



Nutzung von mobilen Augmented Reality Technologien in der Architekturvisualisierung

Zweite Bachelorarbeit

eingereicht von

Sebastian Steiner
0910261100

im Rahmen des
Studiengangs Medientechnik an der Fachhochschule St. Pölten

Betreuung
Jakob Doppler, MSc

St. Pölten,
04.06.2012

(Unterschrift Autor/Autorin)

(Unterschrift Betreuer/Betreuerin)



Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Bachelorarbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Bachelorarbeitsthema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.
- diese Arbeit mit der vom Begutachter/von der Begutachterin beurteilten Arbeit übereinstimmt.
- Ich räume hiermit der Fachhochschule St. Pölten das ausschließliche und räumlich unbeschränkte Werknutzungsrecht für alle Nutzungsarten an dieser Bachelorarbeit ein, und behalte das Recht, als Urheber dieses Werkes genannt zu werden.

St. Pölten,
04.06.2012

(Unterschrift Autor/Autorin)



Zusammenfassung

Diese Arbeit befasst sich mit der Verwendung mobiler Augmented Reality (AR) Applikationen zum Zweck der Architekturvisualisierung. Augmented Reality soll dem Menschen ermöglichen seine wahrgenommene Realität durch virtuelle Inhalte und Informationen zu erweitern. Durch die steigende Leistungsfähigkeit mobiler Prozessoren und der Integration von Positionssensoren und Kameras in Smartphones und Tablets, ist es möglich Augmented Reality Technologie in mobilen Applikationen zur Darstellung komplexer 3D-Modelle einzusetzen. Da in der Architekturvisualisierung häufig Objekte in ihrem räumlichen Kontext dargestellt werden müssen bietet sich der Einsatz mobiler AR-Lösungen an. In dieser Arbeit wird dargelegt ob Augmented Reality auf mobilen Endgeräten sinnvoll zur Darstellung von 3D-Objekten aus dem Architekturbereich eingesetzt werden kann. Nach einer Einführung zum Thema Augmented Reality wird der Stand der Technik mobiler AR-Applikationen dargestellt. Für den Anwendungsfall Architekturvisualisierung werden in weiterer Folge Beweggründe für den Einsatz von AR-Technologie, sowie potentielle Problemfelder aufgeführt. Auf Basis dieser Informationen folgt eine Auflistung konkreter Anforderungen an Augmented Reality Anwendungen, wobei die Standpunkte von Anwendern, Entwicklern und Inhaltslieferanten berücksichtigt werden. Die gesammelten Anforderungen erlauben im weiteren Verlauf der Arbeit eine Evaluierung unterschiedlicher Umsetzungsmöglichkeiten. Um den Einsatz mobiler AR-Technologie zur Architekturvisualisierung umfassend bewerten zu können wird schlussendlich auf den konkreten Entwicklungsprozess einer entsprechenden Prototyp Applikation näher eingegangen.



Inhalt

1. EINLEITUNG	5
2. GRUNDLAGEN AUGMENTED REALITY	7
2.1. BEGRIFFSDEFINITIONEN	7
2.2. AR – TECHNISCHE UND FUNKTIONALE GRUNDLAGEN.....	8
2.3. MOBILE AUGMENTED REALITY – STAND DER TECHNIK.....	10
3. AR ANWENDUNGSFALL - ARCHITEKTURVISUALISIERUNG	16
3.1. EINLEITUNG.....	16
3.2. BEWEGGRÜNDE FÜR DEN EINSATZ MOBILER AR TECHNOLOGIEN	17
3.3. MÖGLICHE PROBLEMFELDER.....	19
3.4. ANWENDUNGSBEISPIELE	20
4. ANFORDERUNGSBASIERTE EVALUIERUNG DER UMSETZUNG VON AR SYSTEMEN ZUR ARCHITEKTURVISUALISIERUNG	22
4.1. EVALUIERUNGSMETHODIK	22
4.2. ANFORDERUNGEN AN AR-APPLIKATIONEN	22
4.3. UMSETZUNGSMÖGLICHKEITEN.....	25
4.4. EVALUIERUNGSERGEBNISSE	27
<i>Sicht der Anwender</i>	27
<i>Sicht der Entwickler</i>	29
<i>Sicht der Content Provider</i>	31
4.5. FAZIT	32
5. IMPLEMENTIERUNG EINER ARCHITEKTURVISUALISIERUNGS-APPLIKATION	34
5.1. EINLEITUNG.....	34
5.2. TECHNISCHE UMSETZUNG.....	34
5.3. VORSTELLUNG DER FUNKTIONALITÄTEN.....	40
5.4. ERKENNTNISSE AUS DER ANWENDUNGSENTWICKLUNG	42
6. FAZIT	44
LITERATURVERZEICHNIS	46
ABBILDUNGSVERZEICHNIS	48
TABELLENVERZEICHNIS	49

1. Einleitung

Bereits vor ungefähr 25 Jahren begann die Forschungsarbeit im Bereich der erweiterten Realität, auch Augmented Reality (AR) genannt (vgl. Wagner 2007, S.III). Versucht wird dabei eine Erweiterung der realen Umgebung durch virtuelle Informationen und Objekte zu erreichen. Ein konkretes Ziel vieler Arbeiten war neue Mensch-Maschinen Schnittstellen für verschiedene Anwendungsfälle zu entwickeln. Als mögliche Einsatzbereiche wurden etwa komplexe Fertigungs- und Wartungsarbeiten sowie militärische Systeme betrachtet. Der Einsatz von Augmented Reality Technologie für private Anwender wurde für relativ lange Zeit nicht beachtet. So wurden Themen wie Usability und sozialer Akzeptanz wenig Interesse geschenkt. Auf Abbildung 1 ist eine der frühen Entwicklungen eines head-worn display (HWD) zu erkennen.



Abbildung 1: HWD Prototyp (Biocca, Rolland o.J., [<http://mindlab.msu.edu/presence/presence.html>])

Mit der Einführung immer leistungsstärkerer mobiler Geräte, angefangen bei Apple's iPhone im Jahr 2007, war auch für private Nutzer die Möglichkeit vorhanden, AR Technologie zu verwenden. Durch die Integration einer Kamera und diverser Sensoren zur Positionsbestimmung waren auch die Voraussetzungen zur Entwicklung erster mobiler Augmented Reality Applikationen für Endanwender geschaffen. In den folgenden Jahren wurden AR-Visualisierungen auf mobilen Geräten hauptsächlich zur Produktpräsentation und von Spielen eingesetzt. Mit der Entwicklung mobiler Mehrkernprozessoren, der Integration schneller Grafichips und der Verwendung HD-fähiger Kamerasensoren konnten die 3D-Darstellungsleistung und die Erkennung visueller Marker verbessert werden. Mobile AR-Applikationen sind dadurch heutzutage in der Lage komplexe 3D-Visualisierungen zu realisieren. Aufgrund dieser Voraussetzungen ist auch der Einsatz von Augmented Reality in der Architekturvisualisierung denkbar geworden.

Im Rahmen dieser Arbeit wird dargelegt inwieweit Augmented Reality auf mobilen Endgeräten eine sinnvolle Möglichkeit zur Visualisierung von 3D-Objekten aus der Architektur sein kann. Zu diesem Zweck wird der derzeitige Stand mobiler AR-Applikationen erhoben sowie die Beweggründe für den Einsatz dieser Technologie im Bereich der Architekturvisualisierung. In Verbindung mit der Aufstellung möglicher Problemfelder in diesem Gebiet werden konkrete Anforderungen an Augmented Reality Anwendungen im Architekturbereich abgeleitet. Dies ermöglicht im weiteren Verlauf der Arbeit eine Evaluierung entsprechender Umsetzungsmöglichkeiten. Durch die Beschreibung des Entwicklungsverlaufs eines mobilen Augmented Reality Prototyps, werden schlussendlich auch die praktischen Erfahrungen aus der Umsetzung eines solchen Systems präsentiert.

2. Grundlagen Augmented Reality

2.1. Begriffsdefinitionen

In den folgenden Kapiteln werden einige Begriffe aus dem Themengebiet Augmented Reality verwendet. Da für einige dieser Fachwörter unterschiedliche Definitionen existieren, soll hier die exakte Bedeutung dieser Begrifflichkeiten für den Rahmen dieser Arbeit definiert werden.

