC was used to generate the files. Subsequently it was experimented with loading these file, as well as the general FERN library, on iOS, via an C++ to Objective-C wrapper and iOS project template, which previously had been made by the iOS developer named Andreas Hanft at Neue Digitale / Razorfish. By comparing file size and performance, the appropriate amount of ferns was estimated to be a total of 15 ferns. Because the screen resolution on iPhones are significant smaller than the one of a PC monitor display it was furthermore experimented with altering the parameters, which concerns the amount of keypoints in each fern, in order to eliminate unnecessary pixel accuracy while maintaining same outcome. Here it was found that the initial number of 400 keypoints could be reduced to 200 without affecting the tracking performance significantly, which resulted in an overall reduction in file size of the tracker file from around 20Mb to 164Kb. For an overview of the parameter that was used for generating the tracker file, the reader is referred to the view the initTracker() function in Appendix B. Here the original parameters are marked in parentheses and written with green color at the same code line, to provide a fast overview of the modifications.

From a descriptive readme file, attached to the FERN source code, it was further explained that the software can run in different modes: With both tracker and detector enabled, or tracker only. When both tracker and detector enabled. While in tracker and detector mode, the detector can be used for NFT, independent from how the reference image is located and rotated within the captured video stream. In other words, the keypoint, found during processing and analysing the incoming video images, are being matched against the reference image's keypoints while counting for scale and rotation between the two sets of data. This makes the recognition very powerful and flexible but obviously also requires a significant increase in needed computations. As a step towards optimizing the performance for iOS, it was chosen to bypass and thereby omit this phase, as the recognition was still functional. Another benefit from such a solution was that only the tracker file would need to be loaded into memory. But a major downside from bypassing the detector was however that the tracking system requires for the reference image to be situated within fixed position and orientation within the video images in order to find an initial match. Even though the recognition to follow was allowing for re-scaling a re-positioning, i.e. was allowing for subsequent dynamical change of the users' pose, it was still limiting the 6DOF for the user. Besides for the fact that the user initially would need to aim the camera according to predefined parameters, the main reason for the limitation was due to the need for the tracking system to be re-initialized once the NFT was lost for merely a single frame. Here the detector would resume the tracking at same pose but in tracker only mode the user would need to return to the initial pose. To compensate for the need of a fixed initial pose, it was decided provide a UI with graphical guidance for how to aim the camera correctly during tracking initialization. Another observation, on tracking without a detector, was that there was appearing dynamical tracking errors if the camera was moved in a too fast pace, which caused the tracking to get lost.

After the above-mentioned customizations of the FERN framework, the following framerates where achieved when test compiled on a variety of handheld Apple devices:

1) Framerate on iPhone 3GS: approximately 8fps.
2) Framerate on iPhone 4G: approximately 15fps.
3) Framerate on iPad 2G: approximately 24fps.

The image stream from the iPhone camera, which the FERN framework obviously depends on for the NFT, was obtained via the iOS AVCaptureDeviceInput Class. When the associated keypoints have been recognized, by matching the processed images from the camera against the tracker data, FERN algorithms furthermore output the location of the match as four corner points, i.e. gives four corners as 2D coordinates with respect to where on the screen they are located. To give an illustrative example, the tracked corner points could be A1, B1, C1 and D1, as represented in the figure below (referred to as square 1), whereas the square made from A2, B2, C2 and D2 (referred to as square 2) could represent the region of interest (ROI), which correspond to the corner points of the reference image that the tracker data is generated from.

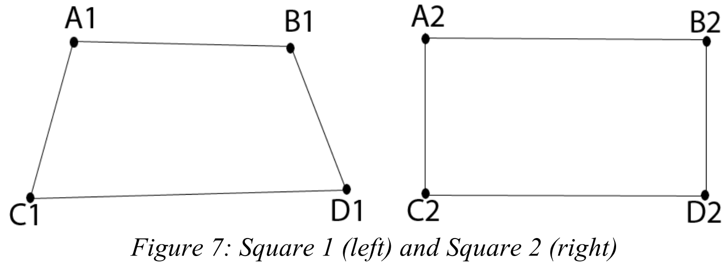


Figure 7: Square 1 (left) and Square 2 (right)

An observed issue with the FERN tracking was that there brief periodically matches was found randomly even though the reference image was not in sight of the camera. To eliminate such static tracking errors, a series of geometrical function checks, to estimate whether square 1 possibly could be any distorted representation of square 2, were implemented as seen in the checkCornerPoints() in Appendix B. By considering a right angled rectangle that are seen from a skewed perspective it is apparent that the vector C1-A1 always will be somewhat parallel to vector D1-B1, give or take a certain threshold tolerance. The same counts for vector B1-A1 and D1-C1, and therefore it was checked whether such relationship of alignment was present for square 1. In the same way it can be predicted that vector D1-C1 should be more or less perpendicular to vector D1-B1 and such a check was therefore also performed. As the ratio between the length of the sum of vector A1-C1 and B1-D1 divided by the length of the sum of vector A1-B1 and C1-D1 should remain consistent, also such a check was added, and hereby the random noise that resulted in incorrect corner points were removed.

4.4. Creating the registration functionality

By considering the actual application concept, with its belonging tracking system, it was decided to use a rather simple registration methodology for positioning the VO. As the VO should be located within fixed position in relation to the 2D poster, and that the poster's four corner points moreover was found via the FERN framework, it was possible to use these corners as reference for the registration. Placing the VO at the center of the poster could be done by translating the VO to the point where the two sets of diagonal corner points intersects (see figure 7), and simple algebra, concerning intersection of two line segments, could thereby be use to find such point. The implementation of calculating the line intersection can be found at Appendix B in the OGLES2ComplexLighting Class' renderScene() function, immediately under the comments stating: "Calculating the line intersection of the diagonal corner points".

The poster's corner points could furthermore be used for determining how the poster had been scaled, as the sum of the above-mentioned line segments gives a relative scaling factor, independent from which angle the poster is seen from, which also could be applied to the VO. The implementation of calculating this scaling factor can also be found in Appendix B in the OGLES2ComplexLighting Class' renderScene() function, where the scaling factor is has the variable name *localScaleFactor*. Due to the restricted time period available for the thesis, the rotations of the VO, with respect to the users' current pose, was intentionally omitted. The reason, it was found acceptable to assign a lower priority to the rotational part of the registration, was based on considering the fact that it was a 2D image being augmented.

4.1. Bringing it all together

Having obtained a the necessary 3D graphics as well as functionally for tracking and registration, the last step in completing the application was to merge these segments into a single Xcode project, which could be compiled for iOS. Additionally the previously mentioned UI, for guiding the user in how to initialize the tracking, had to be added, which was done by adding an extra UIView with graphical reference corners. It was however found appropriate to make these references transparent meanwhile tracking as they were not to be used during this phase. Furthermore the subview, containing the OpenGL graphical content, obviously had to be layered on top of the subview that was displaying the video capture from the AVCaptureDeviceInput Class. Below, the finished result is depicted:

Picture 50: Final application result

5. Composing the test

The intention with the test is not to evaluate the advertisement impact itself, as professionals within the field have conducted the advertisement strategy. Even though the actual design, based upon this concept, was carried out by the author, it is assumable that any resulting lack of advertisement impact could be improved by changing the design, e.g. by also letting professionals carry it out. The interesting focus is therefore not to investigate the application's overall advertisement concept but rather to evaluate how cutting-edge AR technology is experienced when applied to an advertisement concept.

5.1. Overview of general methodology for AR evaluation

5.1.1. Perception, performance, or collaboration

Swan and Gabbard argue that there are three main areas in evaluation of AR applications: *Perception, performance* and *collaboration* [Swan & Gabbard 2005]. Here, *perception* concerns experiments, which study low-level tasks for the purpose of understanding how human perception and cognition operate in AR contexts. *Performance* is related to examination task performance within specific AR applications or application domains, in order to understand how AR technology impacts such underlying tasks. As suggested by the name, *collaboration* involves experiments, which examine how multiple collaborating users interact and communicate. The user task involved with the thesis application is very limited and is restricted to pointing the handheld device at the poster in order to investigate the virtual content, which resembles the car. For such a non-complex user task structure, there is no point in measuring *performance.* In the same way, collaboration is not a significant part of the application concept and is thereby not relevant for answering the problem statement. Understanding how human perception and cognition operate in AR contexts is on the other hand relevant for understanding how AR technology can influence the user experience when utilized for advertising. *Perception* is thereby found to be the relevant area for evaluation.

Having delimited the focus for evaluation to mainly concern the perceptual and cognitive influences of AR, it is appropriate to examine various general methodologies for AR evaluation. Dünser et al. have conducted a survey, where the variety in such AR evaluation methods have been identified and classified by reviewing research publications between the years 1993 and 2007 [Dünser et al. 2008]. An initial 6071 publications was filtered down to 165, where the tendency showed that the majority of evaluations where addressing how to enable technologies (tracking or displays, etc.), as opposed to user experience, which has received only little evaluation. Dünser et al. suggest that a reason for the lack of user evaluation could be related to missing education on how to evaluate AR experiences, i.e. how to properly design experiments, choose the appropriate methods, apply empirical methods, and analyse the results. In the survey, the papers with user evaluation, where classified into the following five types: objective measurements, subjective measurements, qualitative analysis, usability evaluation techniques, and informal evaluation, which are described below.

5.1.2. Objective measurements

These are studies that include objective measurements where the most common measurements are task completion times and accuracy. In general these studies employ a statistical quantitative analysis of the measured variables but can also be limited to merely include a descriptive analysis of the results. As argued above, the non-complex user task structure of the thesis' application does not invite for such measurements, and objective measurements are therefore irrelevant in this context.

5.1.3. Subjective measurements

Subjective measurements are user related measurement conducted through questionnaires, subjective user ratings, or subjective user judgements. In relation to analysis, some of the studies in this category also employ statistical analysis of the results, whereas others only include a descriptive analysis.

5.1.4. Qualitative analysis

Dünser et al.'s category of qualitative analysis contains studies with formal user observations, formal interviews, or classification or coding of user behaviour such as speech or gesture coding.

5.1.5. Usability evaluation techniques

These are publications that employ evaluation techniques that are often used in interface usability evaluations such as heuristic evaluation, expert based evaluation, task analysis, think aloud method, or Wizard of OZ method. For AR applications, where more complex user involvement is afforded or the interface is important, usability evaluation becomes highly relevant. Due to the structure of the thesis application, such an evaluation approach is however irrelevant.

5.1.6. Informal evaluations

In this category we included informal user evaluations such as informal user observations or informal collection of user feedback.

Besides from the classification itself, Dünser et al. does neither evaluate the evaluation methods nor give suggestions on which methods to apply to which scenarios. However they point to the advantage of collecting knowledge on user evaluations in other disciples and to bring it into AR settings, as they state: "*even though the specifics of AR interfaces are different to more traditional interfaces, the basic tools to evaluate user behaviour or perception are quite similar*" [Dünser et al. 2008]. Furthermore they mention that to study peoples' behaviour, which is very common in general Human Computer Interaction (HCI) or Psychology, could be an example of such kind of methods to adapt for evaluating AR experiences.

5.2. Evaluating presence

As described earlier, Lombard and Duch argued that a key in evaluating web related *Interactive Advertisement* could be found in measuring presence. In the same way, presence are often used to evaluate VR experiences where the user is totally immersion in a completely virtual world, which makes it possible to investigate the existing presence related to the experience. To answer whether it also could be a suitable to evaluate presence in an AR context, a further analysis on the topic is necessary.

Lombard and Duch defines presence as follows: *"Presence (a shortened version of the term "telepresence") is a psychological state or subjective perception in which even though part or all of an individual's current experience is generated by and/or filtered through human-made technology, part or all of the individual's perception fails to accurately acknowledge the role of the technology in the experience."* [Lombard 2001]. Such definition suggests presence as being a perceptual state, related to an experience, in which the user is unaware or partly unaware of the technology causing it. In relation to the problem formulation of this thesis such definition is interesting for various reasons. First of all such definition concerns technology and the related perceptual state of the user. This thesis has set out to investigate (AR) technology and such technology's impact on a chosen field, which in this case is advertisement. As advertising obviously is targeting users, it is highly relevant to examine the extent to which the user fails to accurately acknowledge the role of the technology in the experience. Even though the client has the slogan of advancing through technology the intention is not merely to show off new technology but the ultimate goal is rather to keep the experience in focus by making it novel, fascinating and appealing through the use of the technology. To clarify the relationship between the user experience and the perceived role of technology, it therefore seems appropriate to evaluate presence as defined by Lombard. Another interesting point in this definition is concerning the subjective aspect of the perception, which matches how AR has been described and defined as an illusion in the sense that the virtual content should be perceived as part of the real environment without actually being so physically. But even though Lombard provides a general definition of presence with overall characteristics, he also argues that there are various forms of presence, and these should be further examined in order to specify and optimize the evaluation of presence.

5.2.1. Spatial presence

Spatial presence, also called *physical presence*, is related to key sentences such as "a sense of physical space", "perceptual immersion", "transportation" or "a sense of being there" [Lombard 2001]. As with all of Lombard's subdivisions of presence, the sense of presence as occurring when all or part of the user's perception fails to accurately acknowledge the role of technology, which in this case makes it appear as if the user is in a physical location and environment, different from the on actually located in. As AR per definition is meant to keep the user within the actual and real environment, which is being augmented, spatial presence could seem less relevant for AR contexts. However we have seen that the virtual context significantly can alter how the real environment is perceived. As a good example one could argue that The Museum of London's Street Museum application brings the user back in time to a completely different environment even though the user's attention on the real environment is sustained. In this way, a natural user respond to experiencing spatial presence could be: *It seemed as if I was there!*

5.2.2. Sensual presence

Sensual presence, also called *perceptual realism*, is related to key sentences such as "ecological validity", "naturalness", or "tactile engagement" [Lombard 2001]. Sensual presence is related to having the perception of being in a physical environment where the sensory characteristics of technology-evoked stimuli are experiences as if they in fact were real, i.e. corresponds to stimuli as the ones encountered in the real world such as look, sound, feel etc. A natural user respond to sensual presence could be: *It seemed so real!* Dealing with a situation where the virtual content is intended for being perceived as real, via an illusion created by AR, this type of presence becomes interesting. Investigating if the user perceives the virtual content as being real, meanwhile failing to accurately acknowledge the role of AR technology, seems suitable for evaluating how AR can be applied to the given context.

5.2.3. Engagement

Lombard and Duch also argue that engagement is a form of presence. Here engagement should be understood as involvement or psychological immersion when part or all of the user's perception is directed towards objects, events, and/or people created by the technology meanwhile unaware about objects, events, and/or people within the real physical environment. As this conflicts the very fundamental principles of AR, engagement has no relevance in such context.

5.2.4. Social related types of presence

Besides for the above mentioned types of presence, Lombard and Duch point to a range of social related types of presence. One of these is *social realism* in which the technology-generated social characteristics are as real as the ones known from real life. The other branch is *social presence*, where the user fails to accurately acknowledge the technology that makes it appear as the user is communicating with one or more people. Social presence are further divided into *parasocial interaction*, where the user perceives a one-way communication from technology to user as two-way communication, *shared space (transportation),* where remote two-way communication is perceived as taking place within co-located physical space, and *medium as social actor*, where human-computer interaction is perceived as being human-human interaction. These social related types of presence can have relevance for AR applications, which includes social aspects, e.g. collaborative AR, as opposed to the thesis application, which excludes social elements.

5.2.5. Presence in AR contexts

The idea of using presence in order to evaluate AR application is not a novel invention made by the author. For instance, McCall and Braun have previously expressed an opinion that one of the main challenges within AR environments is how to create a unified sense of place and presence between the real and virtual elements. By unified sense of place and presence, they refer to a state in which the user feels as if the virtual elements are as real and natural as those from the real environment and that they are constantly within the overall AR experience [McCall & Braun 2008], where the overall AR experience also can includes time periods without superimposed virtual content. In comparison with the above-described types of presence, such presence corresponds to Lombard's sensual presence. McCall and Braun points out that presence in AR context necessarily is different from presence in VR contexts, and argues that it is equally valid to examine presence in both context but for slightly different reasons. By considering presence as being constituted by perceived physical and social attributes, the difference lies in the fact that such attributes in AR both can be virtual and real as opposed to VR. Thus the sense of AR related presence is according to [McCall & Braun 2008] co-constructed through experiencing both real and virtual elements in combination. That the user personally will interpret such combination to contain meanings and significances, are McCall and Braun's argument for suggesting to evaluate AR as a "unified sense of place and presence", as they ague that a place can be considered as a combination of physical properties, activities and meanings [McCall & Braun 2008]. Even though the AR experience is constituted by such combination of virtual and real elements, [McCall & Braun 2008] argue that the user furthermore can feel more present in the virtual world and not in the overall AR experience, and points to the ability for poorly designed or incoherently chosen virtual content to steal focus for the real environment as an example. McCall and Braun do not provide a general methodology framework for evaluating AR according to their perspective. However they based on their experiences conclude that a combination of qualitative interviews, real-time observation, and video analysis in their case proved to return valid test results.

Regenbrecht and Schubert agrees with McCall and Braun in that evaluating presence is an appropriate way of evaluating AR experiences, as well as in that the character of presence is different than in VR experiences [Regenbrecht & Schubert 2002]. They state that evaluating whether the user has the sense of being there not is working in AR contexts, as AR elicits a different sense of presence, which they describe as "It is here" presence. Therefore they point to evaluating the perceived realness of the VOs, which corresponds to Lombard's *sensual presence* described above. Furthermore Regenbrecht and Schubert argue that it is also relevant to evaluate spatial present. Not in the same sense as in VR, where the user is feeling spatial present by being bodily surrounded by an artificial environment, but rather that the VOs are experienced as co-located in the same space as the body, i.e. that the VOs are perceptually integrated with the real objects.

Based on the reasoning, described above, it was found relevant both to evaluate the perceived sensual presence and spatial presence of the VO, according to Regenbrecht and Schubert's point of view, when investigating how cutting edge handheld Augmented Reality technology affects the user experience. The sensual presence are naturally affected by the conceptual realism of the 3D car in relation to its surroundings, i.e. how realistic the rendering appears in context, whereas the spatial presence is more concerned with how well the 3D car appear to co-exist in the space that are constituted from the poster content, e.g. if the registration of the car causes a sense of such a co-existed space in real-time for a moving user.

5.3. Thesis methodology for evaluating presence

5.3.1. Considering the test scenario

Before settling on a methodology for testing the perceived sensual and spatial presence, it is relevant to consider the location context, which is intended for test. Instead of testing the application within a controlled environment, it was chosen to situate the application in its natural environment, i.e. situate it in an outdoor city environment. Hereby factors such as lighting conditions, external noise influence, distraction from traffic or bypassing people etc. are present during the test, thus resembling a authentic setup, which matches the application concept. By considering such a scenario, a series of reflection was made in relation to finding the most appropriate method for conducting the test, i.e. finding out which of Dünser et al.'s evaluation categories would be the most appropriate approach. As the aim of the test was to evaluate the perceived presence, which should be understood according to the previously described perspective, it was reasoned that the test subject opinion about cars, Audi or the advertisement message in general was unimportant as such opinion not should influence the perceived presence. It was therefore decided to recruit test subjects by addressing random people walking the street, which furthermore would create an authentic way to experience the AR application. But, as it was reasoned that it could be hard to interrupt people's doings and whereabouts by longer qualitative interviews, it was decided to create a shorter questionnaire, which in turn would allow time for having numerous test subject conducting the test, thus providing more general test results on the perceived presence.

By considering presence as a subjective sensation or mental manifestation, Sheridan has argued that subjective measurements generally are the most appropriate way to evaluate presence [Van Baren & Ijsselsteijn 2004]. Inspired from such reasoning, it was decided to give the questionnaire a subjective character. But before creating the actual questions, criteria for presence questionnaires, which has been identified by Lessiter et al., was considered [Van Baren & Ijsselsteijn 2004]. From these criteria, the following was found relevant and therefore adopted:

5.3.2. Criteria for presence questionnaires

1) Understanding of presence should not be assumed by directly asking the test subjects how present they feel. The word "presence" should therefore intentional be left out of the questionnaire.

2) Questions should avoid addressing two issues in one question.

3) Response options should ideally be consistent across items. To meet such criterion, it was decided to use a Likert-type scale but with the significant difference of omitting a neutral centre position, thus forcing the test subject to take either a positive or negative standpoint. Therefore a 6-point scale, ranging from *Strongly disagree* to *Strongly Agree,* was chosen for the questions related to presence. More specifically the chosen point order where as follows: *Strongly disagree, Disagree, Disagree somewhat, Agree somewhat, Agree, Strongly Agree.*

4) Presence is likely to be a multidimensional construct, and questionnaires should reflect this by investigating a range of characteristics, i.e. because of the complexity of presence, the question should address various aspects related to the experience.

5) Questions should not make reference to specific media system and content properties.

6) Questionnaires should be piloted with a sufficient number of test subjects.

5.3.3. Creating the questionnaire

As explained earlier, there is not any commonly agreed upon way to evaluate AR relation presence or AR in general. However Regenbrecht and Schubert have created a questionnaire, which addresses the exact issues concerning both the sensual and spatial presence that are related to the VOs of an AR application [Regenbrecht & Schubert 2002], i.e. addresses the "it is there" sense of presence and not a "being there" form of presence. Furthermore these questions met the above-mentioned criteria for presence questionnaires, were constituted by an appropriate amount of question that can be answered within short duration of time, and were generally found sufficient for answering the thesis' problem formulation. Regenbrecht and Schubert's questions are, except for small customizations to suit the actual AR application content, as follows (Regenbrecht and Schubert's original questions are appended under *Regenbrecht and Schubert's original questions* in the Appendix C):

1): *Was watching the car just as natural as watching the real world?*
2): *Did you have the impression that the car belonged to the poster?*
3): *Did you have the impression that you could have touched and grasped the car?*
4): *Did the car appear to be (visualized) on a screen?*
5): Did you have the impression that the car was located in space?
6): *Did you have the impression of seeing the car as a flat image?*
7): *Did you pay attention at all to the difference between the poster and car?*
8): *Did you see the car as a 3-dimensional object?*

In addition to the above questions, it was found appropriate to address the issue, which concerns the fact that the poster is a 2D representation of a 3D space rather than actually being a 3D space. Therefore an additional question was created:

9): *Was watching the poster just as natural as watching the real world?*
10): *Did you have the impression of seeing the poster space as a three-dimensional space?*

As a general good practise, which could serve in obtaining a more in-dept understanding of the test result, an empty text field was added for general comments on the experience, i.e. raising the question:

11): *Do you have any general comments?*

Besides for the questions that directly address the perceived presence, it was found relevant to consider the test subjects' prior knowledge about AR as well as of using smartphones, due to the fact that it might influence the experience. For this reason, the following two questions was to be initially asked prior to the presence related questions:

a): *Do you consider yourself an experienced smartphone user?*
b): *Do you know what Augmented Reality is?*
c): *If yes – please explain how.*

Furthermore it was decided to note down the test subjects age and gender in order to be able to identify possible age and gender related tendencies. For the purpose of making sure that language barriers would not dilute the test result, the questionnaire was translated into German as seen in Appendix C under *The questionnaire translated into German*. Also an English version of the questionnaire was made for the test in order to enable none-German speakers to participate in the test (English version is appended under *The questionnaire in English*). As it was reasoned that the subjective experience, related to how the test subjects perceived the VO, not would be reflected in the user behaviour, it was decided to omit using user observations during the test. Due to a wish of obtaining gender-independent test results, it was furthermore strived to recruit an equal amount of males and females as subjects.

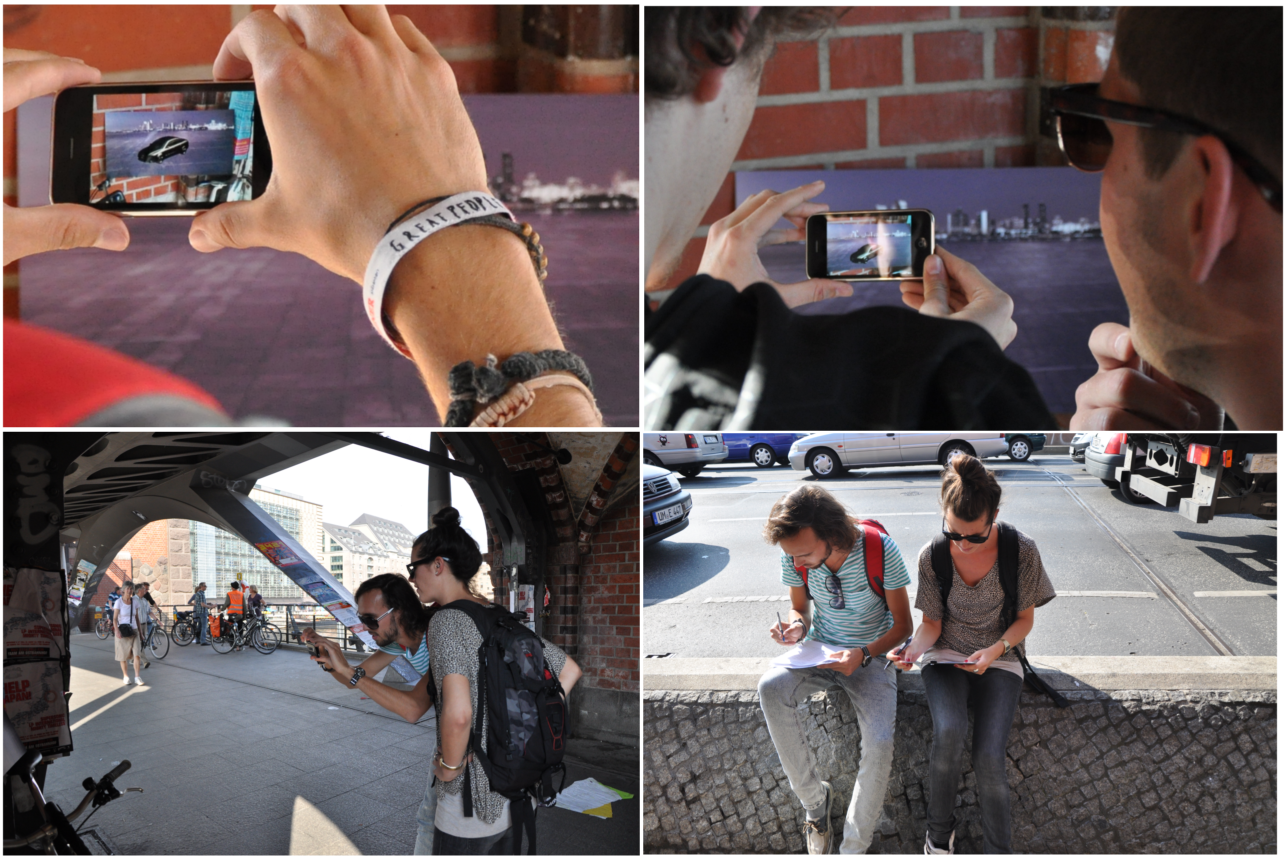
6. Conducting the test

The test was conducted on May 21, 2011, at the Oberbaumbrücke in Berlin, under conditions such as depicted below.

An initial pilot test of the application's tracking functionality was made on the actual location, prior to the user test. This test showed that there were significant light reflections on the poster, if situated in direct illumination from the sun, which caused the tracking not to work properly. The originally area of the bridge, where the poster was intended to hang, was therefore discarded and exchanged for a nearby location within shadow, where a bicycle was used as stand because of missing possibilities for the poster to hang.

Picture 51 & 52: Test setup

Prior to testing the application, the test subjects were informed that it was the application rather than themselves that was being tested. Furthermore they were instructed in how to use the application, e.g. that they should aim the four reference corners to the actual corners of the poster, and they where subsequently asked to fill out the questionnaire in either German or English, depending on preferences. At the same time they were invited to ask, in case the meaning of any question was unclear or not understandable. The testing device used throughout the testing phase was an iPhone 3GS, which means that the application was tested with a resulting lower framerate, around 8 fps.



Picture 53 - 56: Conducting the test

A general impression was that it was quit hard to recruit test subjects merely by approaching them directly on the street, as compared to recruiting subjects within a University faculty. People seemed to be occupied with their daily routines, and around four out of five, addressed people, refused to even listen to why they were approached, and merely shaked their heads to decline any contact. This might also have to do with the fact that it in Berlin is quit common to be addressed by beggars and street sellers, in comparison with for instance Copenhagen, whereby people assumed they were approached for such reasons. That the test subjects, which did get recruited generally seemed to be busy with their daily whereabouts also can be reflected in that only a single subject took time to write a general comment.

A total of twenty test subjects were recruited, divided into eleven males and nine females. The average age was 28,35 years.

7. Discussion

This chapter will discuss the outcome of the conducted test. The results from the questionnaires, used during the test, can be found in appendix D, where they are depicted in bar charts.

The initial questions (a, b and c) were asked in order to collect general information on the test subject experience with smartphone usage, as well as prior knowledge on AR. Such information was intended to aid in obtaining a broader understanding of the subsequent answers, which are related to the perceived presence. The high Standard Deviation of question *a* shows that there was great variety in how experienced the subject were in using smartphones. By comparing the results from the four subjects, who answered *strongly agree*, with the four subjects, who answered *strongly disagree*, there did not seem to be any significant tendencies related to how the subjects were answering, and the discussion can therefore be generalized across smartphone experience. In relation to question b and c, it is noticeable that a vast majority *strongly disagreed* to having prior knowledge on AR whereas the average answer was *disagreeing*. As none of the subjects *strongly agreed* and only three subjects *agreed* or *somewhat agreed*, meanwhile neither proving nor showing it through elaboration on such knowledge (question c), the forthcoming discussion and conclusion are thereby based on the test subjects' impressions from engaging with a new technology for the first time.

Question 1 and 9 was intended as counter questions, which could be used for comparing the perceived sensual presence in the photography and of the VO. As the photography is a direct depiction of the real world, it was initially assumed that the test results would show tendencies within the *agreeing* side of the used scale. Naturally, watching a picture of a real world segment is not exactly the same as watching 2D photography of the same segment, but a mean result, pointing towards *Somewhat disagree,* however shows that the subjects generally might have answered the question according to a very literal and none-abstract understanding of question 9. From such observation it can be assumed that the reality, which was augmented by the application, was perceived as somewhat unnatural, and that question 1 furthermore might have been answered in the same manner of literal understanding. Taking a look on the results for question 10 however shows that the subjects generally *agreed somewhat* to have the impression of seeing the poster space as a three-dimensional space, from which is can be reflected that it probably rather was the banal truth that the photography merely was an representation of the reality than the poster's 2D nature, with the related depth issues, that could have made the poster appear unnatural. Such reflection however has to be left undocumented.