- **Augmented Reality (AR):** Wird gleichbedeutend mit dem Begriff der erweiterten Realität verwendet. Augmented Reality beschreibt die Erweiterung der realen Welt um virtuelle (computer-generierte) Objekte welche eingebettet in den realen Raum dargestellt werden. (vgl. Azuma et al. 2001, S.34)
- **Virtual Reality (VR):** Wird auch virtuelle Realität genannt. Durch die Simulation physischer Anwesenheit in einer interaktiven virtuellen Umgebung wird eine Form von Realität geschaffen.
- **Augmented Reality System:** Ein AR System dient zur Realisierung einer erweiterten Realität. Dazu benötigt es drei Eigenschaften. Es muss reale und virtuelle Objekte in einer realen Umgebung kombinieren können. Darüber hinaus muss es auch in Echtzeit Interaktivität und die Manipulation von virtuellen Objekten unterstützen. Außerdem muss es auch in der Lage sein virtuelle und reale Objekte zu kombinieren. (vgl. Azuma et al. 2001, S.34)
- **Marker:** Marker helfen in einem AR System virtuelle Objekte in der realen Umgebung zu platzieren – siehe Abbildung 2. Dabei sind verschiedene Ausprägungen möglich. Zwei- und dreidimensionale Objekte können als Marker verwendet werden, solange dies vom AR System unterstützt wird. Als Ankerpunkte für die Berechnung der räumlichen Positionierung sind Marker ein wichtiger Bestandteil von AR Systemen.



Abbildung 2: Mehrere Marker in einer AR Anwendung (Hincapié 2011, S.1)

2.2. AR – Technische und funktionale Grundlagen

Im Bereich der erweiterten Realität gibt es sehr viele unterschiedliche konkrete Implementierungen. So breit gefächert die möglichen Einsatzbereiche sind, sind auch die technischen Umsetzungen. Je nach Einsatzbereich und benötigter Funktionalität können unterschiedliche AR Teilbereiche von hoher Wichtigkeit sein. Zur grundsätzlichen Realisierung eines AR Systems sind aber einige grundlegende Technologien unbedingt notwendig. Azuma nennt hier etwa Displays, Tracking, Registrierung und Kalibrierung (vgl. Azuma et al. 2001, S.34f.)

Displays sind notwendig um die Kombination aus realer und virtueller Umgebung darzustellen. Häufig werden head-worn displays (HWD) und handheld displays eingesetzt um dem Benutzer die Möglichkeit zu geben seine Umgebung in möglichst gewohnter Weise wahrzunehmen. Handheld displays sind mittlerweile durch die Popularität von Smartphones und Tablets recht stark verbreitet. Ein aktuelles Beispiel für ein HWD dagegen liefert Google's Project Glass, ein AR Forschungsprojekt in dem eine Brille mit Display verwendet wird um virtuelle Inhalte darzustellen, siehe Abbildung 3.

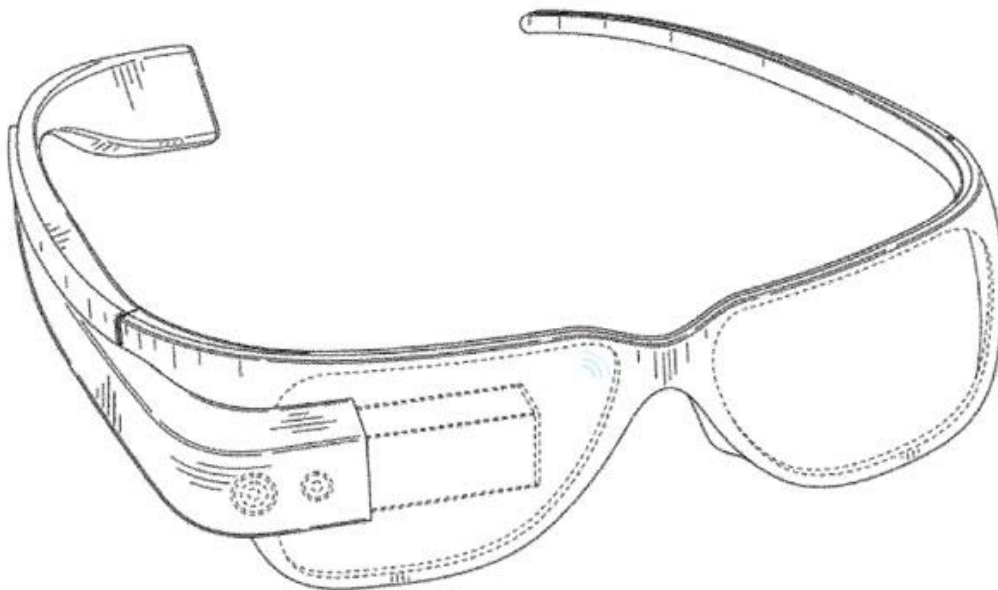


Abbildung 3: Google Project Glass (Sakr 2012, [<http://www.engadget.com/2012/05/15/google-glasses-design-patent/>])

Unter Tracking versteht man die konstante Erfassung der Blickrichtung und Position des Benutzers. Um eine möglichst hohe Interaktionsqualität und -robustheit zu erreichen, können mehrere verschiedene Tracking Technologien kombiniert werden. Die Triangulation von Sensormessdaten und visuellem Tracking wäre ein konkreter Anwendungsfall. Geringe Latenzzeiten und hohe Genauigkeit im Tracking sind auch eine wichtige Voraussetzung für die Registrierung der virtuellen Inhalte im realen Umfeld. Um wirklich den Eindruck einer erweiterten Realität entstehen lassen zu können muss die Verschmelzung von virtuellen und realen Objekten



eine besonders hohe Qualität aufweisen. Die Registrierung ist hier ein maßgeblicher Faktor. Damit diese zufriedenstellend funktionieren kann, müssen die visuellen Marker für das AR System möglichst gut erkennbar sein.

Wie Azuma (2001, S.37) beschreibt, ist Kalibrierung ein wichtiger Punkt für akkurate Registrierungsergebnisse. Es können unter anderem Kamera Parameter, Sensor Schwellwerte und Verzerrungen gemessen werden. Durch moderne Visualisierungserweiterungen kann aber auch schon häufig auf eine manuelle Kalibrierung verzichtet werden. Gerade bei mobilen Anwendungen ist durch die Integration von Kamera und Sensorik in die Geräte, üblicherweise die Kalibrierung in der Software des AR Systems automatisiert.

Heutzutage sind für Endanwender hauptsächlich mobile AR Applikationen interessant und auch zugänglich. Diese Apps sind speziell auf bestimmte mobile Betriebssysteme und Geräte optimiert. Durch den stetigen technologischen Fortschritt bei handheld devices eignen sich moderne Smartphones und Tablet-Computer immer besser als AR Systeme. Folgende Technologien sind von besonderer Bedeutung:

- **Prozessor:** Die Darstellung und Transformation von 3D-Objekten sowie das Tracking der Geräteposition und des Sichtfelds stellen rechenintensive Prozesse dar. Moderne Smartphones und Tablets mit Mehrkernprozessoren und eigenen Grafikprozessoren sind daher besonders gut zur Verwendung als AR System geeignet.
- **Kamera:** Die Kamera eines mobilen Gerätes muss sämtliche visuellen Informationen über die reale Umgebung liefern die das AR System benötigt. Durch viele Verbesserungen an Optiken und Chips sind diese mittlerweile in der Lage Videos in HD Qualität aufzuzeichnen, sowie Bilder mit mehreren Megapixeln Auflösung aufzunehmen. Durch die räumlich extrem limitierte Bauweise sind die Chip-Größen im Vergleich zu Kompakt-Digitalkameras oder gar Spiegelreflexkameras sehr gering, was zu höherem Bildrauschen vor allem bei mäßigen bis schlechten Lichtverhältnissen führt. Die Qualität des Live-Bildes hängt natürlich auch mit der Belastbarkeit der mobilen CPU bzw. GPU zusammen. Eine hohe Genauigkeit an visuellen Daten muss aber angestrebt werden um eine möglichst genaue Registrierung der virtuellen Inhalte zu ermöglichen.
- **Sensorik:** Je nach AR-System werden unterschiedlich genaue Positionsdaten des Gerätes im Raum benötigt. Durch GPS, WLAN-Netze und Funkzellen kann eine geografische Position bestimmt werden. Accelerometer, Gyroskope und Neigungssensoren dienen dazu das aktuelle Blickfeld zu berechnen.
- **Touchscreen:** Der Touchscreen dient primär zur Anzeige der erweiterten Realität. Zusätzlich kann er benutzt werden für verschiedene Interaktionspatterns wie (Multitouch)-Gesten. Die Anzeige und gleichzeitige Manipulierbarkeit der virtuellen Inhalte kann für eine sehr direkte Interaktion verwendet werden, was den Effekt der erweiterten Realität weiter verstärken kann.