In returning to question 1, it is difficult yet to draw concrete conclusions on the perceived sensual presence, besides for noticing that the appearance of the VO not was perceived as somewhat un-naturalistic. It is however assumed that the test subjects answered according to a literal understanding of the question in the same way, as it was the case for question 9. That neither the poster nor the VO was perceived as significantly naturalistic raises a dilemma: If the not even the complete photorealistic representation of the reality is perceived to be as naturalistic as the real world, how is it possible for the VO to be so? Making completely photorealistic 3D renderings with OpenGL ES and iPhone's hardware is still beyond possible reach, and any ES rendering of today is therefore still only an approximation. For this reason, it was never the intention to achieve photorealism even though the thesis' application concept dictated a naturalistic appearing car model. It would however be scientific incorrect to discard the result from question 1 without further analysis of remaining related results. Taking the opposite perspective on the answers for question 1 and 9, i.e. considering that the VO in fact was seen almost as photorealistic as the photography itself, could however suggest a contradictive strong sensual presence of the VO, but again it has to be considered the replies were in the negative range, which yields the opposite. In this context it is interesting to examine the answers for question 7, about whether the subjects paid attention to the difference between poster and VO, which showed a very clear tendency to *Agree*. The fact that there was perceived significant differences in appearance of poster content and VO reveals that they visually was not supplementing each other very well, and that various differences might have caused a visual conflict, which made neither of the two appear naturalistic. There can be a great variety of rendering related reasons that could have caused this difference, such as for instance un-matching colours or illumination, missing shadows, etc., but yet other possibilities could be that the subjects were considering the difference between the static background and the dynamically appearing/disappearing superimposition, or the more banal fact that the VO visualised a car whereas the background not. Seen in retrospect, the questionnaire should therefore have contained an additional question, which called for the test subject to elaborate on their answers to question 7. Such an additional question could have made it possible to clarify, which of the above mentioned speculations in fact were affecting the conceptual realism. In order investigate the perceptual influences from engaging with a novel technology, it generally would have been interesting also to have subjects with experience with AR technology for comparison. This applies for the discussion as a whole, but in relation to question 7 in particular it might have given a deeper understanding of the replies.

Question 2 makes a fine transition into discussing the spatial presence of the VO, in relation to the poster, as it touches aspects of both sensual and spatial presences. Despite the high variance, within the related answers, the fact that the Mean value is somewhat neutral suggest a correspondence to the findings discussed above, i.e. that there is a coherence between observing a visual difference between poster and car, and having the impression that the car did not belong to the poster. By considering the answers to question 8 and 10, which concerns whether the poster depiction and the VO contained a third dimension, it is apparent that there is an equal, but however only partial, amount of perceived spatiality attached to the VO and the poster content respectively. These slightly positive results are consistent with the previously described slightly negative results from question 1 and 9, which shows that poster and VO only partly are perceived to be as natural as the real world, i.e. having the same physical properties. Moreover there is coherence between *agreeing somewhat* to seeing the VO as 3-dimensional and *disagreeing somewhat* to seeing the VO as merely flat (the result from question 6). Together this also shows that the concept of augmenting the reality, by superimposing 3D graphics onto a 2D representation of a 3D space, not necessarily evokes a conflicting spatial perception of the two, as long as the location 3D content also is fixed, rather than rotating according to the users change in pose, thus actually also can be considered a 2D representation of a 3D object. To investigate the aspects of possible conflict of depth-cues, related to application concept of using a poster as "reality", such rotation would have to be implementing and tested.

To examine the perceived spatial presence of the VO, in relation to the poster content, it is not sufficient to compare the spatiality of the VO and the poster content individually. Therefore it is relevant to compare that there was *somewhat agreed* in the overall application content, having three dimensions, to the fact that there was *somewhat disagreed* on the VO actually belonging to the poster (question 2), which suggest that there the VO was not significantly spatial present within the poster environment. That the Mean of the results to question 5, about whether the car was located in space, points towards *somewhat agree* however shows that the tracking and registration, provided by the application, to certain extend made the VO spatial present within an actual space, rather than merely having spatial 3D properties. There was also *agreed* or *agreed somewhat* to question 4, concerning if the car appeared to be (visualized) on a screen, which raises the question about whether the VO in fact merely was co-existing within the screen space rather than in the real world (the poster space)? An additional, but unfortunately missing, counter question, directed towards whether the poster content also appeared to be visualized on a screen, could have aided in answering such question. That there in average was *somewhat disagreed* to question 3, regarding the impression of being able to touch and grasp the VO, could support such an assumption, by the fact that a screen space could be considered an external and unreachable space. By considering the screen space accessible via the multi-touch features of the screen it could however also suggest the opposite. The answers to question 3 however supports that the perceived sensual presence of the VO was somewhat missing, as its physical tangible properties did not corresponds to stimuli as the ones encountered in the real world, which according to Lombard's definition is a prerequisite for sensual presence.

With the relative high Standard Deviations and especially the general none-distinct answers within the *somewhat disagree* to *somewhat agree* range, throughout the question results, it is hard to draw concrete and well-documented conclusions on the perceived sensual and spatial presences of the VO. In summing up the findings from the analysis above, the following however can be said:

1) Regarding sensual presence, it can be stated to be somewhat missing according to the users perception of the VO. Here the key points was that the VO only partly are perceived to be as natural as the real world, and furthermore not was perceived as having physical properties that correspond to the ones in the real world (touch, grasp). Therefore, there is reason to doubt whether the AR, as implemented with the thesis application, is an appropriate approach for revealing the shapes and contours of a new car brand to the consumers. It is however too drastic to completely discard the concept of an augmented car release, as there is reason to believe that better graphics rendering, better poster graphics, and better coherence between the two could enhance the perceived sensual presence of the VO. The general comment, stating that *the car was a bit unclear, and it could have had more colours and contours*, support this as well as does the significant finding that there generally was *agreed* to having observed differences between car and poster, with a very low Standard Deviation. Due to missing knowledge about what differences consists of; it is impossible to give specific explanations on how to delimit them.

2) In relation to the perceived spatial presence of the VO, it was observed that the VO to some extent was seen as three-dimensional and located in space. This does not necessarily mean that the VO in fact had somewhat spatial presence within the poster environment but rather that the VO contained such spatial properties. Also here it is unknown how the test subjects was interpreting the term *space*, so again an additional elaborative question would have been appropriate. It can however be assumed that the VO had spatial presence within some sort of space, which tells that the registration to some extend created such an illusion. But considering that there was high variance in the replies to whether the VO belonged to the poster, with a somewhat neutral Mean, there was great diversity in how the spatial presence of the VO was perceived, i.e. great diversity in how well the illusion of co-existence was provided with the application. That there were positive replies to whether the VO appeared to be (visualized) on a screen however suggest that the application did not succeed in making the screen appear as a transparent window, through which the reality with its augmentations were seen, as it is the idea with video-see-through AR systems. Low video

framerate, missing rotations (when registering the VO), dynamic tracking errors, poor sensual presence of the VO, un-matching visual coherence between VO and poster, are all factors, which could cause the VO never to be fully present within the poster space. To suggest the amounts of individual influences of each of these factors are would not be scientifically possible when considering the findings in the test results. It could however be interesting to investigate an implemented registration process with rotations would influence the experience spatial presence, as this could afford the user to visually explore the VO from various perspectives interactively, thus revealing the VO's third-dimension dynamically while changing view point. As the general comment stated*: "Furthermore it (the car) could also be able to move".* Hereby it also could be possible to obtain further knowledge about the significance of the previously described conflicts in depth-cues that might exist while displaying dynamic 3D graphics onto 2D backgrounds.

In reflecting on the experience of using Regenbrecht and Schubert's questions, related to sensual and spatial presences, it has been the impression that they were not sufficient for obtaining a wholly understanding of the users' perception of the presences. Even though certain counter questions were added for comparison purposes, the need for additional counter questions has been identified during analysis of the test results. In the same way, Regenbrecht and Schubert's questions have showed not to provide a sufficient precise and clear understanding of how the test subjects have interpreted them. Answers to one of the counter question, added by the author, has however suggested that the subjects generally replied according to a literal and non-abstract understanding of the questions, which should be considered when reviewing the findings. More open questions for the questionnaire and/or a consistent use of additional sub-questions, which forces the test subjects to elaborate, could assumable have provided more clear and detailed test results. Using qualitative interview as methodology, rather than questionnaires, might have provided a more clarified picture on the test subjects perception of the presences, but this would however be on behalf of testing the application in the intended environment: The outdoor cityscape.

8. Conclusion

The following chapter will conclude on the thesis problem formulation, which stated:

"How can cutting edge handheld Augmented Reality technology, utilized within the public domain, affect the user experience when applied to an advertisement context?"

Towards answering such question, it has been investigated what AR is, what constitutes it, and what it can be used for. The latter showed how AR could be applied to various areas for a broad variety of purposes and benefits, and having identified a need for the advertisement industry to adapt to new technologies, this area had been chosen as target for the thesis' AR application. As methodology for applying AR to an advertisement context, a field study was initiated, and the application concept and strategy got developed through collaboration with a professional advertising agency. During such development, it was considered for the application to be directed towards a smartphone platform, which portable and handheld nature holds the potential of bringing AR reality to mobile outdoor settings, as well as contains all build-in means for utilizing egocentric *inside-out* AR tracking systems.

In focusing on the state of the art within related technology, a criterion for the AR system to be cutting edge was set and defined as being a markerless tracking system. To meet such requirement, a NFT tracking system, based on the FERN framework, was implemented and customized/optimized for iOS.
Such modifications however resulted in reduced video framerates, restricted 6DOF, and a need for re-initializing whenever the tracking was lost. The tracking system was furthermore prone to static tracking errors when the tracked reference image was in directs sun illumination, and thereby not a suitable solution for all outdoor conditions. Additionally the system did not allow for occlusion from obstacles between camera and reference image, such as for instance walking people etc.

The application's virtual content was rendered with OpenGL ES 2.0 via the POWERVR SDK, which is optimised for the PowerVR Graphics Chip that are present in newer iPhones. This was done to reduce computations on the CPU, which processes the tracking, and exploit the potential of the GPU. The VO was registered by a translation and scale, which was calculated according to the user current pose, but the corresponding rotations were however not implemented. The registration was thereby not a perfect spatial alignment, which invited for interactive exploration of the VO's three dimensions, and rather a displacement of a 3D object within a 2D coordinate space than an a perspective projection.

The application was tested in an authentic outdoor cityscape with its affect on the user experience in focus. Here it was attempted to measure the perceived sensual and spatial presences (as defined by Lombard et al.) of the VO, in relation to the poster content that constituted the "reality" to be augmented. The data was collected through questionnaires, inspired by Regenbrecht and Schubert, and findings showed that the VO only were experienced as limited sensual presence due to missing "real world" physical properties attached to it perceptually. Furthermore there design-wise were visual incoherencies between the poster and VO, and even though the VO was spatially experienced as located in a space, it was never fully perceived as having spatial presence within the poster environment. There have been given suggestions on why that might be so, but the collected data have proved to be too imprecise to provide a deeper understanding of the different variables individual influences. But it can be concluded that a complete illusion of a unified sense of place and presence, between the real and virtual elements, not was achieved with the thesis application.

9. Future work

Immediate future work, related to the thesis' application implementation, would logically consist in implementing the discussed missing rotation of the VO. Through the analysis, the importance of a correct registration has been pointed out, and such alignment has furthermore been identified as a prerequisite for providing the illusion that the virtual content belongs to the real world. As Ekengren for example stated: *„The objects of the virtual and the real world must be perfectly aligned at all times or the illusion of coexistence will fail"* [Ekengren 2004 p.24]. Having seen that the test subjects only perceived a very limited spatial presence of the VO, it would therefore be interesting to investigate if the possibility to dynamically view the VO's three-dimensions, with the provided 6DOF, would influence the related test results.

As the test showed that the graphics included with poster and VO not where suiting and supplementing each other, further work on the graphical side of the application could also be appropriate, e.g. matching colours, light etc. In the same way it is assumable that an implementation of for instance cast shadows for the rendering could influence such cohesion positively. Furthermore it would probably have caused a greater sense of unification, between poster and VO, if professional graphical designers (from Neue Digitale / Razorfish) actually had developed the graphical content for both poster and 3D graphics. But this would not necessarily have resulted in better scientific research, and as the application never had been intended to become a concrete commercial product, the learning involved with such development was justifiably prioritised. However an extension of the conducted field study, where consultancy with designers were included throughout the thesis' iterative design and implementation phase, might have given insight in visual principles, which could have used in the test questionnaire or for understanding the current test results better.

To elaborate on the application's concept of being a virtual car release, within an augmented real world environment, it can be put into perspective that the current application design not would be suitable for the intended purpose. The missing sense of sensual presence attached to the car, i.e. the absent of perceptual realness, conflicts with the very nature of a car release, where the physical properties of the car model is suppose to convince the consumers to buy the car. It is however too drastically to completely discard the potential of an AR car release, as there are reasons to believe that the sensual presence could be enhanced through modifying the application's general graphical side.

10. Bibliography

Azuma, Ronald T. (1997): *A Survey of Augmented Reality.* In Presence: Teleoperators and Virtual Environments 6, 4 (August 1997), pp. 355-385. http://www.cs.unc.edu/~azuma.

Azuma, Ronald. Baillot, Yohan. Behringer, Reinhold. Feiner, Steven. Julier, Simon. Blair, MacIntyre (2001): *Recent Advances in Augmented Reality.* IEEE Computer Graphics and Applications 2001.

Bay, Herbert. Ess, Andreas. Tuytelaars, Tinne, Gool, Luc Van (2008): *SURF: Speeded Up Robust Features*. Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346-359, 2008.

Billinghurst, Mark (2008): *Usability testing of augmented/mixed reality systems.* ACM SIGGRAPH ASIA 2008 courses.

Bimber, Oliver. Raskar, Ramesh (2005): *Spatial Augmented Reality: Merging Real and Virtual Worlds.* A K Peters, Ltd. 2005. ISBN 1-56881-230-2.

Carmigniani, Julie. Furht, Borko. Anisetti, Marco. Ceravolo, Paolo. Damiani, Ernesto. Ivkovic, Misa (2010): *Augmented reality technologies, systems and applications.* Springer Science&Business Media, LLC 2010.

Chou, C. S.. Welch, R. B.. Gilford, R. M.. Joula, J. F. (1975): *The "ventriloquist effect": Visual dominance or response bias?* Perception & Psychophysics, Vol. 18, No. 1, p. 55–60.

Dow, S.. MacIntyre, B.. Lee, J.. Oezbek, C.. Bolter, J.D.. Gandy, M. (2005): *Wizard of Oz Support throughout an Iterative Design Process.* Pervasive Computing, IEEE vol. 4, No. 4, Oct.-Dec. 2005. ISSN**:** 1536-1268.

Dünser, Andreas. Grasset, Raphaël. Billinghurst, Mark (2008): *A survey of evaluation techniques used in augmented reality studies.* ACM SIGGRAPH ASIA 2008 courses.

Ekengren, Björn (2009): *Mobile Augmented Reality.* Royal Institute of Technology 2009. ISSN-1653-5715.

Fiala, M. (2007): *Magic Mirror and Hand-held and Wearable Augmentations.* Presented at IEEE Virtual Reality, NRC 48811, 2007.

Haller, Michael. Billinghurst, Mark. Thomas, Bruce H. (2007): *Emerging Technologies of Augmented Reality: Interfaces and Design.* IDEA Group publishing 2007. ISBN 1-59904-066-2

Höllerer, Tobias H.. Feiner, Steven K. (2004): *Mobile Augmented Reality.* Telegeoinformatics: Location-Based Computing and Services, Chapter Nine. H Karimi and A. Hammad (eds.) Taylor & Francis Books Ltd., 2004.

Izkara, J.L., Pérez, J., Basogain, X., and Borro, D. (2007): *Mobile Augmented Reality, an Advanced Tool for the Construction Sector*. Bringing ICT knowledge to work: proceedings of CIB 24th W78 Conference, pp. 453-460. Maribor, Slovenia, 2007.

João Paulo S. M. Lima, Veronica Teichrieb, Judith Kelner (2009): *A Standalone Markerless 3D Tracker for Handheld Augmented Reality*. Presented at CoRR, 2009.

Klein, Georg. Murray, David (2009): *Parallel Tracking and Mapping on a Camera Phone.* ISMAR '09 Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality. IEEE Computer Society 2009. ISBN: 978-1-4244-5390-0

Lalanne, Denis. Kohlas, Jürg (2009): *Human Machine Interaction: Research Results of the MMI Program.* LNCS 5440, pp. 280-319. Springer 2009. ISBN: 9783642004360.

Lima, João. Paulo, Simões. Figueiredo, Lucas. Teichrieb, Veronica (2009): *Online Monocular Markerless 3D Tracking for Augmented Reality*, Chapter 1. www.cin.ufpe.br/~in1123/material/livro-svr2009.pdf

Lombard, M.. Snyder-Duch, J. (2001): *Interactive advertising and presence: A framework*. Journal of Interactive Advertising, Vol. 1, No. 2, 2001.

McCall, Rod & Braun, Anne-Kathrin (2009): *Experiences of Evaluating Presence in Augmented Realities.* PsychNology Journal, Vol. 6, No 2, 157-172, 2008.

Meng, Yu (2005): *Implementing the Scale Invariant Feature Transform (SIFT) Method.* http://www.cs.st-andrews.ac.uk/~yumeng/yumeng-SIFT

Milgram, P.. Kishino, F. (1994): *A taxonomy of mixed reality visual displays*. IEICE Trans. Information and Systems, 1994.

Milgram, Paul, Takemura, Haruo, Utsumi, Akira, Kishino, Fumio: Augmented Reality (1994): *A class of displays on the reality-virtuality continuum.* ATR Communication Systems Research Laboratories, Japan. SPIE Vol. 2351, Telemanipulator and Telepresence Technologies, 1994.

Möhring, Mathias. Lessig, Christian. Bimber, Oliver (2005): *Video see-through AR on consumer cell-phones*. Third IEEE and ACM International Symposium on Mixed and Augmented reality, 2005.

Poslad, Stefan (2009): *Ubiquitous Computing: Smart Devices, Environments and Interactions.* Ubiquitous Computing: Smart Devices, Environments and Interactions. John Wiley & Sons, Ltd. 2009. ISBN: 978-0-470-03560-3

Pupilli, Mark. Calway, Andrew (2002): *Real-time Structure from Motion for Augmented Reality.* http://www.cs.bris.ac.uk/~pupilli/publications/y1t

Regenbrecht, H. & Schubert, T. (2002): *Measuring Presence in Augmented Reality Environments: Design and a First Test of a Questionnaire.* Proceedings of the Fifth Annual International Workshop Presence 2002, Porto, Portugal - October 9-11.

Reitmayr, G. Langlotz, T. Wagner, D. Mulloni, A. Schall, G. Schmalstieg, D. Qi Pan (2010): *Simultaneous Localization and Mapping for Augmented Reality*. International Symposium on Ubiquitous Virtual Reality (ISUVR), 2010. P.5-7. ISBN**:** 978-1-4244-7702-9

Rideout, Phillip (2010): *iPhone 3D Programming: Developing Graphical Applications with OpenGL ES.* O'Reilly Media, 2010. ISBN: 978-0-596-80482-4

Riisgaard, søren. Blas, M. R. (2004): *SLAM for Dummies: A Tutorial Approach to Simultaneous Localization and Mapping*. http://www.mendeley.com/research/slam-for-dummies-1/

Roberts, R.. Potthast, C.. Dellaert, F. (2009): *Learning General Optical Flow Subspaces for Egomotion Estimation and Detection of Motion Anomalies*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, 57-64.

Rowe, Helga A. H. (1991): *Intelligence: reconceptualization and measurement.* Lawrence Erlbaum Associates. Psychology Press 1991.

Swan, J. E.. Gabbard, J. L. (2005): *Survey of User-Based Experimentation in Augmented Reality*. Presented at 1st International Conference on Virtual Reality, Las Vegas, Nevada, 2005.

Van Baren, J.. Ijsselsteijn, W. (2004): *Measuring presence: A guide to current measurement approaches*. OmniPres project, IST-2001-39737.

Van Krevelen, D.W.F.. Poelman, R. (2010): *A Survey of Augmented Reality: Technologies, Applications and Limitations.*The International Journal of Virtual Reality, vol. 9, No. 2, June 2010.

Wagner, Daniel. Reitmayr, Gerhard. Mulloni, Alessandro. Drummond, Tom. Schmalstieg, Dieter (2010): *Real-Time Detection and Tracking for Augmented Reality on Mobile Phones*. IEEE Transactions on visualization and computer graphics, Vol. 16, No. 3, May/June 2010.

Wagner, Daniel (2009): *Real-time 3D Tracking with Camera Phones*. http://people.csail.mit.edu/kapu/EG_09_MGW/EG2009_Wagner.pdf

Yilmaz, A.. Javed, O.. Shah, M. (2006): *Object tracking: A survey*. ACM Comput. Surv. Vol. 38, No. 4, Article 13 (Dec. 2006), 45 pages. http://doi.acm.org/10.1145/1177352.1177355

Yu, Fu (2009): *Real Time Augmented Reality on Mobile Phone.* http://www.eecs.umich.edu/~silvio/

Zhou, F., Duh, H.B.L., Billinghurst, M. (2008): *Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR*. Cambridge, UK: 7th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2008), 2008.

Özuysal, Mustafa. Calonder, Michael. Lepetit, Vincent. Fua, Pascal (2010): *Fast Keypoint Recognition using Random Ferns*. IEEE Trans. Pattern Anal. Mach Intell. vol. 32, no. 4, pp. 448-461. March 2010.

11. Web references

Web 1: http://www.google.com/trends?q=augmented+reality&date=all&geo=all&ctab=0&sa=N
Web 2: http://www.abiresearch.com/press/1516-ABI+Research+Anticipates+%E2%80%9CDramatic+Growth%E2%80%9D+for+Augmented+Reality+via+Smartphones
Web 3: http://www.email-marketing-reports.com/wireless-mobile/smartphone-statistics.htm
Web 4: http://www.gamasutra.com/php-bin/news_index.php?story=10809
Web 5: http://www.visiongain.com/Report/564/Mobile-Augmented-Reality-2011-2016
Web 6: http://www.gartner.com/it/page.jsp?id=1447613
Web 7: http://www.businessweek.com/technology/content/nov2009/tc2009112_434755.htm
Web 8: http://www.juniperresearch.com/reports/mobile_augmented_reality
Web 9: http://oxforddictionaries.com/view/entry/m_en_gb0048760#m_en_gb0048760
Web 10: http://www.hastac.org/events/international-symposium-mixed-and-augmented-reality
Web 11: http://developer.apple.com/library/ios/#documentation/3DDrawing/Conceptual/OpenGLES_ProgrammingGuide/Introduction/Introduction.html
Web 12: http://developer.apple.com/library/ios/#technotes/tn2008/tn2199.html
Web 13: http://www.junaio.com/
Web 14: http://www.layar.com
Web 15: http://www.wikitude.org
Web 16: http://androinica.com/2009/12/07/google-goggles-visual-search-and-augmented-reality-app-for-android
Web 17: http://www.web-strategist.com/blog/2009/11/16/video-augmented-reality-at-lego-store-digital-box
Web 18: http://www.mobilecommercedaily.com/2010/07/02/ikea-takes-its-product-catalog-mobile-with-augmented-reality-app
Web 19: http://blog.t-immersion.com/2011/02/01/total-immersion-and-ebay-bring-virtual-%E2%80%98see-it-on%E2%80%99-feature-to-ebays-fashion-app/
Web 20: http://www.sensetrix.com/index.php option=com_content&view=article&id=103&Itemid=196
Web 21: http://www.petapixel.com/2010/05/24/museum-of-london-releases-augmented-reality-app-for-historical:-photos
Web 22: http://archeoguide.intranet.gr/project.htm
Web 23: http://trendland.net/2009/12/16/esquire-augmented-reality-cover-video/#
Web 24: http://ardrone.parrot.com/parrot-ar-drone/de
Web 25: http://site.layar.com/company/blog/layar-creation-tools-buildar
Web 26: http://wikitude.me/
Web 28: http://www.uxmatters.com/mt/archives/2009/08/inside-out-interaction-design-for-augmented-reality.php

Web 30: http://www.hitl.washington.edu/artoolkit/news/index.php
Web 31: http://studierstube.icg.tu-graz.ac.at/handheld_ar/
Web 32: http://www.junaio.com/publisher/llamarker
Web 33: http://blog.t-immersion.com/2010/03/23/total-immersion-to-unveil-first-commercial-markerless-tracking-on-mobile-at-international-ctia-wireless-2010-in-las-vegas/
Web 34: http://www.junaio.com/publisher/junaioglue
Web 35: http://www.junaio.com/publisher/main
Web 36: http://studierstube.icg.tu-graz.ac.at/handheld_ar/stbtracker.php
Web 37: http://www.neue-digitale.de/#/Referenzen/Projekte/
Web 38: http://www.parallax.com/tabid/768/ProductID/92/Default.aspx
Web 39: http://www.tinmith.net/wayne/thesis/piekarski-thesis.htm
Web 40: http://opencv.willowgarage.com/wiki/TemplateMatching
Web 41: http://cvlab.epfl.ch/
Web 42: http://cvlab.epfl.ch/software/ferns/index.php
Web 43: http://studierstube.icg.tu-graz.ac.at/handheld_ar/nft.php
Web 44: http://www.robots.ox.ac.uk/~gk
Web 45: http://www.robots.ox.ac.uk/~gk/PTAM/download.html
Web 46: http://usa.autodesk.com
Web 47: http://www.blender.org
Web 48: http://www.imgtec.com/powervr/insider/powervr-sdk.asp
Web 49: http://www.gfx-3d-model.com/2009/04/audi-a4-32-quattro-3d-model
Web 50: http://www.istockphoto.com/
Web 51: http://sourceforge.net/projects/opencvlibrary
Web 52: http://www.smartdraw.com/resources/tutorials/uml-statechart-diagrams/#/resources/tutorials/UML-Statechart-Diagrams
Web 53: http://www.agilemodeling.com/artifacts/stateMachineDiagram.htm

12. Figure references

Figure 1: [Milgram et al. 1994]
Figure 2: [Bimber & Raskar 2005 p.21]
Figure 3: [Haller et al. 2007]
Figure 4: http://www.siddjain.com/TiltIllusion/explanation.html
Figure 5: [Lima et al. 2009]
Figure *6*: [Özuysal et al. 2010]
Figure 7: Private figure

13. Picture references

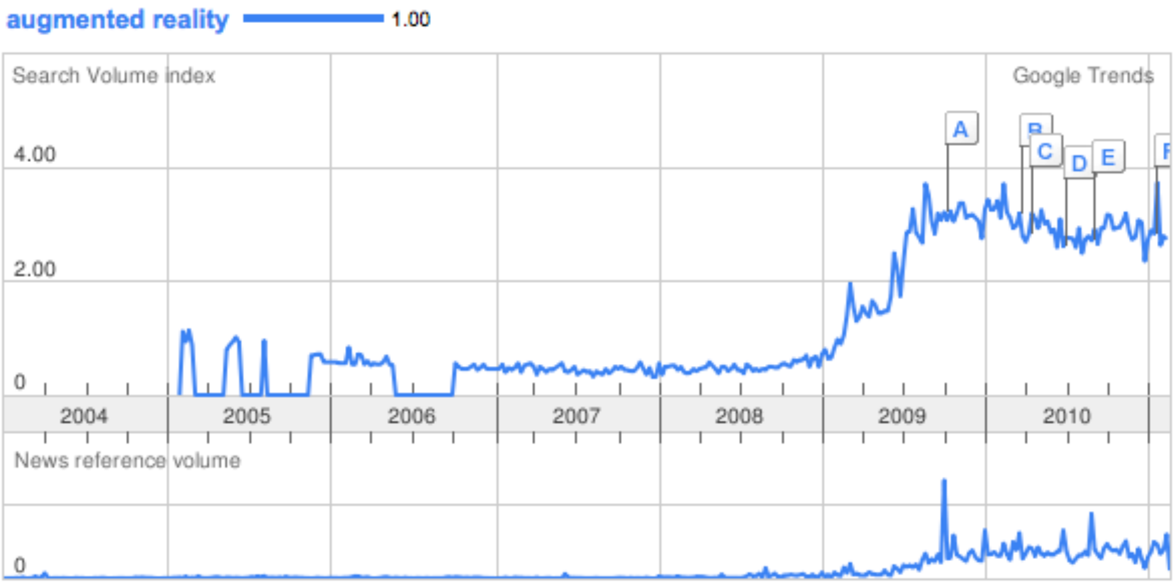
Picture 1: http://www.halolz.com/2008/05/
Picture 2: http://www.adverblog.com/archives/004140.htm

Picture 3: http://electronics.howstuffworks.com/gadgets/audio-music/reactable3.htm
Picture 4: http://www.opentechinc.com/vewl.html
Picture 5: http://www.psfk.com/2010/12/augmented-reality-app-adds-an-ebay-layer-to-the world.html
Picture 6: http://museicapitolini.net/wikitude.xql
Picture 7: http://www.bitmanagement.de/products/authoring-tools/bs-screen-library
Picture 8: http://www.flickr.com/photos/metaio/page2
Picture 9 & 10: http://hi.baidu.com/vrar/blog/item/bc15bffd5530810408244d52.html
Picture 11 & 12: http://www.onlineandmobile.net/2009_12_01_archive.html
Picture 13 & 14: http://ebayinkblog.com/2011/01/07/see-it-on-sunglasses-feature
Picture 15: http://www.petapixel.com/2010/05/24/museum-of-london-releases-augmented-reality-app-for-historical photos/
Picture 16: http://trendland.net/2009/12/16/esquire-augmented-reality-cover-video/#
Picture 17: http://telayar.com/video-gaming-takes-to-the-skies/
Picture 18: http://singularityhub.com/wp-content/uploads/2010/06/AR-Drone-Parrot-New-Games.jpg
Picture 19 & 20: https://www.icg.tugraz.at/~daniel/HistoryOfMobileAR
Picture 21: http://www.m3oyq.com
Picture 22: Private photo
Picture 23-26: Private photos
Picture 27: http://wapedia.mobi/en/Euler_angles?t=5
Picture 28: http://news.cnet.com/8301-13924_3-20008247-64.html
Picture 29: http://itunes.apple.com/us/app/n-o-v-a-near-orbit-vanguard/id343596730?mt=8
Picture 30: http://kbbcollective.com/2010/11/page/2
Picture 31: http://opencv.willowgarage.com/documentation/feature_detection.html
Picture 32-35: http://cvlab.epfl.ch/alumni/oezuysal/ferns.html
Picture 36 & 37: www.cs.st-andrews.ac.uk/~yumeng/yumeng-SIFTreport-5.18_bpt.pdf
Picture 38: http://www.vision.ee.ethz.ch/~surf/papers.html
Picture 39-41: http://www.roborealm.com/help/Optical_Flow.php
Picture 42: Private screenshot
Picture 43 & 44: Private screenshots
Picture 45 & 46: Private screenshots
Picture 47: http://www.istockphoto.com/stock-photo-15965744-beijing-national-aquatics-center-the-water-cube.php?st=0e0a1bf
Picture 48: http://www.istockphoto.com/stock-photo-15970933-san-diego.php
Picture 49: Private picture (edited from picture 47 & 48)
Picture 50: Private photo
Picture 51 & 52: Private photos
Picture 53-56: Private photos

Appendix A

Google Trends statistics

Screenshot from Google Trends ([web 1]) showing a graph of the average search traffic for the term *Augmented Reality* during a given period of time (the upper graph). On the X-axis the actual years are shown whereas the Y-axis shows a scale where 1.0 is a total average. The graph shows that web searches on augmented reality approximately tripled during 2009 without significant degrease throughout 2010.