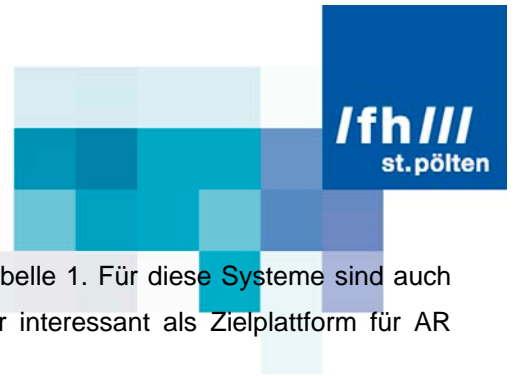
- **Mobile Datenkommunikation:** Der mobile Zugang zum Internet ist zwar keine Notwendigkeit für ein AR System, bietet aber die Möglichkeit für sehr viele konkrete Anwendungsszenarien. Die virtuellen Inhalte sind somit nicht mehr statisch an das System gebunden sondern können durch dynamisch geladene Daten ergänzt werden. Die in Österreich entwickelte AR Applikation Wikitude ist ein prominentes Beispiel für ein AR System welches sich sehr stark auf die Anzeige dynamisch geladener Inhalte spezialisiert. Je nach Geo-Position des mobilen Gerätes werden Daten von lokalen Content-Providern geladen die durch den Wikitude World Browser über der realen Umgebung angezeigt werden können (vgl. Wikitude GmbH o.J.a, [<http://www.wikitude.com/tour/wikitude-world-browser>]). Abbildung 4 zeigt ein Beispiel einer solchen Überlagerung mit den Positionsdaten der 50 nächstgelegenen Geldautomaten.



Abbildung 4: Wikitude Screenshot

2.3. Mobile Augmented Reality – Stand der Technik

Mobile Augmented Reality Systeme sind mittlerweile in Form von nativen mobilen Applikationen relativ weit verbreitet. Die Systeme sind also speziell für einzelne mobile Betriebssysteme implementiert und auf die jeweils zur Verfügung stehenden Libraries und APIs optimiert. Im App Store und auf Google Play, den App-Marktplattformen der mobilen Betriebssysteme iOS und Android, finden sich mit Stand Mai 2012 Augmented Reality Applikationen in vielen unterschiedlichen App-Kategorien. AR Technologien werden hier in unterschiedlichen Bereichen eingesetzt. Sie dienen zur Navigation, finden Verwendung in Spielen und unterstützen Marketing Kampagnen. Im Rahmen dieser Arbeit wurden speziell Applikationen für die bereits erwähnten Betriebssysteme iOS und Android analysiert. Im ersten Quartal 2012 halten sie im Smartphone



Bereich zusammen einen Marktanteil von 82% weltweit, siehe auch Tabelle 1. Für diese Systeme sind auch die derzeit leistungsfähigsten mobilen Geräte verfügbar, was sie sehr interessant als Zielplattform für AR Entwickler macht.

Top Six Smartphone Operating Systems, Shipments, and Market Share, 2012 Q1 (Units in Millions)

Mobile Operating System	1Q12 Unit Shipments	1Q12 Market Share	1Q11 Unit Shipments	1Q11 Market Share	Year-over-Year Change
Android	89.9	59.0%	36.7	36.1%	145.0%
iOS	35.1	23.0%	18.6	18.3%	88.7%
Symbian	10.4	6.8%	26.4	26.0%	-60.6%
BlackBerry OS	9.7	6.4%	13.8	13.6%	-29.7%
Linux	3.5	2.3%	3.2	3.1%	9.4%
Windows Phone 7/Windows Mobile	3.3	2.2%	2.6	2.6%	26.9%
Other	0.4	0.3%	0.3	0.3%	33.3%
Total	152.3	100.0%	101.6	100.0%	49.9%

Tabelle 1: Marktanteile mobile Betriebssysteme 1. Quartal 2012 (IDC Corporate USA 2012, [<http://www.idc.com/getdoc.jsp?containerId=prUS23503312>])

Die derzeit auf Android und iOS verfügbaren mobilen AR Applikationen lassen sich in zwei grundsätzlich unterschiedliche Kategorien einordnen. Die erste Kategorie sind Apps die AR Technologien einsetzen um Information verschiedenster Art geografisch zu verorten. Ein typischer Anwendungsfall ist die Anzeige von POIs (Points Of Interest) in einer gewissen Umgebung. Klassischerweise verwenden mobile Applikationen Kartensysteme und einfache Listendarstellungen um derartige Informationen anzuzeigen. AR Applikationen bieten nun zusätzlich die Möglichkeit diese Punkte überlagert über das Live-Kamerabild zu sehen. Diese Darstellung soll den Benutzern eine möglichst direkte Sicht auf die interessanten Punkte in ihrer Umgebung liefern. Es existieren in dieser Kategorie Applikationen die zur Anzeige für ganz spezielle Informationen wie Satellitenpositionen, Golfplatzdaten oder auch Berggipfel entwickelt wurden. Ein Beispiel hierfür stellt die App Peak.AR von Salzburg Research dar. Die Anwendung stellt Berggipfel und deren Höhen in einer einstellbaren Umgebung an ihrer geografischen Position dar, siehe Abbildung 5.

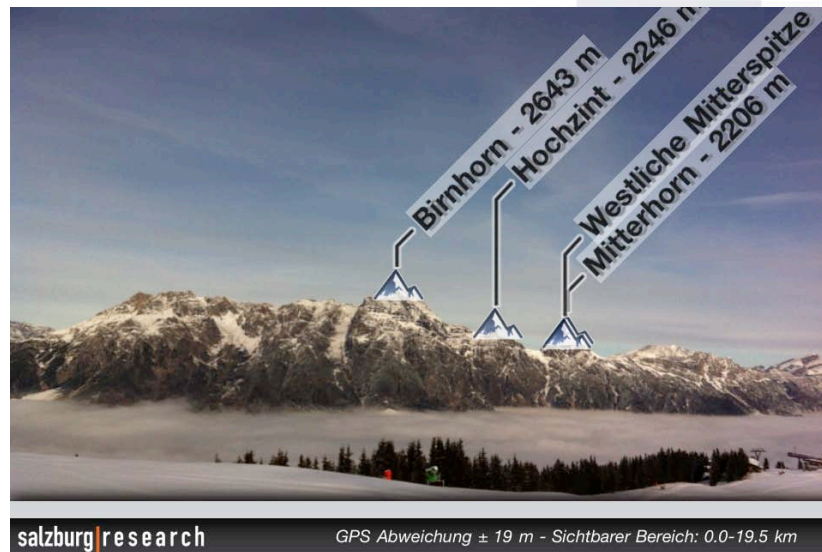


Abbildung 5: Peak.AR Überlagerung von virtuellen Daten (Salzburg Research o.J., [https://peakar.salzburgresearch.at])

Neben den Applikationen die stark auf die Anzeige einer gewissen Art von Daten spezialisiert sind, existieren auch einige Anwendungen die sich als Plattform zur Anzeige einer größeren Vielfalt an Informationen eignen sollen. Wie schon in Kapitel 2.2 am Beispiel Wikitude beschrieben, nutzen diese Apps sehr stark die Möglichkeit der Anzeige dynamisch geladener Inhalte. Im Fall von Wikitude, können externe Inhaltsanbieter ihre georeferenzierten Informationen in das System einpflegen. Benutzer sehen eine Liste von Inhaltsanbietern, die in ihrer Nähe Inhalte liefern und können sich diese Anzeigen lassen. Die Wikitude GmbH nennt dieses System „World Browser“ (vgl. Wikitude GmbH o.J.a, Hervorheb. i.O., [http://www.wikitude.com/tour/wikitude-world-browser]). Einen Ähnlichen Ansatz verfolgen auch die Applikationen junaio augmented reality und layar die jeweils für iOS und auch Android verfügbar sind. Dieser Typ von AR Plattformen verzichtet, wie auch die spezialisierteren Anwendungen aus der ersten Kategorie, größtenteils auf die Darstellung komplexer virtueller Objekte. Es werden hauptsächlich POI Positionsmarker gerendert die aus simplen 2D-Grafiken bestehen, siehe Abbildung 4 und 5. Dies erlaubt den Applikationen eine relativ große Menge an Grafiken gleichzeitig anzuzeigen. Bei Bewegungen des handheld device müssen jeweils nur 2D-Grafiken neu gezeichnet werden, komplexere 3D-Grafik Operationen können vermieden werden.