UML 2 State Machine Diagram

In UML 2.0 (Unified Modeling Language), a State machine diagram depicts the dynamic flow of control from state to state within a system and is therefore appropriate to use for illustrating event-driven systems [web 52]. In this context, controls are generally external stimuli, i.e. user interaction, which causes part of the system (objects) to change state. The following diagram will show a UML State Machine Diagram of the thesis application, according to the previously described flow of events, i.e. the various objects' states and transitions between them. For more information about the UML syntax, symbols and guidelines, the reader is referred to for instance [web 53].

Appendix B

Source code

The OGLES2ComplexLighting Class

```
/******************************************************************************

 @File         OGLES2ComplexLighting.cpp
 @Title        Complex Lighting
 @Version
 @Copyright    Copyright (C)  Imagination Technologies Limited.
 @Platform     Independent
 @Description  Shows how to impliment directional, point, and spot lights.

******************************************************************************/
#include <math.h>
#include "PVRShell.h"
#include "OGLES2Tools.h"
#import "FERNProcessor.h";
#import "ACTrackingViewControler.h"

/******************************************************************************
 Constants
******************************************************************************/

// Camera constants used to generate the projection matrix
const float CAM_FOV  = PVRT_PI / 6;
const float CAM_NEAR = 1.0f;
//const float CAM_FOV  = 0.03f;
//const float CAM_NEAR = 0.2f;

// Index to bind the attributes to vertex shaders
const int VERTEX_ARRAY   = 0;
const int NORMAL_ARRAY   = 1;
const int TEXCOORD_ARRAY = 2;

// Enum and List to select lighting type
enum ELightType
{
      eDirectionalDiffuse = 0,
      eDirectionalDiffuseSpecular,
      ePointDiffuse,
      ePointDiffuseSpecular,
      eSpotDiffuse,
      eSpotDiffuseSpecular,
      eNumLightTypes,
};
const char* const c_aszLightTypeList[] = {
      "Directional Light (Diffuse)",
      "Directional Light (Diffuse+Specular)",
      "Point Light (Diffuse)",
      "Point Light (Diffuse+Specular)",
      "Spot Light (Diffuse)",
      "Spot Light (Diffuse+Specular)"
};

/******************************************************************************
 Content file names
******************************************************************************/

// Source and binary shaders
const char c_szFragShaderSrcFile[]   = "FragShader.fsh";
```

```
const char c_szFragShaderBinFile[]   = "FragShader.fsc";
const char c_szVertShaderSrcFile[]   = "VertShader.vsh";
const char c_szVertShaderBinFile[]   = "VertShader.vsc";

// PVR texture files
//const char c_szTextureFile[]              = "Basetex.pvr";

const char c_szTextureFile1[]        = "TIRE.pvr"; //ok
const char c_szTextureFile2[]        = "LICENSE.pvr"; //ok
const char c_szTextureFile3[]        = "BRAKDISC.pvr"; //ok
const char c_szTextureFile4[]        = "BOTTOM.pvr"; //ok
const char c_szTextureFile5[]        = "TREAD.pvr";
const char c_szTextureFile6[]        = "TIREBACK.pvr";
const char c_szTextureFile7[]        = "BRAKDISC.pvr";

// POD scene files (the car model)
const char c_szSceneFile[]                 = "Mask.pod";

/*!****************************************************************************
 Class implementing the PVRShell functions.
******************************************************************************/
class OGLES2ComplexLighting : public PVRShell
{

      // 3D Model
      CPVRTModelPOD    m_Scene;

      // Projection and view matrices
      PVRTMat4 m_mProjection, m_mView;

      FERNCornerPoints        m_Corners;
      CGPoint bottomLeft, bottomRight, topLeft, topRight;
      float intersectCheckX, intersectCheckY, intersectPointX, intersectPointY;

      // OpenGL handles for shaders, textures and VBOs
      GLuint m_uiVertShader;
      GLuint m_uiFragShader;

      GLuint m_uiTexture1;
      GLuint m_uiTexture2;
      GLuint m_uiTexture3;
      GLuint m_uiTexture4;
      GLuint m_uiTexture5;
      GLuint m_uiTexture6;
      GLuint m_uiTexture7;

      GLuint* m_puiVbo;
      GLuint* m_puiIndexVbo;

      // Group shader programs and their uniform locations together
      struct
      {
        GLuint uiId;
        GLuint uiMVPMatrixLoc;
        GLuint uiModelViewLoc;
        GLuint uiModelViewITLoc;
        GLuint uiLightSelLoc;
        GLuint uiLightPosLoc;
        GLuint uiLightDirLoc;
        GLuint uiLightColorLoc;
        GLuint uiLightDiffuseLoc;
        GLuint uiLightShininessLoc;
        GLuint uiLightSpecularLoc;
        GLuint uiLightOpacityLoc;

      }
      m_ShaderProgram;

      // Array to lookup the textures for each material in the scene
      GLuint*  m_puiTextures;

      // Model rotation
```

```
	float m_fAngle_x;
	float m_fAngle_y;
	float m_fAngle_z;
	float m_fTranslate_x;
	float m_fTranslate_y;
	float m_fTranslate_z;

	// Lighting parameters
	ELightType	m_eLightType;

public:
	virtual bool InitApplication();
	virtual bool InitView();
	virtual bool ReleaseView();
	virtual bool QuitApplication();
	virtual bool RenderScene();

	bool LoadTextures(CPVRTString* pErrorStr);
	void LoadMaterial(int i32Index);
	bool LoadShaders(CPVRTString* pErrorStr);
	void LoadVbos();

	void DrawMesh(int i32NodeIndex);
};

/****************************************************************************
 * Function Name  : LoadMaterial
 * Input		  : i32Index index into the material list
 * Description    : Loads the material index
 ****************************************************************************/

void OGLES2ComplexLighting::LoadMaterial(int i32Index)
{
	//	Load the model's material
	SPODMaterial* pMaterial = &m_Scene.pMaterial[i32Index];

	PVRTVec4 fMat_3;
	PVRTVec4 fMat_1;
	int i;
	fMat_3.ptr()[3] = 1.0f;
	fMat_1 = 0.0f;

	for(i = 0; i < 3; ++i)
	  fMat_3.ptr()[i] = pMaterial->pfMatSpecular[i];
	glUniform3fv(m_ShaderProgram.uiLightSpecularLoc, 1, fMat_3.ptr());

	for(i = 0; i < 3; ++i)
	  fMat_3.ptr()[i] = pMaterial->pfMatDiffuse[i];
	glUniform3fv(m_ShaderProgram.uiLightDiffuseLoc, 1, fMat_3.ptr());


	fMat_1 = pMaterial->fMatOpacity;
	glUniform3fv(m_ShaderProgram.uiLightOpacityLoc, 1, fMat_1.ptr());
}

/*!***************************************************************************
 @Function		LoadTextures
 @Output		pErrorStr		A string describing the error on failure
 @Return		bool			true if no error occured
 @Description	Loads the textures required for this training course
*****************************************************************************/
bool OGLES2ComplexLighting::LoadTextures(CPVRTString* const pErrorStr)
{

	if(PVRTTextureLoadFromPVR(c_szTextureFile1, &m_uiTexture1) != PVR_SUCCESS)
	{
	  *pErrorStr = "ERROR: Failed to load texture.";
	  return false;
	}
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
```

```
	if(PVRTTextureLoadFromPVR(c_szTextureFile2, &m_uiTexture2) != PVR_SUCCESS)
	{
	  *pErrorStr = "ERROR: Failed to load texture.";
	  return false;
	}
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);

	if(PVRTTextureLoadFromPVR(c_szTextureFile3, &m_uiTexture3) != PVR_SUCCESS)
	{
	  *pErrorStr = "ERROR: Failed to load texture.";
	  return false;
	}
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);

	if(PVRTTextureLoadFromPVR(c_szTextureFile4, &m_uiTexture4) != PVR_SUCCESS)
	{
	  *pErrorStr = "ERROR: Failed to load texture.";
	  return false;
	}
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);

	if(PVRTTextureLoadFromPVR(c_szTextureFile5, &m_uiTexture5) != PVR_SUCCESS)
	{
	  *pErrorStr = "ERROR: Failed to load texture.";
	  return false;
	}
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);

	if(PVRTTextureLoadFromPVR(c_szTextureFile6, &m_uiTexture6) != PVR_SUCCESS)
	{
	  *pErrorStr = "ERROR: Failed to load texture.";
	  return false;
	}
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);

	if(PVRTTextureLoadFromPVR(c_szTextureFile7, &m_uiTexture7) != PVR_SUCCESS)
	{
	  *pErrorStr = "ERROR: Failed to load texture_7.";
	  return false;
	}
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
	glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);

	return true;
}

/*!****************************************************************************
 @Function		LoadShaders
 @Output		pErrorStr		A string describing the error on failure
 @Return		bool			true if no error occured
 @Description	Loads and compiles the shaders and links the shader programs
				required for this training course
******************************************************************************/
bool OGLES2ComplexLighting::LoadShaders(CPVRTString* pErrorStr)
{
	/*
	  Load and compile the shaders from files.
	  Binary shaders are tried first, source shaders
	  are used as fallback.
	*/
	if (PVRTShaderLoadFromFile(
			c_szVertShaderBinFile, c_szVertShaderSrcFile, GL_VERTEX_SHADER, GL_SGX_BINARY_IMG,
&m_uiVertShader, pErrorStr) != PVR_SUCCESS)
	{
	  return false;
	}
```

```
	if (PVRTShaderLoadFromFile(
			c_szFragShaderBinFile, c_szFragShaderSrcFile, GL_FRAGMENT_SHADER, GL_SGX_BINARY_IMG,
&m_uiFragShader, pErrorStr) != PVR_SUCCESS)
	{
	 return false;
	}

	//Set up and link the shader program
	const char* aszAttribs[] = { "inVertex", "inNormal", "inTexCoord" };
	if (PVRTCreateProgram(&m_ShaderProgram.uiId, m_uiVertShader, m_uiFragShader, aszAttribs, 3,
pErrorStr) != PVR_SUCCESS)
	{
	 PVRShellSet(prefExitMessage, pErrorStr->c_str());
	 return false;
	}
	// Set the sampler2D variable to the first texture unit
	glUniform1i(glGetUniformLocation(m_ShaderProgram.uiId, "sTexture"), 0);

	// Store the location of the MVP matrix uniform for later use
	m_ShaderProgram.uiMVPMatrixLoc		= glGetUniformLocation(m_ShaderProgram.uiId,
"MVPMatrix");
	m_ShaderProgram.uiModelViewLoc		= glGetUniformLocation(m_ShaderProgram.uiId,
"ModelView");
	m_ShaderProgram.uiModelViewITLoc	= glGetUniformLocation(m_ShaderProgram.uiId,
"ModelViewIT");
	m_ShaderProgram.uiLightSelLoc		= glGetUniformLocation(m_ShaderProgram.uiId,
"iLightSel");
	m_ShaderProgram.uiLightPosLoc		= glGetUniformLocation(m_ShaderProgram.uiId,
"LightPosition");
	m_ShaderProgram.uiLightDirLoc		= glGetUniformLocation(m_ShaderProgram.uiId,
"LightDirection");
	m_ShaderProgram.uiLightColorLoc			= glGetUniformLocation(m_ShaderProgram.uiId,
"LightColor");
	m_ShaderProgram.uiLightDiffuseLoc	= glGetUniformLocation(m_ShaderProgram.uiId,
"DiffuseLight");
	m_ShaderProgram.uiLightSpecularLoc	= glGetUniformLocation(m_ShaderProgram.uiId,
"SpecularLight");
	m_ShaderProgram.uiLightShininessLoc = glGetUniformLocation(m_ShaderProgram.uiId,
"cShininess");
	//m_ShaderProgram.uiLightOpacityLoc = glGetUniformLocation(m_ShaderProgram.uiId,
"cShininess");

	return true;
}

/*!****************************************************************************
 @Function		LoadVbos
 @Description	Loads the mesh data required for this training course into
				vertex buffer objects
******************************************************************************/
void OGLES2ComplexLighting::LoadVbos()
{
	if (!m_puiVbo)      m_puiVbo = new GLuint[m_Scene.nNumMesh];
	if (!m_puiIndexVbo) m_puiIndexVbo = new GLuint[m_Scene.nNumMesh];

	/*
	 Load vertex data of all meshes in the scene into VBOs

	 The meshes have been exported with the "Interleave Vectors" option,
	 so all data is interleaved in the buffer at pMesh->pInterleaved.
	 Interleaving data improves the memory access pattern and cache efficiency,
	 thus it can be read faster by the hardware.
	*/
	glGenBuffers(m_Scene.nNumMesh, m_puiVbo);
	for (unsigned int i = 0; i < m_Scene.nNumMesh; ++i)
	{
	 // Load vertex data into buffer object
	 SPODMesh& Mesh = m_Scene.pMesh[i];
	 unsigned int uiSize = Mesh.nNumVertex * Mesh.sVertex.nStride;

	 glBindBuffer(GL_ARRAY_BUFFER, m_puiVbo[i]);
	 glBufferData(GL_ARRAY_BUFFER, uiSize, Mesh.pInterleaved, GL_STATIC_DRAW);
```

```
		// Load index data into buffer object if available
		m_puiIndexVbo[i] = 0;

		if (Mesh.sFaces.pData)
		{
			glGenBuffers(1, &m_puiIndexVbo[i]);
			uiSize = PVRTModelPODCountIndices(Mesh) * sizeof(GLshort);
			glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, m_puiIndexVbo[i]);
			glBufferData(GL_ELEMENT_ARRAY_BUFFER, uiSize, Mesh.sFaces.pData, GL_STATIC_DRAW);
		}
	}
	glBindBuffer(GL_ARRAY_BUFFER, 0);
	glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, 0);
}


/*!****************************************************************************
 @Function		InitApplication
 @Return		bool		true if no error occured
 @Description	Code in InitApplication() will be called by PVRShell once per
				run, before the rendering context is created.
				Used to initialize variables that are not dependant on it
				(e.g. external modules, loading meshes, etc.)
				If the rendering context is lost, InitApplication() will
				not be called again.
******************************************************************************/
bool OGLES2ComplexLighting::InitApplication()
{
	m_puiVbo = 0;
	m_puiIndexVbo = 0;

	// Get and set the read path for content files
	CPVRTResourceFile::SetReadPath((char*)PVRShellGet(prefReadPath));

	// Load the scene
	if (m_Scene.ReadFromFile(c_szSceneFile) != PVR_SUCCESS)
	{
	  PVRShellSet(prefExitMessage, "ERROR: Couldn't load the .pod file\n");
	  return false;
	}

	m_fAngle_x = 0.0f;
	m_fAngle_y = 0.0f;
	m_fAngle_z = 0.0f;
	m_fTranslate_x = 0.0f;
	m_fTranslate_y = 0.0f;
	m_fTranslate_z = 0.0f;
	intersectCheckX = intersectCheckY = 0.0f;
	intersectPointX = intersectPointY = 0.0f;

	m_eLightType = ePointDiffuseSpecular;

	return true;
}

/*!****************************************************************************
 @Function		QuitApplication
 @Return		bool		true if no error occured
 @Description	Code in QuitApplication() will be called by PVRShell once per
				run, just before exiting the program.