In die zweite Kategorie von AR Applikationen fallen all jene Anwendungen welche nicht nur 2D-Grafiken über dem Live-Kamerabild verorten, sondern auch 3D-Objekte in einer realen Umgebung darstellen. Durch Tracking der aktuellen Geräteposition werden die virtuellen Objekte auch in Echtzeit transformiert. Die Darstellung und Transformation von 3D-Objekten in Verbindung mit der konstanten Berechnung zu Positions- und Sichtfeldänderungen stellen hohe Rechenanforderungen dar. Applikationen dieser Kategorie zeigen daher meistens deutlich weniger virtuelle Objekte gleichzeitig an als etwa Applikationen die sich auf die Anzeige von 2D-Grafiken beschränken. In Google Play wie auch dem App Store finden sich häufig Spiele die die Überlagerung von 3d-Objekten bereits verwenden. ARBasketball verwendet etwa einen visuellen Marker, der

vom Benutzer ausgedruckt werden muss, um einen virtuellen Basketballkorb anzuzeigen. Mit Wischgesten kann der Benutzer nun versuchen virtuelle Basketballobjekte in den Korb zu werfen. Abbildung 6 zeigt den benötigten Marker, auf Abbildung 7 sieht man eine Szene aus dem Spiel.

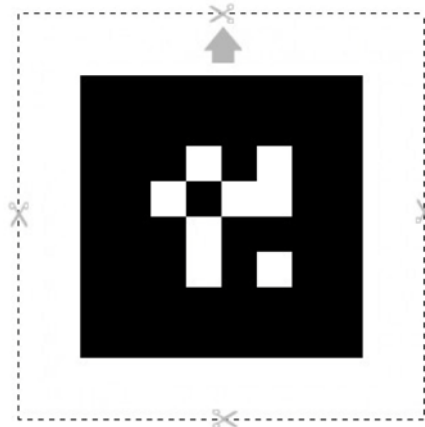


Abbildung 6: ARBasketball Marker (Tarovik o.J., [<http://simiotica.com/index.php>])

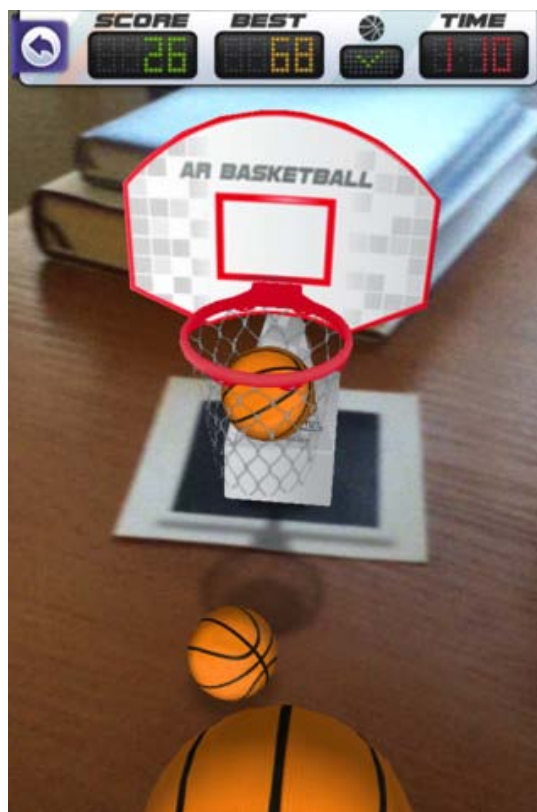


Abbildung 7: ARBasketball Spielszene (Apple Inc. 2012, [<http://itunes.apple.com/us/app/arbasketball/id393333529>])

Gerade Applikationen der zuletzt genannten Kategorie eignen sich gut um die technischen Grenzen derzeitiger mobiler AR Lösungen zu erkunden. Im Zuge der Erstellung dieser Arbeit konnten durch zahlreiche Tests von AR Apps immer wieder folgende Problemfelder bzw. technischen Einschränkungen festgestellt werden.

- **Marker:** Alle getesteten Applikationen benötigen visuelle Marker um ihre virtuellen Inhalte in der realen Umgebung zu platzieren. Der Benutzer muss somit um die App verwenden zu können erst den benötigten Marker drücken. Diese Tracking Marker sind speziell gestaltet um vom AR System möglichst einfach erkannt zu werden, trotzdem stoßen die Systeme an Grenzen. Gerade schlechte Lichtverhältnisse und sehr spitze Betrachtungswinkel führen häufig zu einem Verlust der Registrierung des Markers. Die virtuellen Inhalte können somit nicht mehr korrekt platziert werden bis der Marker wieder erkannt wurde. Gerade bei Spielen oder laufenden Animationen kann dieses Problem negative Auswirkungen auf die User Experience haben. Der Einsatz von zwei-dimensionalen Markern schränkt auch den möglichen Bewegungsraum des Benutzers ein.
- **Umgebungs- und Kontexterkennung:** Die Erkennung der realen Umgebung beschränkt sich bei den derzeitig vorhandenen mobilen AR Applikationen auf die Erkennung visueller Marker. Reale Objekte die etwa zwischen dem Benutzer und dem Marker liegen, werden nicht erkannt. Diese können somit auch nicht im Rendering der virtuellen Inhalte berücksichtigt werden. Die AR Darstellung basiert insgesamt auf separaten Ebenen. Die reale Umgebung befindet sich auf der Hintergrundebene. Alle erweiterten, virtuellen Inhalte werden auf darüber liegenden Ebenen dargestellt. Bis auf das System der bereits erwähnten visuellen Marker ist die Darstellung der virtuellen Inhalte somit vom realen Hintergrund nicht beeinflusst.
- **Ansprechverhalten und Responsiveness:** Bei einigen der analysierten Applikationen fiel mangelnde Responsiveness auf. Die Responsiveness ist ein wichtiger Bestandteil einer positiven User Experience. Grundsätzlich sollte jede Applikation möglichst direkt und sofort auf User Interaktionen reagieren können (vgl. Steiner 2011, S.11).
- **Geschwindigkeit:** Die für den Benutzer wahrgenommene Geschwindigkeit einer Applikation, hängt oft sehr stark mit der Responsiveness zusammen. Es gibt natürlich gewisse Zustände in einer Applikation, in denen der Benutzer mit gewissen Ladezeiten rechnet, etwa bei einem erstmaligen Aufruf der Anwendung oder beim Laden neuer Daten. Trotzdem sollten auch an diesen Stellen die Wartezeiten möglichst gering bleiben. Einige AR Anwendungen laden relativ häufig Daten bzw. Inhalte nach, diese Wartezeiten lassen die Benutzung leicht etwas schwerfällig wirken. Die Transformation und Animation von 3D-Objekten stellt im Rahmen von AR-Applikationen auch oft eine hohe Rechenlast für das mobile Gerät dar. Bei jeder Bewegung des handheld device müssen die dargestellten 3D-Inhalte neu gezeichnet werden. Um dem Benutzer eine flüssige Darstellung dieser Änderungen liefern zu können, müssen die dargestellten Objekte schnell genug neu gezeichnet werden damit das menschliche Auge die Einzelbilder nicht mehr unterscheiden kann. Zu diesem Zweck sollte die 3D-Darstellung so performant wie möglich abgewickelt werden.

- **Usability:** Augmented Reality ist im Bereich der mobilen Applikationen eine relativ junge Technologie. Neben einigen technischen Limitierungen existieren auch nur begrenzte Erfahrungswerte im Bereich der Benutzerinteraktion und Usability. Im Interaktionsdesign mobiler Anwendungen wird häufig auf erprobte Designpatterns und Best Practice Lösungen zurückgegriffen. Benutzer mobiler Anwendungen sind mit häufig verwendeten Designpatterns bereits vertraut, was die Einarbeitungszeit in neue Applikationen drastisch reduzieren kann. Mullet und Sano stellten schon 1994 fest:

“Because all graphical user interfaces (GUI’s) are communication systems, their design should be held to the same standards of functional and aesthetic relevance that have evolved over the centuries” (Mullet / Sano 1994, S.IX)

Entwickler mobiler Betriebssysteme wie Apple und Google stellen eigene Userinterface (UI) Guidelines zu diesem Zweck zur Verfügung (vgl. Steiner 2011, S.8). Im Bereich der AR Interaktion sind noch keine vergleichbaren UI Patterns vorhanden. Azuma attestierte hier schon 2001 Handlungsbedarf:

We need a better understanding of how to display data to a user and how the user should interact with the data. Most existing research concentrates on low-level perceptual issues, such as properly perceiving depth or how latency affects manipulation tasks. However, AR also introduces many high-level tasks, such as the need to identify what information should be provided, what’s the appropriate representation for that data, and how the user should make queries and reports. (Azuma et al. 2001, S.43)

In vielen aktuell verfügbaren AR Applikationen wirken sich inkonsistente Interaktionsmöglichkeiten und mangelnde Anwendungsregeln negativ auf die Usability aus.

3. AR Anwendungsfall - Architekturvisualisierung

3.1. Einleitung

Die Architekturvisualisierung befasst sich mit der visuellen Darstellung von Architekturentwürfen und bestehenden architektonischen Strukturen. Sie kommt in allen Bereichen der Architektur zum Einsatz, von der Innenraumgestaltung bis zur Planung von Gebäudekomplexen. Klassischerweise wurden zum Zwecke der Visualisierung häufig handgefertigte Zeichnungen erstellt und auch physische Modelle gebaut, siehe Abbildung 8. Mit der Entwicklung der ersten interaktiven CAD-Systeme (Computer Aided Design) begannen auch technische Büros mit der computerunterstützten Planung in architektonischen Projekten. Die verfügbaren CAD-Programme wurden stetig weiterentwickelt und vor allem um Möglichkeiten der 3D-Modellierung erweitert. So konnte auch ab 1995 ein zunehmender Umstieg von zweidimensionaler auf durchgängig dreidimensionale Modellierung bei produzierenden Architekturbüros verzeichnet werden (vgl. Beuchert 2010, S.20).

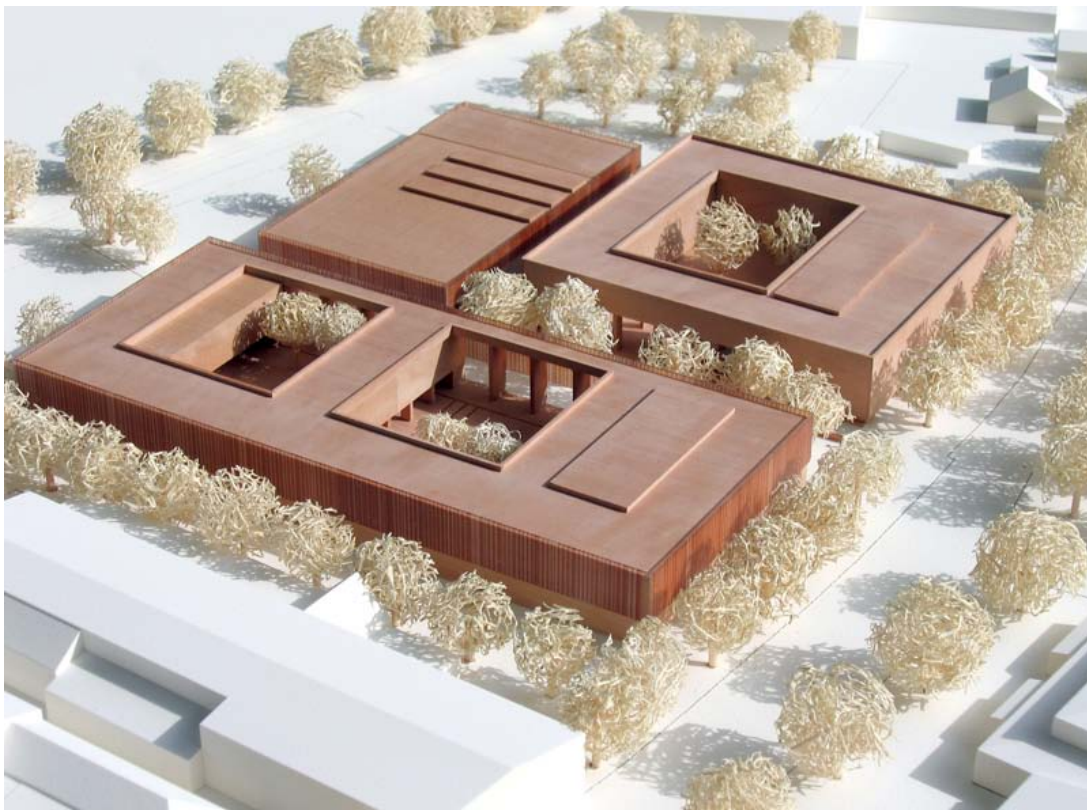


Abbildung 8: Physisches Architekturmodell (Leibl o.J., [<http://www.architekturmodelle.com/architekturmodelle/wettbewerbsmodelle.html>])

CAD-Software erlaubt die Erstellung sehr genauer, informationsreicher Pläne und Modelle. Durch Fotomontage-Techniken lassen sich erstellte 3D-Modelle auch in reale Umgebungen, wie mögliche Bauplätze, einfügen. Eine Alternative dazu stellen fotorealistische Darstellungen direkt aus den 3D-

Modellierungsprogrammen dar. Spezielle Visualisierungs-Erweiterungen ermöglichen diese realitätsnahe Darstellungsform. Ein Beispiel für eine derartige fotorealistische 3D-Visualisierung ist auf Abbildung 9 zu sehen.



Abbildung 9: Architektur 3D-Visualisierung (BlueLime Studio Inc. 2008, [<http://commons.wikimedia.org/wiki/File:Bluelime002.jpg>])

3.2. Beweggründe für den Einsatz mobiler AR Technologien

Der Einsatz von AR Technologien ist grundsätzlich in vielen verschiedenen Bereichen der Architekturvisualisierung möglich. Marker-basierte Anwendungen könnten etwa in der Darstellung von Gebäudemodellen sowie der Innenarchitektur Einsatz finden. Während Applikationen, welche ohne visuelle Marker tracken auch für Gebäudedarstellungen in realer Größe im Freien geeignet sind. Für all diese unterschiedlichen Einsatzbereiche lassen sich folgende gemeinsame Beweggründe definieren, warum mobile AR Technologien zum Zweck der Architekturvisualisierung eingesetzt werden könnten:

- **Interaktive Darstellungsform:** Die Darstellung von virtuellen Objekten, direkt eingebettet in eine reale Umgebung, stellt eine sehr direkte Form der Visualisierung dar. In relativ kleinen Maßstäben können zwar auch klassische, physische Modelle gebaut werden (Abbildung 8), durch die Darstellung rein virtueller Objekte würden die Grenzen der Darstellungsgröße allerdings komplett entfallen. So könnten Gebäude auch in realer Größe direkt am geplanten Bauort angezeigt werden. Größenverhältnisse innerhalb von Bauwerken oder anderen Objekten, sowie Verhältnisse zur räumlichen Umgebung, werden so sehr einfach visualisiert und für den Anwender auch besser begreifbar gemacht.