				If the rendering context is lost, QuitApplication() will
				not be called.x
******************************************************************************/
bool OGLES2ComplexLighting::QuitApplication()
{
	// Free the memory allocated for the scene
	m_Scene.Destroy();

	delete [] m_puiVbo;
	delete [] m_puiIndexVbo;
```

```
    return true;
}

/*!*****************************************************************************
 @Function          InitView
 @Return            bool          true if no error occured
 @Description  Code in InitView() will be called by PVRShell upon
                    initialization or after a change in the rendering context.
                    Used to initialize variables that are dependant on the rendering
                    context (e.g. textures, vertex buffers, etc.)
******************************************************************************/
bool OGLES2ComplexLighting::InitView()
{
      CPVRTString ErrorStr;

      //Initialize VBO data
      LoadVbos();

      //Load textures
      if (!LoadTextures(&ErrorStr))
      {
        PVRShellSet(prefExitMessage, ErrorStr.c_str());
        return false;
      }

      //Load and compile the shaders & link programs
      if (!LoadShaders(&ErrorStr))
      {
        PVRShellSet(prefExitMessage, ErrorStr.c_str());
        return false;
      }

      // Set default light color and position
      float afLightPosition[] = { -252, -520, -120 };
      float afLightDirection[] = { 0, 0, 1 };
      //float afLightColor[] = { 1, 1, 1 };
      float afLightColor[] = { 255.0/138.0, 255.0/122.0, 255.0/157.0 };// BGR and not RGB!

      glUniform3fv(m_ShaderProgram.uiLightPosLoc, 1, afLightPosition);
      glUniform3fv(m_ShaderProgram.uiLightDirLoc, 1, afLightDirection);
      glUniform3fv(m_ShaderProgram.uiLightColorLoc, 1, afLightColor);

      // Is the screen rotated?
      bool bRotate = PVRShellGet(prefIsRotated) && PVRShellGet(prefFullScreen);

      /*
        Calculate the projection and view matrices
      */
      float fAspect = PVRShellGet(prefWidth) / (float)PVRShellGet(prefHeight);
      //m_mProjection = PVRTMat4::PerspectiveFovFloatDepthRH(CAM_FOV, fAspect, CAM_NEAR,
PVRTMat4::OGL, bRotate);
      m_mProjection = PVRTMat4::PerspectiveFovFloatDepthRH(CAM_FOV, fAspect, CAM_NEAR,
PVRTMat4::OGL, bRotate);
      m_mView = PVRTMat4::LookAtRH(PVRTVec3(0, 0, 150), PVRTVec3(0, 0, 0), PVRTVec3(0, 1, 0));

      /*
        Set OpenGL ES render states needed for this training course
      */
      // Enable backface culling and depth test
      glCullFace(GL_BACK);
      glEnable(GL_CULL_FACE);

      // Enable z-buffer test
      // We are using a projection matrix optimized for a floating point depth buffer,
      // so the depth test and clear value need to be inverted (1 becomes near, 0 becomes far).
      glEnable(GL_DEPTH_TEST);
      glDepthFunc(GL_GEQUAL);
      glClearDepthf(0.0f);

      // Use a nice black as clear colour
      glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
```

```
	/*
	 Initialise an array to lookup the textures
	 for each material in the scene.
	 */
	m_puiTextures = new GLuint[m_Scene.nNumMaterial];

	for(unsigned int i = 0; i < m_Scene.nNumMaterial; ++i)
	{
	  m_puiTextures[i] = 0;

	  SPODMaterial* pMaterial = &m_Scene.pMaterial[i];

	  if(strcmp(pMaterial->pszName, "Tire_Sidewall") == 0)
	  {
		m_puiTextures[i] = m_uiTexture1;
	  }
	  else if(strcmp(pMaterial->pszName, "License") == 0)
	  {
		m_puiTextures[i] = m_uiTexture2;
	  }
	  else if(strcmp(pMaterial->pszName, "Brake_Disc") == 0)
	  {
		m_puiTextures[i] = m_uiTexture3;
	  }
	  else if(strcmp(pMaterial->pszName, "Bottom") == 0)
	  {
		m_puiTextures[i] = m_uiTexture4;
	  }
	  else if(strcmp(pMaterial->pszName, "Tire_Back") == 0)
	  {
		m_puiTextures[i] = m_uiTexture5;
	  }
	  else if(strcmp(pMaterial->pszName, "Tire_Thread") == 0)
	  {
		m_puiTextures[i] = m_uiTexture6;
	  }
	  else if(strcmp(pMaterial->pszName, "mat_7") == 0)
	  {
		m_puiTextures[i] = m_uiTexture7;
	  }
	}
	return true;
}

/*!****************************************************************************
 @Function		ReleaseView
 @Return		bool		true if no error occured
 @Description	Code in ReleaseView() will be called by PVRShell when the
				application quits or before a change in the rendering context.
******************************************************************************/
bool OGLES2ComplexLighting::ReleaseView()
{
	// Delete textures
	glDeleteTextures(1, &m_uiTexture1);
	glDeleteTextures(1, &m_uiTexture2);
	glDeleteTextures(1, &m_uiTexture3);
	glDeleteTextures(1, &m_uiTexture4);
	glDeleteTextures(1, &m_uiTexture5);
	glDeleteTextures(1, &m_uiTexture6);
	glDeleteTextures(1, &m_uiTexture7);

	// Delete program and shader objects
	glDeleteProgram(m_ShaderProgram.uiId);

	glDeleteShader(m_uiVertShader);
	glDeleteShader(m_uiFragShader);

	// Delete buffer objects
	glDeleteBuffers(m_Scene.nNumMesh, m_puiVbo);
	glDeleteBuffers(m_Scene.nNumMesh, m_puiIndexVbo);

	return true;
```

```
}

/*!****************************************************************************
 @Function		RenderScene
 @Return		bool		true if no error occured
 @Description	Main rendering loop function of the program. The shell will
			call this function every frame.
			eglSwapBuffers() will be performed by PVRShell automatically.
			PVRShell will also manage important OS events.
			Will also manage relevent OS events. The user has access to
			these events through an abstraction layer provided by PVRShell.
******************************************************************************/
bool OGLES2ComplexLighting::RenderScene()
{
	glEnable(GL_BLEND);
	glEnable(GL_ALPHA);
	glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);

	// Clears the color and depth buffer
	glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

	m_Corners = [ACTrackingViewController currentCornerPoints];

	float width = 680.0;
	float height = 460.0;

	float vec1X, vec1Y, vec2X, vec2Y;
	float s, t;

	// calculating the line intersection of the diagonal corner points
	if (m_Corners.valid)
	{

	  bottomLeft = m_Corners.corner[FERNCornerBottomLeft];
	  bottomRight = m_Corners.corner[FERNCornerBottomRight];
	  topLeft = m_Corners.corner[FERNCornerTopLeft];
	  topRight = m_Corners.corner[FERNCornerTopRight];

	  float scale = width / m_Corners.imageSize.width;
	  float yOfset = ((m_Corners.imageSize.height - (height / scale)) / 2.0f);

	  bottomLeft.y += yOfset;
	  bottomRight.y += yOfset;
	  topLeft.y += yOfset;
	  topRight.y += yOfset;

	  vec1X = topRight.x - bottomLeft.x;
	  vec1Y = topRight.y - bottomLeft.y;
	  vec2X = bottomRight.x - topLeft.x;
	  vec2Y = bottomRight.y - topLeft.y;
	  //NSLog(@"vec1X value is: %f", vec1X);
	  //NSLog(@"vec1Y value is: %f", vec1Y);

	  s = (-vec1Y * (bottomLeft.x - topLeft.x) + vec1X * (bottomLeft.y - topLeft.y)) / (-vec2X *
vec1Y + vec1X * vec2Y);
	  t = ( vec2X * (bottomLeft.y - topLeft.y) - vec2Y * (bottomLeft.x - topLeft.x)) / (-vec2X *
vec1Y + vec1X * vec2Y);

	  if (s >= 0 && s <= 1 && t >= 0 && t <= 1)
	  {
		intersectPointX = (bottomLeft.x + (t * vec1X));
		intersectPointY = (bottomLeft.y + (t * vec1Y));
	  } else{
		intersectPointX = intersectPointY = 0.0;
	  }
	} else{
	  intersectPointX = intersectPointY = 0.0;
	}

	// Use shader program
	glUseProgram(m_ShaderProgram.uiId);
```

```
	glUniform1i(m_ShaderProgram.uiLightSelLoc, m_eLightType);

	float scale = 0.25;
	if (intersectPointX > 0.0 && intersectPointY > 0.0 && intersectPointX != intersectCheckX &&
intersectPointY != intersectCheckY){

	  m_fTranslate_x = (intersectPointX - width/2)*scale;
	  m_fTranslate_y = (intersectPointY - height/2)*scale;
	  m_fTranslate_z = 0.0;

	  intersectCheckX = intersectPointX;
	  intersectCheckY = intersectPointY;

	  NSLog(@"m_fTranslate_x value is: %f", m_fTranslate_x);
	  NSLog(@"m_fTranslate_y value is: %f", m_fTranslate_y);
	  NSLog(@"intersectPointX value is: %f", intersectPointX);
	  NSLog(@"intersectPointY value is: %f", intersectPointY);
	}

	m_fAngle_x = 15.0;
	m_fAngle_y = -140.0;

	PVRTMat4 mRotX, mRotY, mRotZ, mTrans, mScale, mModel;

	//1 degree = 0.0174532925 radian
	mRotX = PVRTMat4::RotationX((180+m_fAngle_x)*0.0174532925);
	mRotY = PVRTMat4::RotationY((180+m_fAngle_y)*0.0174532925);
	mRotZ = PVRTMat4::RotationZ((180+m_fAngle_z)*0.0174532925);
	mTrans = PVRTMat4::Translation(m_fTranslate_x, m_fTranslate_y, m_fTranslate_z);

	float xBorder = 8.0;
	float yBorder = 40.0;

	//localScaleFactor is lenght of the found Vec1 devided by lenght of known initial vector
lenght
	float localScaleFactor = ( sqrt(powf(vec1X, 2) + powf(vec1X, 2)) ) / sqrt( powf(xBorder-
(width-xBorder), 2) + powf((height-yBorder)-yBorder, 2) );

	//globalScaleFactor is overall scalefactor
	float globalScaleFactor = 11.0;

	//float localScaleFactor = ( sqrt(vec1X*vec1X + vec1Y*vec1Y)) / sqrt( (xBorder-(width-
xBorder))*(xBorder-(width-xBorder)) + ((height-yBorder)-yBorder)*((height-yBorder)-yBorder) );
	mScale = PVRTMat4::Scale(globalScaleFactor*localScaleFactor,
globalScaleFactor*localScaleFactor, globalScaleFactor*localScaleFactor);

	// Set model view projection matrix
	mModel = mRotX * mRotY * mRotZ * mTrans * mScale;
	PVRTMat4 mModelView, mMVP;
	mModelView = m_mView * mModel;
	mMVP = m_mProjection * mModelView;

	glUniformMatrix4fv(m_ShaderProgram.uiMVPMatrixLoc, 1, GL_FALSE, mMVP.ptr());

	for (unsigned int i32NodeIndex = 0; i32NodeIndex < m_Scene.nNumMeshNode; ++i32NodeIndex)
	{
	  SPODNode& Node = m_Scene.pNode[i32NodeIndex];

	  SPODNode* pNode2 = &m_Scene.pNode[i32NodeIndex];

	  LoadMaterial(pNode2->nIdxMaterial);

	  // Loads the correct texture using our texture lookup table
	  if(Node.nIdxMaterial == -1)
		glBindTexture(GL_TEXTURE_2D, 0); // It has no pMaterial defined. Use blank texture
(0)
	  else
		glBindTexture(GL_TEXTURE_2D, m_puiTextures[Node.nIdxMaterial]);

	  // Set model view matrix
	  glUniformMatrix4fv(m_ShaderProgram.uiModelViewLoc, 1, GL_FALSE, mModelView.ptr());
```

```
        // Set model view inverse transpose matrix
        PVRTMat3 mModelViewIT = PVRTMat3(mModelView).inverse().transpose();
        glUniformMatrix3fv(m_ShaderProgram.uiModelViewITLoc, 1, GL_FALSE, mModelViewIT.ptr());

        if (m_Corners.valid) DrawMesh(i32NodeIndex);
      }
      return true;
}

/*!****************************************************************************
 @Function            DrawMesh
 @Input               i32NodeIndex          Node index of the mesh to draw
 @Description  Draws a SPODMesh after the model view matrix has been set and
                      the meterial prepared.
******************************************************************************/
void OGLES2ComplexLighting::DrawMesh(int i32NodeIndex)
{

      int i32MeshIndex = m_Scene.pNode[i32NodeIndex].nIdx;
      SPODMesh* pMesh = &m_Scene.pMesh[i32MeshIndex];

      // bind the VBO for the mesh
      glBindBuffer(GL_ARRAY_BUFFER, m_puiVbo[i32MeshIndex]);
      // bind the index buffer, won't hurt if the handle is 0
      glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, m_puiIndexVbo[i32MeshIndex]);

      // Enable the vertex attribute arrays
      glEnableVertexAttribArray(VERTEX_ARRAY);
      glEnableVertexAttribArray(NORMAL_ARRAY);
      glEnableVertexAttribArray(TEXCOORD_ARRAY);

      // Set the vertex attribute offsets
      glVertexAttribPointer(VERTEX_ARRAY, 3, GL_FLOAT, GL_FALSE, pMesh->sVertex.nStride, pMesh-
>sVertex.pData);
      glVertexAttribPointer(NORMAL_ARRAY, 3, GL_FLOAT, GL_FALSE, pMesh->sNormals.nStride, pMesh-
>sNormals.pData);
      glVertexAttribPointer(TEXCOORD_ARRAY, 2, GL_FLOAT, GL_FALSE, pMesh->psUVW[0].nStride, pMesh-
>psUVW[0].pData);

      /*
        The geometry can be exported in 4 ways:
        - Indexed Triangle list
        - Non-Indexed Triangle list
        - Indexed Triangle strips
        - Non-Indexed Triangle strips
      */
      if(pMesh->nNumStrips == 0)
      {
        if(m_puiIndexVbo[i32MeshIndex])
        {
              // Indexed Triangle list
              glDrawElements(GL_TRIANGLES, pMesh->nNumFaces*3, GL_UNSIGNED_SHORT, 0);
        }
        else
        {
              // Non-Indexed Triangle list
              glDrawArrays(GL_TRIANGLES, 0, pMesh->nNumFaces*3);
        }
      }
      else
      {
        for(int i = 0; i < (int)pMesh->nNumStrips; ++i)
        {
              int offset = 0;
              if(m_puiIndexVbo[i32MeshIndex])
              {
                    // Indexed Triangle strips
                    glDrawElements(GL_TRIANGLE_STRIP, pMesh->pnStripLength[i]+2,
GL_UNSIGNED_SHORT, (GLshort*)(offset*2));
              }
              else
              {
```

```
                    // Non-Indexed Triangle strips
                    glDrawArrays(GL_TRIANGLE_STRIP, offset, pMesh->pnStripLength[i]+2);
                }
                offset += pMesh->pnStripLength[i]+2;
            }
        }

        // Safely disable the vertex attribute arrays
        glDisableVertexAttribArray(VERTEX_ARRAY);
        glDisableVertexAttribArray(NORMAL_ARRAY);
        glDisableVertexAttribArray(TEXCOORD_ARRAY);

        glBindBuffer(GL_ARRAY_BUFFER, 0);
        glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, 0);
}

/*!****************************************************************************
 @Function          NewDemo
 @Return            PVRShell*            The demo supplied by the user
 @Description  This function must be implemented by the user of the shell.
                    The user should return its PVRShell object defining the
                    behaviour of the application.
******************************************************************************/
PVRShell* NewDemo()
{
        return new OGLES2ComplexLighting();
}

/*****************************************************************************
 End of file (OGLES2ComplexLighting.cpp)
******************************************************************************/
```