- **Mobilität:** Der mögliche Verzicht auf den Bau physischer Modelle oder auch der Aufstellung von Plakaten und Schautafeln, kann auf längere Sicht nicht nur eine Kostenersparnis darstellen, sondern auch einen Gewinn an Mobilität schaffen. Sei es bei Kundenpräsentationen, oder Besichtigungen am Objekt, der Baustelle oder in speziellen Schauräumen, mithilfe mobiler AR Technologie könnten viele Visualisierungsanwendungsfälle mithilfe einiger Tablet-Computer oder Smartphones und visuellen Markern erfüllt werden.
- **Zugänglichkeit:** Smartphones und Tablet-Computer sind mittlerweile weit verbreitet. Wie in Tabelle 1 zu sehen, wurden alleine im ersten Quartal 2012 weltweit 152,3 Millionen Smartphones ausgeliefert. Auch wenn nicht alle dieser Geräte in der Lage sind rechenintensive AR Applikationen auszuführen, besteht eine breite Basis an möglichen Benutzern solcher Anwendungen. Natürlich können bestehende CAD-Modelle auch in Form von Render-Bildern und Videos veröffentlicht werden. Eine Applikation würde den Benutzern jedoch eine direkte Interaktion mit dem Modell ermöglichen. Alleine die einfache Möglichkeit der Betrachtung aus mehreren Blickwinkeln, kann Usern einen besseren Überblick über räumliche Ausdehnungen und Verhältnisse bieten. Benutzer können durch das sogenannte 6DOF Tracking jederzeit ihre gewünschte Ansicht frei im dreidimensionalen Raum wählen. 6DOF steht für „six degrees of freedom“, die Position des handheld device wird also bestimmt durch die Lage auf den drei Raumachsen sowie der jeweiligen Rotation. Benutzern bietet sich dadurch die Möglichkeit, virtuelle Objekte aus jeder erdenklichen Lage zu betrachten. Die einzige Einschränkung ergibt sich durch die marker-basierte Registrierung von Inhalten.
- **Flexibilität:** Gerade in der Entwicklungsphase eines Projektes können sich Modelle und Visualisierungsanforderungen häufig ändern. Durch die rein virtuelle Anzeige der architektonischen Modelle kann der Änderungsaufwand stark reduziert werden. Wikitude, eine mobile AR Plattform, greift etwa dynamisch auf Webinterfaces zu um aktuelle Daten abzufragen. Ein ähnliches System wäre auch für Modelldaten aus CAD-Programmen denkbar. Durch einen solchen dynamischen Abgleich wären auch kürzere Entwicklungszyklen und häufigere Abstimmungen mit Kunden und Nutzern möglich, was einen direkteren Austausch von Verbesserungsvorschlägen, gerade in frühen Projektphasen ermöglicht.
- **Zusätzliche Informationen:** Durch Augmented Reality können nicht nur Geometriedaten von 3D-Modellen angezeigt werden sondern auch diverse Meta-Informationen. Diese Daten können etwa auch nach dem Vorbild Wikitudes dynamisch per Webinterface bereit gestellt werden. Benutzer haben per Touchscreen die Möglichkeit sich diese Informationen speziell anzeigen zu lassen. Beuchert erwähnt diese Möglichkeit von AR Systemen sehr positiv:

Ein virtuelles Gebäudemodell mit der Verknüpfung geometrischer und semantischer Informationen ist [...] als Grundlage für mobile Augmented Reality-Anwendungen denkbar. Insbesondere an den Schnittstellen zum und von diesem Modell könnten die hinterlegten Informationen besser dargestellt werden, was einen intuitiveren Umgang mit diesen Daten ermöglichen könnte. (Beuchert 2010, S.23)



3.3. Mögliche Problemfelder

Mobile Augmented Reality ist eine relativ junge Technologie und birgt noch einige Probleme. Die Grenzen derzeitiger Applikationen im Allgemeinen wurden bereits in Abschnitt 2.3 dargelegt. Für den Anwendungsfall der Architekturvisualisierung spielen natürlich viele dieser Punkte auch eine Rolle. Zusätzlich lassen sich aber auch einige spezielle Problemfelder identifizieren, die gerade in der Entwicklung von AR Applikationen in diesem Bereich, nicht übersehen werden dürfen. Azuma (2001) beschreibt etwa folgende Problemfelder bei der Darstellung von 3D-Inhalten in realen Umgebungen:

- **Fehlerabschätzung in der Darstellung:** Bei der Registrierung von virtuellen Inhalten in der realen Umgebung können leicht Fehler auftreten durch ungenaue bzw. unzureichende Positionierungsdaten. Die derzeit verfügbaren Applikationen stellen virtuelle Inhalte in solchen Fällen falsch, oder gar nicht dar. Eine bessere Lösung wäre im AR System bewusst auf ungenaue oder fehlende Daten zu reagieren und dem Benutzer Feedback darüber zu geben.
- **Datendichte:** Die Displaygrößen mobiler Geräte sind beschränkt, daher kann auch nicht jede beliebige Menge an Informationen gleichzeitig dargestellt werden. AR Systeme sollten in diesem Bereich besonders viel Wert auf Simplizität legen, da die reale Umgebung bereits eine hohe Informationsdichte aufweisen kann, welche durch das Hinzufügen weiterer Daten leicht den Benutzer überfordern kann. Im Gegensatz zu anderen Applikationen die mit großen Datenmengen umgehen müssen, haben AR Applikationen auch die Aufgabe die Interaktion zwischen der realen Welt und den virtuellen Informationen zu gewährleisten ohne die reale Welt verändern zu können (vgl. Azuma et al. 2001, S.39).
- **Fotorealistische Darstellung:** Moderne CAD-Software und 3D-Modellierungsprogramme erlauben die Erstellung fotorealistischer Visualisierungen. Zum Teil sind Unterschiede zu Fotografien realer Objekte nur mehr sehr schwer auszumachen. Für Darstellungen dieser Art bedarf es aber in der Regel sehr komplexer Programme zur Berechnung von Lichtquellen, Schattenwürfen, Spiegelungen, Materialien, Oberflächen und sonstigen Darstellungsfeinheiten. In diesem Bereich treten sehr schnell die technischen Grenzen mobiler Prozessoren und Grafikprozessoren in Erscheinung. Zumindest mit der derzeitigen Rechenleistung ist eine fotorealistische Darstellung in Echtzeit auf mobilen Geräten praktisch unmöglich. Für eine realistischere Darstellung müssten sich die Systeme auch mehr Kenntnisse über die geometrischen Eigenschaften der realen Umgebung erarbeiten können, damit etwa reale Objekte im Vordergrund die Darstellung der erweiterten Inhalte beeinflussen können. Azuma beschrieb diese nicht-triviale Aufgabe bereits 2001: „The problem of removing real objects goes beyond extracting depth information from a scene; the system must also segment individual objects in that environment. (Azuma et al. 2001, S. 39)

3.4. Anwendungsbeispiele

Im Rahmen der Rechercharbeiten zum Thema Architekturvisualisierung und Augmented Reality wurden zwei interessante Applikationen für diesen Anwendungsfall gefunden. SnapShop Showroom der Firma SnapShop Inc. Soll durch Augmented Reality Technologie neue Möglichkeiten der Innenraumgestaltung bieten. 3DOn ARchitecture der Firma 3DOn Ltd kann genutzt werden um 3D-Modelle mobil zu visualisieren und auch durch AR Technologie in realer Umgebung anzuzeigen. Obwohl die Anwendungen jeweils einige spezielle Funktionalitäten aufweisen, ist die Technik der AR Darstellung relativ ähnlich. SnapShop Showroom verwendet in diesem Bereich eine sehr simple Methode der Überlagerung virtueller Inhalte, die nur einen extrem kleinen Teil der Möglichkeiten erweiterter Realitätsdarstellungen unterstützt. Konkret steht eine große Bibliothek an Einrichtungsgegenständen zur Auswahl, welche über einen realen Hintergrund dargestellt werden können. Dem Benutzer soll ein visueller Eindruck des Objekts in einem von ihm frei wählbaren Raum geliefert werden. Bei dieser Überlagerung passiert jedoch überhaupt kein Tracking der räumlichen Umgebung, die virtuellen Einrichtungsgegenstände können zwar rotiert und verschoben werden, sie bestehen jedoch nur aus simplen zwei-dimensionale Bilddateien. Diese Bilder können über einer beliebigen Fotografie positioniert werden. Hier passiert also keine Einbettung von virtuellen Inhalten in eine reale Umgebung, als vielmehr eine Fotomontage. Das User Interface von SnapShop Showroom ist zu sehen in Abbildung 10, eine „AR“ Montage aus der Applikation in Abbildung 11.



Abbildung 10: SnapShop Showroom Interface

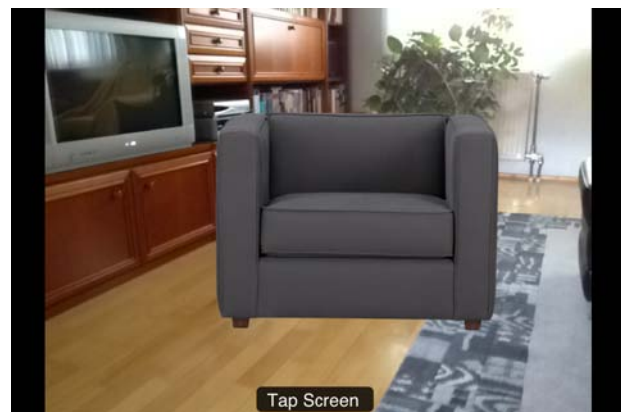


Abbildung 11: SnapShop Showroom „AR“ Montage

3DOn ARchitecture bietet im Gegensatz zu SnapShop Showroom die Anzeige echter 3D-Modelle. Diese Modelle können durch (Multitouch)-Gesten verkleinert, vergrößert, verschoben und gedreht werden. Allerdings ist auch hier keine echte Einbettung in eine Reale Umgebung möglich, sondern nur eine Überlagerung einer Fotografie. In der Applikation sind zwar auch noch andere Darstellungsmöglichkeiten vorgesehen, die eine Visualisierung mithilfe von GPS-Daten des Gerätes ermöglichen sollen, jedoch schienen sie noch fehlerhaft, so dass keine Überlagerung virtueller Inhalte vorgenommen werden konnte. Auf Abbildung 12 sieht man die 3D-Ansicht eines Architekturmodells in 3DOn ARchitecture. Es ist zu erkennen, dass keine Einbettung in einen realen Kontext vorgenommen wird, sondern lediglich ein nicht weiter interpretiertes Hintergrundbild dargestellt wird.



Abbildung 12: 3DOn ARchitecture Visualisierung eines 3D-Modells

4. Anforderungsbasierte Evaluierung der Umsetzung von AR Systemen zur Architekturvisualisierung

4.1. Evaluierungsmethodik

Das Ziel dieser Evaluierung ist es zu bestimmen, inwieweit derzeitige Umsetzungsmöglichkeiten von mobilen AR Systemen dazu geeignet sind zum Zweck der Architekturvisualisierung eingesetzt zu werden. Wie in Abschnitt 3.2 beschrieben, gibt es einige Beweggründe für den Einsatz von AR Technologie in diesem Bereich. Aus diesen Beweggründen lassen sich gewisse Anforderungen an AR-Applikationen und deren Umsetzung aufstellen. Zum Einen aus der Anwendersicht, zum Anderen aber auch aus der Sicht der Entwickler und Zulieferer. Unter Zulieferern werden im Rahmen der Evaluierung Content Provider von Geometrie-Rohdaten, Modellen und generell Inhalten und Informationen betrachtet, die für die Architekturdarstellung notwendig sind. Die Anforderungsaufstellung stellt im weiteren Verlauf die Basis der Evaluierung verschiedener Umsetzungsmöglichkeiten dar. Schlussendlich sollte ein umfassender Überblick gegeben sein, inwieweit die derzeit verfügbaren technischen Möglichkeiten, eine zufriedenstellende Verwirklichung von mobilen AR-Projekten im Bereich der Architekturdarstellung zulassen.

4.2. Anforderungen an AR-Applikationen

Um AR Technologie zum Zweck der Architekturdarstellung sinnvoll einsetzen zu können, müssen gewisse Grundvoraussetzungen bzw. Anforderungen erfüllt sein. Wie in Abschnitt 2.3 ersichtlich können etwa bei derzeitig verfügbaren AR-Applikationen einige technische und funktionale Grenzen festgestellt werden. Da diese Technologie im mobilen Einsatz noch recht jung ist, sind diese Erkenntnisse auch nicht weiter verwunderlich. Betrachtet man aber den speziellen Anwendungsfall Architekturvisualisierung, gibt es einige Beweggründe für den Einsatz einer solchen Technologie. Aus diesen Beweggründen ergeben sich im Falle einer konkreten Umsetzung natürlich gewisse Erwartungen und Anforderungen verschiedener Seiten. Im Rahmen dieser Evaluierung werden daher auch Umsetzungsmöglichkeiten und deren Auswirkungen aus mehreren Perspektiven betrachtet. Die wohl direktesten Anforderungen an eine Applikation oder ein System haben dessen Anwender. Sie sind die Benutzer der fertiggestellten Anwendung. Funktionalität, Preis und auch User Experience spielen hier eine wichtige Rolle. Eine weitere Sicht auf das AR-System haben die Entwickler. Hier kann es nicht nur Anforderungen an die eigentliche Implementierung geben, sondern auch an Wartbarkeit, Erweiterbarkeit, Dokumentation und Rentabilität. Eine auch nicht ganz unwichtige Rolle stellen die Content Provider, also Zulieferer von Inhalten dar. Im Architekturbereich kann es sich hier etwa um Geometriedaten, Gebäudeinformationen und Grafiken handeln. Da auch Zulieferer oftmals direkt von der späteren Verwendung ihrer Arbeiten betroffen und zum Teil auch Abhängig sind, können hier Anforderungen an neue Visualisierungssysteme auftreten. Die folgende Auflistung beschreibt die Anforderungen an mobile AR-Applikationen und deren Umsetzung im Bereich der Architekturvisualisierung:



- **Sicht der Anwender**

- **Darstellungsqualität:** Wie auch auf Abbildung 9 zu sehen, gibt es im Bereich der Architekturvisualisierung bereits seit einigen Jahren die Möglichkeit zur Visualisierung von Objekten in fotorealistischer Qualität. Natürlich handelt es sich hier um vorgerendertes Bildmaterial und nicht um eine transformierbare Live-Darstellung wie es in AR-Applikationen der Fall ist. Da aber bei großen räumlichen Strukturen auch Details eine wichtige Rolle spielen können muss eine gewisse Darstellungsqualität erreicht werden. Azuma merkt hierzu auch an: „For some applications, virtual augmentations should be indistinguishable from real objects.“ (Azuma et al. 2001, S.39)
- **Robustheit:** Wie auch bei anderen Anwendungen ist die Stabilität und Robustheit ein wichtiger Faktor für eine positive User Experience. Im Falle von AR-Anwendungen ist dieser Punkt auf zwei unterschiedliche Arten von Bedeutung. Zum einen sollte das Tracking und die Registrierung der virtuellen Inhalte eine gewisse Stabilität gegenüber diversen Umwelteinflüssen, wie direkter Sonneneinstrahlung und schwacher Helligkeit aufweisen. Zum Anderen muss das System als Ganzes relativ Ausfallssicher sein. Das Laden größerer Datenmengen und die Darstellung komplexer 3D-Objekte müssen durch das System bewältigt werden. Dieser Punkt kann auch im Kontext allgemeinerer Kriterien für die Qualität von User Interfaces gesehen werden: „The fifth point stability can be the crucial factor [...] Especially when working with complex applications an abnormal program abort can be a big problem for the user.“ (Steiner 2011, S.12)
- **Geschwindigkeit:** Grundsätzlich sollten auch komplexere Visualisierungen und Überlagerungen nicht zu einem Einbruch der Geschwindigkeit führen. Ein wichtiges Ziel in der Entwicklung benutzerfreundlicher User Interfaces ist es, möglichst rasch auf Benutzereingaben zu reagieren. (vgl. Colborne 2010, S.44)
- **Zugänglichkeit:** Der Zugang zum AR-System sollte für den Endanwender nicht zu kompliziert sein. Wie bei anderen mobilen Applikationen wäre der Download über eine App-Marktplattform eine simple Lösung, da etwa die Installation ohne weitere Benutzerinteraktion ablaufen kann.
- **Usability:** Wie auch bei allen anderen mobilen Anwendungen ist die Usability ein wichtiger Faktor für eine optimale Bedienbarkeit. Eine genauere Beschreibung bezogen auf AR-Applikationen ist unter Punkt 2.3 zu finden.
- **Wartung / Service:** Speziell im Fall von kommerziellen Applikationen, gibt es auch die Anforderung nach Serviceleistungen und Wartung. Dazu zählen etwa Software-Updates zur Fehlerbehebung und Kundensupport bei Fragen und Problemen im Umgang mit der AR-Applikation.