The initTracker function

```
void initTracker()
{
      affine_transformation_range range;
      detector = planar_pattern_detector_builder::build_with_cache(model_image.c_str(),
                  &range,//affine_transformation_range* range,
                  200,   //(400)  int maximum_number_of_points_on_model
                  5000,  //(5000)   int number_of_generated_images_to_find_stable_points
                  0.0,   // (0.0)   double      minimum_number_of_views_rate
                  32,    // (32)    int patch_size
                  7,     // (7)     int yape_radius
                  4,     // (4)     int   number_of_octaves
                  15,    //  (30)   int   number_of_ferns
                  6,     // (12)    int number_of_tests_per_fern
                  5000,  // (10000)      int number_of_samples_for_refinement
                  200);  // (200)   int number_of_samples_for_test

  if (!detector) {
    cerr << "Unable to build detector.\n";
    return ;
  }

  detector->set_maximum_number_of_points_to_detect(500); // (1000)

  tracker = new template_matching_based_tracker();
  string trackerfn = model_image + string(".tracker_data");
  if (!tracker->load(trackerfn.c_str())) {
    cout << "Training template matching..."<<endl;
    tracker->learn(detector->model_image,
          5, // number of used matrices (coarse-to-fine)
          40, // max motion in pixel used to train to coarser matrix
          20, 20, // defines a grid. Each cell will have one tracked point.
```

```
            detector->u_corner[0], detector->v_corner[1],
            detector->u_corner[2], detector->v_corner[2],
            40, 40, // neighbordhood for local maxima selection
            5000 // 10000 number of training samples
            );
        tracker->save(trackerfn.c_str());
    }
    tracker->initialize();
}
```

The checkCornerPoints function

```
- (BOOL) checkCornerPoints:(FERNCornerPoints)corners
{
      CGPoint long1 = CGPointMake((corners.corner[0].x - corners.corner[1].x), (corners.corner[0].y
- corners.corner[1].y));
      CGPoint long2 = CGPointMake((corners.corner[3].x - corners.corner[2].x), (corners.corner[3].y
- corners.corner[2].y));

      CGPoint short1 = CGPointMake((corners.corner[3].x - corners.corner[0].x), (corners.corner[3].y
- corners.corner[0].y));
      CGPoint short2 = CGPointMake((corners.corner[2].x - corners.corner[1].x), (corners.corner[2].y
- corners.corner[1].y));

      const float tolerance1 = 70.0f;
      const float tolerance2 = 40.0f;

      BOOL shortAligned      = (fabs(angleBetween(short1, short2)) < tolerance1);
      BOOL longAligned= (fabs(angleBetween(long1, long2)) < tolerance2);

      BOOL perpendicular1     = (fabs(angleBetween(long1, short1)) > 30.0f) &&
(fabs(angleBetween(long1, short1)) < 150.0f);

      BOOL perpendicular2     = (fabs(angleBetween(long2, short2)) > 30.0f) && (fabs(angleBetween
      (long2, short2)) < 150.0f);

    float ratio = (length(long1) + length(long2)) / (length(short1) + length(short2));

    const float kRatio = 620.0f / 318.0f;

      const float kRatioThreshold = 0.08f;
      BOOL goodRatio = (ratio < kRatio + kRatioThreshold && ratio > kRatio - kRatioThreshold);

      return (longAligned && shortAligned && perpendicular1 && perpendicular2 && goodRatio);
}

float angleBetween(CGPoint vector1, CGPoint vector2)
{
      return (float) (atan2( vector1.x * vector2.y - vector1.y * vector2.x, vector1.x * vector2.x +
vector1.y * vector2.y ) * (180.0f/M_PI));
}

float length(CGPoint vector01)
{
    return sqrt(vector01.x * vector01.x + vector01.y * vector01.y);
}
```

Appendix C

Regenbrecht and Schubert's original questions

1) Was watching the virtual objects just as natural as watching the real world?

2) Did you have the impression that the virtual objects belonged to the real object (dinosaur skull), or did they seem separate from it?

3) Did you have the impression that you could have touched and grasped the virtual objects?

4) Did the virtual objects appear to be (visualized) on a screen, or did you have the impression that they were located in space?

5) Did you have the impression of seeing the virtual objects as merely flat images or as three-dimensional objects?

6) Did you pay attention at all to the difference between real and virtual objects?

7) Did you have to make an effort to recognize the virtual objects as being three-dimensional?

[Regenbrecht & Schubert 2002]

Test questionnaire translated into German

Halten Sie sich selbst für einen erfahrenen Smartphone Benutzer?

Überhaupt Keine Zustimmung	Keine Zustimmung	Eher keine Zustimmung	Etwas Zustimmung	Zustimmung	Völlige Zustimmu

Wissen Sie, was "Augemented Reality" ist?

Überhaupt Keine Zustimmung	Keine Zustimmung	Eher keine Zustimmung	Etwas Zustimmung	Zustimmung	Völlige Zustimmu

Falls Sie zustimmen – Bitte begründen Sie:

War das Auto zu betrachten so real, wie ein Blick in die reale Welt?

Überhaupt Keine Zustimmung	Keine Zustimmung	Eher keine Zustimmung	Etwas Zustimmung	Zustimmung	Völlige Zustimmu

Erschien es so, als würde das Auto zu dem Poster gehören?

Überhaupt Keine Zustimmung	Keine Zustimmung	Eher keine Zustimmung	Etwas Zustimmung	Zustimmung	Völlige Zustimmu

Wurde der Eindruck erweckt, man könnte das Auto berühren?

Überhaupt Keine Zustimmung	Keine Zustimmung	Eher keine Zustimmung	Etwas Zustimmung	Zustimmung	Völlige Zustimmu

Wirkte das Auto wie auf einem Bildschirm?

Überhaupt Keine Zustimmung	Keine Zustimmung	Eher keine Zustimmung	Etwas Zustimmung	Zustimmung	Völlige Zustimmu

Hatten Sie den Eindruck, als ob das Auto im Raum stand?

Überhaupt Keine Zustimmung	Keine Zustimmung	Eher keine Zustimmung	Etwas Zustimmung	Zustimmung	Völlige Zustimmu

Hatten sie den Eindruck, dass das Auto zweidimensional war?

Überhaupt Keine Zustimmung	Keine Zustimmung	Eher keine Zustimmung	Etwas Zustimmung	Zustimmung	Völlige Zustimmu

Waren Unterschiede zwischen Poster und Auto zu erkennen?

Überhaupt Keine Zustimmung	Keine Zustimmung	Eher keine Zustimmung	Etwas Zustimmung	Zustimmung	Völlige Zustimmu

Erschien es so, als ob das Auto ein dreidimensionales Objekt war?

Überhaupt Keine Zustimmung	Keine Zustimmung	Eher keine Zustimmung	Etwas Zustimmung	Zustimmung	Völlige Zustimmu

War das Bild zu betrachten so real, wie ein Blick in die reale Welt?

Überhaupt Keine Zustimmung	Keine Zustimmung	Eher keine Zustimmung	Etwas Zustimmung	Zustimmung	Völlige Zustimmu

Hatten Sie den Eindruck, dass das Bild im Poster zweidimensional war?

Überhaupt Keine Zustimmung	Keine Zustimmung	Eher keine Zustimmung	Etwas Zustimmung	Zustimmung	Völlige Zustimmu

Haben Sie weitere Kommentare?

Geschlecht (Mann/Frau): Alter:

Test questionnaire in English

Do you consider yourself an experienced smartphone user?

Strongly disagree	Disagree	Disagree somewhat	Agree somewhat	Agree	Strongly Agree

Do you know what Augmented Reality is?

Strongly disagree	Disagree	Disagree somewhat	Agree somewhat	Agree	Strongly Agree

If Agree – please explain how:

Was watching the car just as natural as watching the real world?

Strongly disagree	Disagree	Disagree somewhat	Agree somewhat	Agree	Strongly Agree

Did you have the impression that the car belonged to the poster?

Strongly disagree	Disagree	Disagree somewhat	Agree somewhat	Agree	Strongly Agree

Did you have the impression that you could have touched and grasped the car?

Strongly disagree	Disagree	Disagree somewhat	Agree somewhat	Agree	Strongly Agree

Strongly disagree	Disagree	Disagree somewhat	Agree somewhat	Agree	Strongly Agree

Did the car appear to be (visualized) on a screen?

Did you have the impression that the car was located in space?

Strongly disagree	Disagree	Disagree somewhat	Agree somewhat	Agree	Strongly Agree

Did you have the impression of seeing the car as a flat image?

Strongly disagree	Disagree	Disagree somewhat	Agree somewhat	Agree	Strongly Agree

Did you pay attention at all to the difference between the poster and car?

Strongly disagree	Disagree	Disagree somewhat	Agree somewhat	Agree	Strongly Agree

Did you see the car as a 3-dimensional object?

Strongly disagree	Disagree	Disagree somewhat	Agree somewhat	Agree	Strongly Agree

Was watching the poster just as natural as watching the real world?

Strongly disagree	Disagree	Disagree somewhat	Agree somewhat	Agree	Strongly Agree

Did you have the impression of seeing the poster space as a three-dimensional space?

Strongly disagree	Disagree	Disagree somewhat	Agree somewhat	Agree	Strongly Agree

Do you have any general comments?

____

____

____

____

Gender (Male/Female): Age:

Appendix D

General info about test subjects

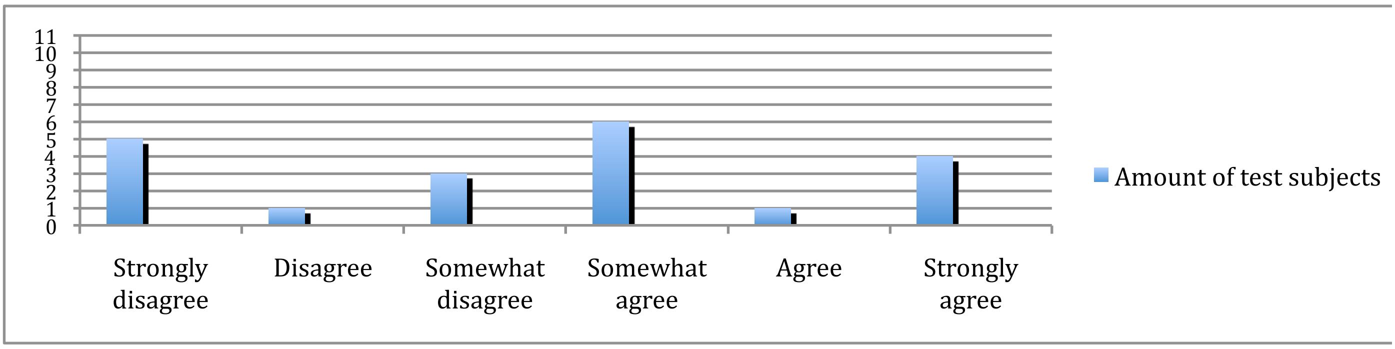
Test subjects according to Gender: 11 males, 9 females.

Average age: (23+23+25+24+28+25+32+24+26+32+25+40+26+41+29+29+20+36+29+30) / 20 = 28,35 years.

Bar charts of the test results

Question a: *Do you consider yourself an experienced smartphone user?*

Result:

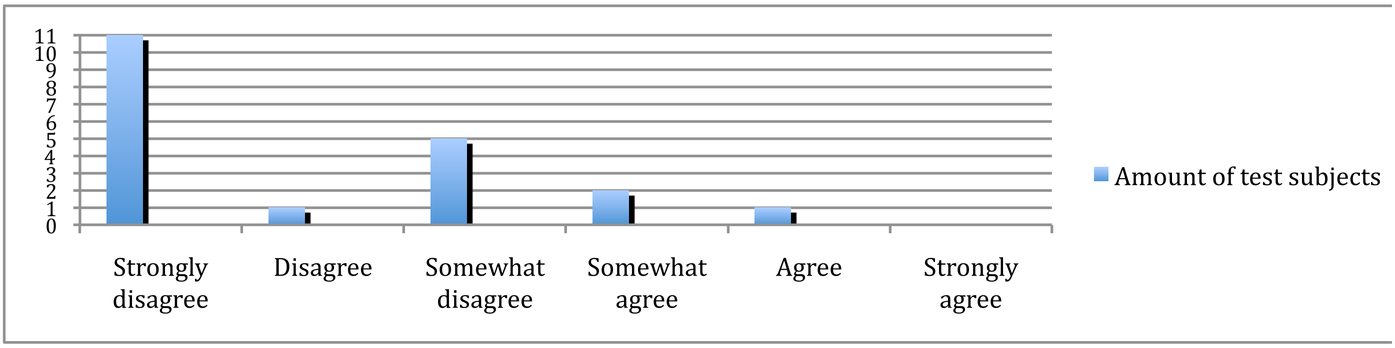


Mean Value (assuming *Strongly agree* = 1, *Disagree* = 2, *Somewhat disagree* = 3, etc.): (5*1+1*2+3*3+6*4+1*5+4*6) / 20 = 3.45, which points towards a mean lying around halfway between *Somewhat disagree* and *Somewhat agree*.
Standard Deviation: $\sqrt{(((1-3.45)^2 *5 + (2-3.45)^2 *1 + (3-3.45)^2*3 + (4-3.45)^2*6 + (5-3.45)^2*1 + (6-3.45)^2*4)) / (20-1))} = 1.82021$

The Mean Values for the result of the other test questions, as well as the Standard Deviations, has been calculated in the same manner as described above.

Question b: *Do you know what Augmented Reality is?*

Result:



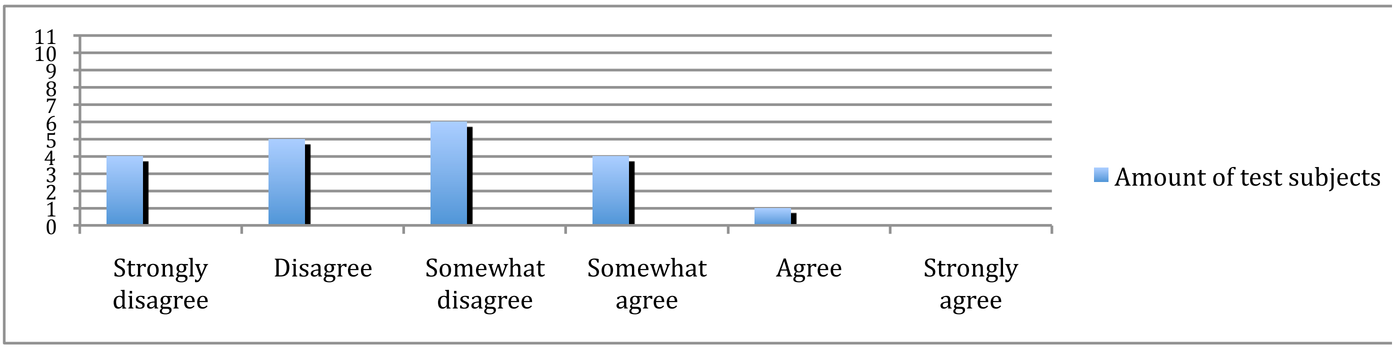
Mean Value: 2.05, which points towards *Disagree*.
Standard Deviation: 1.31689

Question c: *If agree – please explain how:*

> *I have seen it in a film on Facebook.*
> *I heard the term before – but could not explain what it means in details.*
> *My boyfriend told me about it.*

Question 1: *Was watching the car just as natural as watching the real world?*

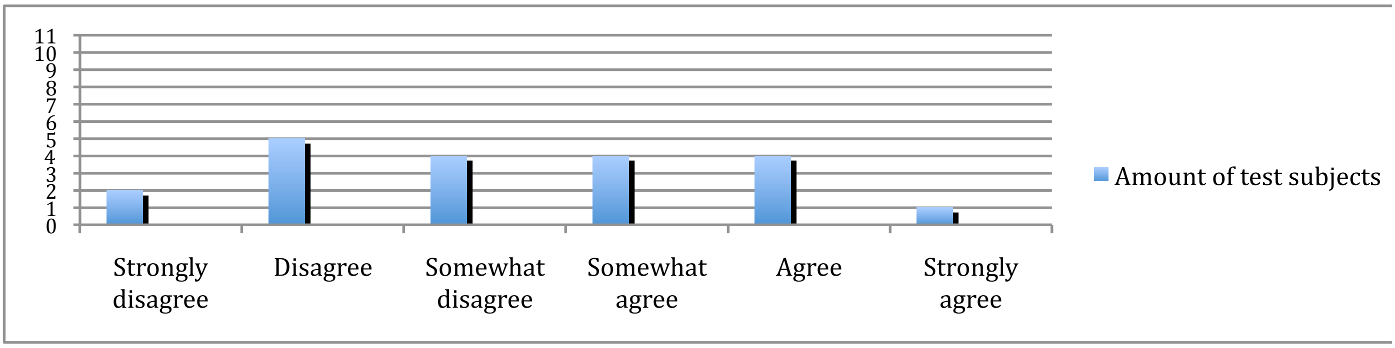
Result:



Mean Value: 2.65, which points towards halfway between *Disagree* and *Somewhat disagree*.
Standard Deviation: 1.1821

Question 2: *Did you have the impression that the car belonged to the poster?*

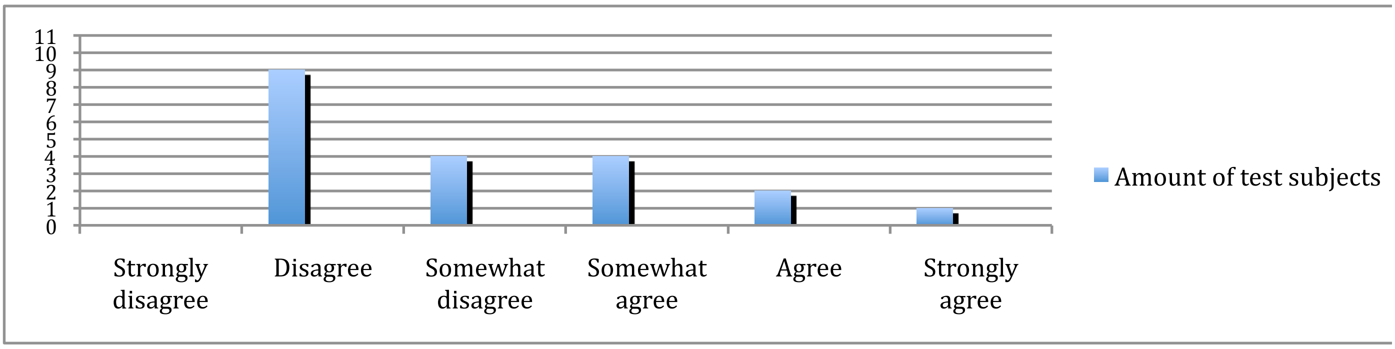
Result:



Mean Value: 3.3, which points towards halfway between *Somewhat disagree* and *Somewhat agree*.
Standard Deviation: 1.45458

Question 3: *Did you have the impression that you could have touched and grasped the car?*

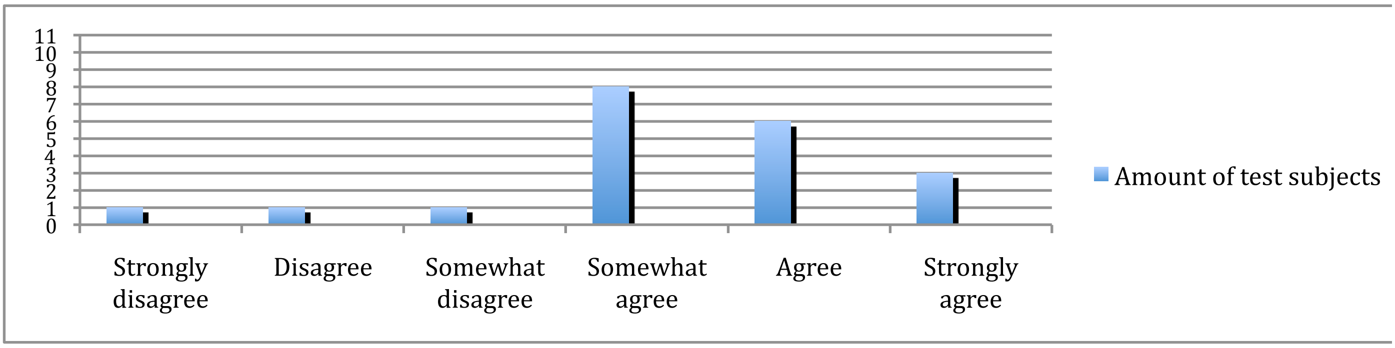
Result:



Mean Value: 3.1, which points towards *Somewhat disagree*.
Standard Deviation: 1.25237

Question 4: *Did the car appear to be (visualized) on a screen?*

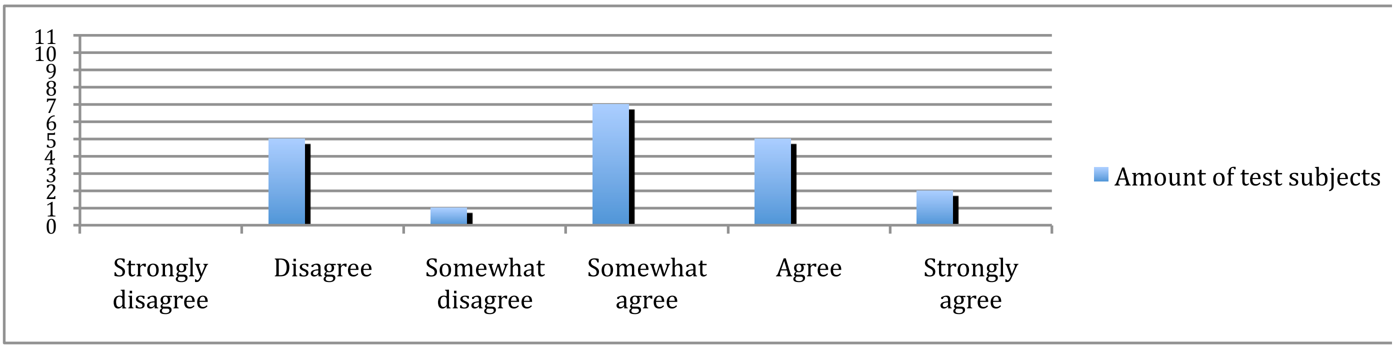
Result:



Mean Value: 4.3, which points towards *Somewhat agree*.
Standard Deviation: 1.26074

Question 5: *Did you have the impression that the car was located in space?*

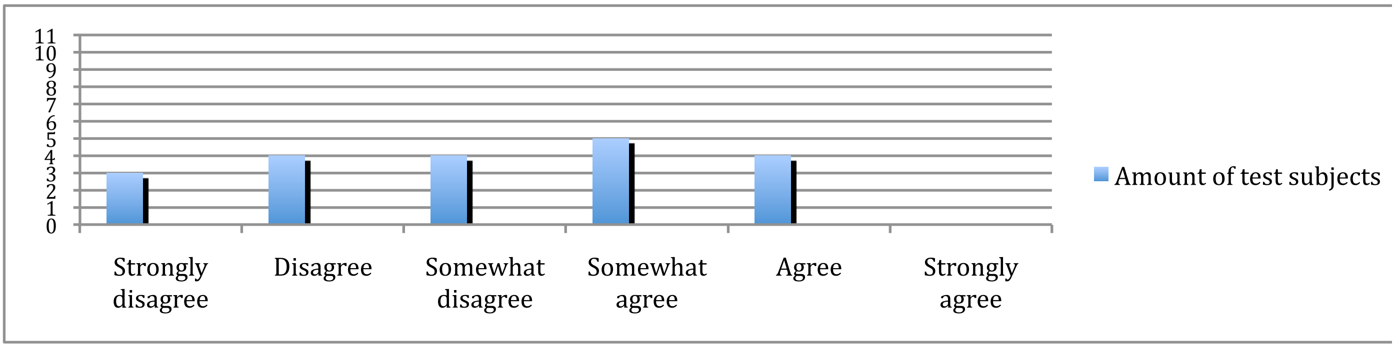
Result:



Mean Value: 3.9, which points towards *Somewhat agree*.
Standard Deviation: 1.33377

Question 6: *Did you have the impression of seeing the car as a flat image?*

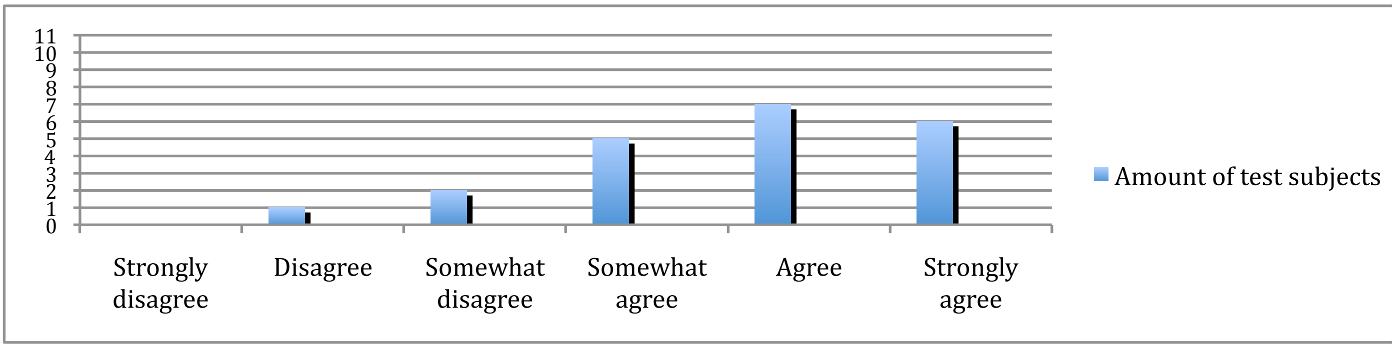
Result:



Mean Value: 3.15, which points towards *Somewhat disagree*.
Standard Deviation: 1.38697

Question 7: *Did you pay attention at all to the difference between the poster and car?*

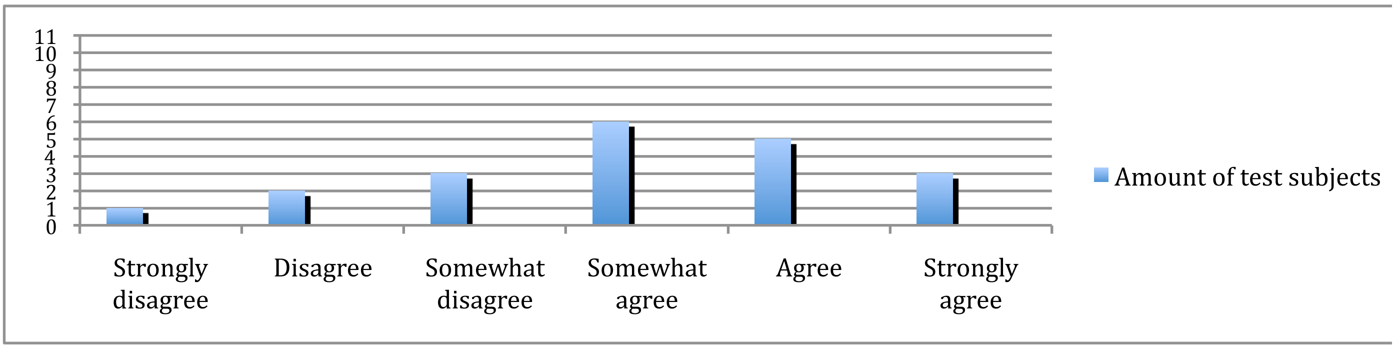
Result:



Mean Value: 4.85, which points towards *Agree*.
Standard Deviation: 0.98809

Question 8: *Did you see the car as a 3-dimensional object?*

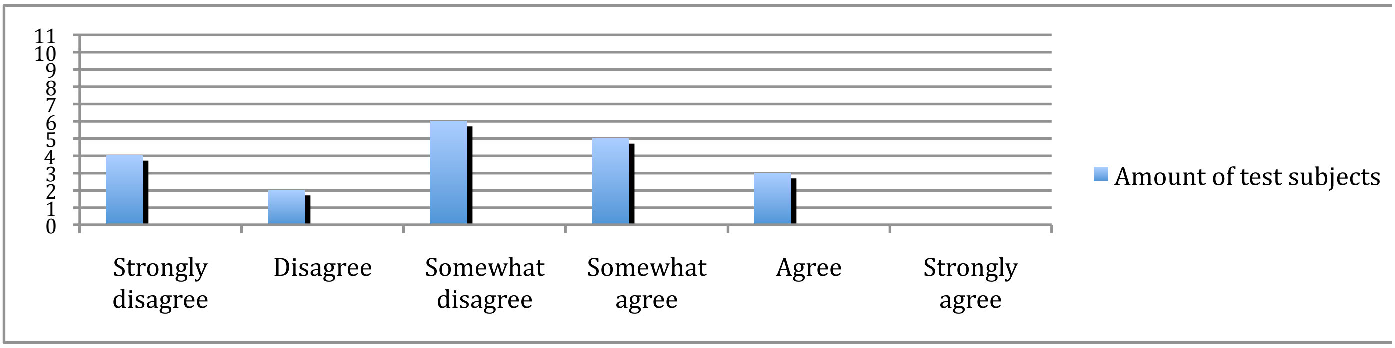
Result:



Mean Value: 4.05, which points towards *Somewhat agree*.
Standard Deviation: 1.39454

Question 9: *Was watching the poster just as natural as watching the real world?*

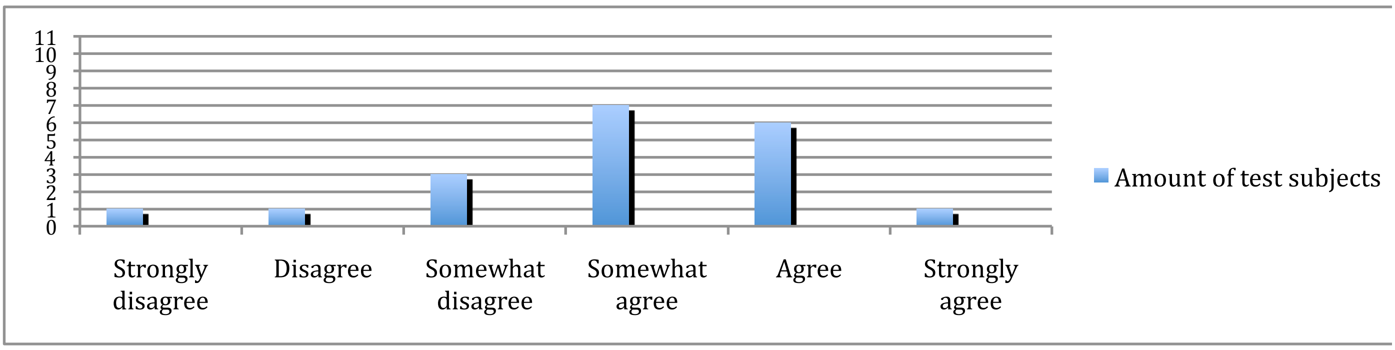
Result:



Mean Value: 3.05, which points towards *Somewhat disagree*.
Standard Deviation: 1.35627

Question 10: *Did you have the impression of seeing the poster space as a three-dimensional space?*

Result:



Mean Value: 3.9, which points towards *Somewhat agree*.

Standard Deviation: 1.25237

Question 11: *Do you have any general comments?*

Only one of the test subjects added a general comment: *The car was a bit unclear, and it could have had more colours and contours. Furthermore it could also be able to move.*