- **Sicht der Entwickler**

- **Coding:** Anforderungen bzw. Vorlieben im Bereich der Programmierung sind je nach Entwicklungsabteilung und Entwickler unterschiedlich. Ausbildung, Berufserfahrung und

bestehende Frameworks können hier Entscheidungen beeinflussen. Einige Punkte können jedoch die Arbeit für die meisten Entwickler erleichtern. Etwa eine umfassende Dokumentation, eine große Entwickler-Community und die Verwendung verbreiteter, bewährter Programmiersprachen.

- **Wartung:** Oftmals werden in Softwareprojekten große Teile des Gewinns durch Wartung nach der eigentlichen Erstellung eines Produkts erzielt. Auch bei auftretenden Fehlern sind Softwareupdates oft unumgänglich. Im Rahmen dieser Evaluierung spielt die Wartung in zweierlei Hinsicht eine Rolle. Einerseits ist die Wartung und Aktualisierung des eigentlichen Frameworks oder der Programmierschnittstelle ein wichtiger Punkt, andererseits ist aber auch interessant welche Möglichkeiten der späteren Fehlerbehebung und des Debuggings eine Umgebung für den Entwickler bietet.
- **Zielplattformen:** Für welche Betriebssysteme und Geräte kann eine Applikation ausgeliefert werden. Wie auch Tabelle 1 zeigt kann dies zu drastischen Auswirkungen auf die Menge der potentiellen Anwender führen.
- **Erweiterbarkeit:** Ein wichtiger Punkt für die Mittel- bis Längerfristige Planung können Erweiterungsmöglichkeiten eines Frameworks darstellen und wie flexibel auf weitere Kundenwünsche reagiert werden kann.
- **Sicht der Content Provider**
 - **Datenformate:** Wichtig für den Content Provider ist die Kompatibilität seiner Daten zu den erwarteten und unterstützten Eingabeformaten des AR-Systems. Falls bereits in der Vergangenheit Daten für Visualisierungszwecke bereitgestellt wurden, ist es interessant ob durch die Erweiterung oder Ersetzung durch ein mobiles Augmented Reality System Formate angepasst werden müssen und welche Aufwände solche Veränderungen mit sich bringen. Gängige Formate sind im CAD Bereich etwa das DXF (Drawing Interchange Format) und DWG (Drawing) Datenformat, welche im populären Programm AutoCAD Einsatz finden. In der 3D-Modellierung werden häufig die Formate 3DS (3D-Studio File Format) und OBJ (Wavefront Object) zum Datenaustausch eingesetzt.
 - **Darstellungsleistung:** Möglicherweise müssen sich auch die Daten an sich in ihrer Struktur und Granularität ändern. Im Falle von 3D-Daten führt vielleicht eine Obergrenze an darstellbaren Polygonen zu geänderten Anforderungen in der Datenaufbereitung.
 - **Zugang:** Für den Content Provider ist schlussendlich auch wichtig in welcher Form die Datenanlieferung ans AR-System funktioniert. Ob etwa ein direkter Zugang zum System besteht oder die Daten nur über den eigentlichen Entwickler eingepflegt werden können.



4.3. Umsetzungsmöglichkeiten

Mobile AR Applikationen können auf viele unterschiedliche Arten umgesetzt werden. Die Bandbreite der Möglichkeiten reicht von einer kompletten Eigenentwicklung bis hin zu einem reinen Datenimport in ein bestehendes AR-System. Im Rahmen dieser Evaluierung wurde der Fokus aber auf Frameworks und Libraries gelegt, die dem Entwickler grundlegende Teile der AR Technologie bereits zur Verfügung stellen. Für den Zweck der Architekturvisualisierung ist es von Interesse ob diese Umgebungen überhaupt die Entwicklung von Anwendungen erlauben, die den oben genannten Anforderungen entsprechen. Und wenn ja, in welchem Ausmaß sie die Entwicklung unterstützen und beschleunigen könnten. Die folgende Liste von Augmented Reality Frameworks wurde zur Evaluierung herangezogen:

- **Proprietäre Frameworks**

- **Vuforia Augmented Reality SDK:** Das Vuforia SDK (Software Development Kit) ist ein proprietäres AR Framework der Firma Qualcomm Incorporated. Obwohl es sich um keine quelloffene Software handelt, ist das Framework ohne Lizenzkosten auch kommerziell einsetzbar. Es unterstützt die Entwicklung von AR-Applikation für die mobilen Betriebssysteme Android (ab Version 2.1) und iOS (Version 4 und 5). Integriert ist räumliches Tracking mithilfe selbst erstellbarer visueller Marker und etwa auch die Unterstützung für Benutzerinteraktionen mithilfe virtueller Buttons, die auf dieselbe Art wie andere virtuelle Inhalte über die reale Umgebung platziert werden können. Die Entwicklung von AR-Apps kann mithilfe dieses Frameworks entweder über die üblichen Entwicklungsumgebungen für native Android und iOS Applikationen erfolgen (Eclipse und Xcode), oder per Plugin auch im Unity Editor. Der Unity Editor ist die Entwicklungsumgebung zur Spieleentwicklung mithilfe der Unity Game-Engine und erlaubt auch den Export auf Android- und iOS-Geräte. Nähere Informationen zum Vuforia SDK finden sich auf der offiziellen Webseite des Herstellers Qualcomm (vgl. Qualcomm Incorporated o.J.a, [<http://developer.qualcomm.com/mobile-development/mobile-technologies/augmented-reality>]).
- **Studierstube Tracker:** Die Studierstube Tracker Library ist speziell für das Tracken visueller Marker entwickelt worden und soll sich durch besonders hohe Performanz auszeichnen. Sie wurde zwar nicht speziell für die Verwendung auf mobilen Geräten entwickelt, da aber externe Abhängigkeiten vermieden wurden soll die Portierung auf mobile Plattformen möglich sein. Erstellt wurde die Library im Christian Doppler Laboratory für Handheld Augmented Reality der Technischen Universität Graz. Weitere Details finden sich auf der Webseite der Entwickler (vgl. Christian Doppler Laboratory for Handheld Augmented Reality - TU Graz 2011, [<http://handheldar.icg.tugraz.at/stbtracker.php>]).
- **metaio Mobile SDK:** Die Firma metaio befasst sich seit dem Jahr 2003 mit der Entwicklung von AR Lösungen. Unter mehreren verschiedenen angebotenen Frameworks ist speziell das metaio Mobile SDK zur Entwicklung von mobilen AR-Lösungen auf handheld devices konzipiert. Das proprietäre Framework ist gratis verwendbar wenn man die Einblendung von metaio Logos und

Schriftzügen in den erstellten Applikationen akzeptieren kann. Um Applikationen ohne vorgeschriebenes Branding erstellen zu können fallen hohe Lizenzkosten an. Zu den Features zählt ein eingebauter 3D Renderer, marker-basiertes und marker-freies Tracking sowie ein Plugin zur Erstellung von Applikationen mithilfe des Unity Editors. Nähere Informationen zu den einzelnen Komponenten und ein Entwickler Forum finden sich auf der metaio Webseite (vgl. metaio GmbH 2012a, [<http://www.metaio.com/software/mobile-sdk/>]).

- **Wikitude Augmented Reality SDK:** Die Entwicklerfirma der schon in Kapitel 2 erwähnten AR-Applikation Wikitude bieten mit dem SDK Entwicklern die Möglichkeit mehr als nur georeferenzierte Informationen in den Wikitude World Browser zu laden. Basierend auf den Webtechnologien HTML, JavaScript und CSS sollen sich flexible AR-Lösungen erstellen lassen. Das SDK kann man auf dem Android Betriebssystem ab Version 2.2 und auf iOS ab Version 4 verwenden. Für nicht kommerzielle Projekte ist die Umgebung gratis verwendbar. Wikitude Branding ist dann aber auch Teil jeder entwickelten Applikation. Lizenzen ohne Wikitude Branding für kommerzielle Projekte sind ebenfalls erhältlich. Weitere Informationen sind auf der Wikitude Webseite verfügbar (vgl. Wikitude GmbH o.J.b, [<http://www.wikitude.com/developer/wikitude-augmented-reality-for-developers>]).
- **D'Fusion Mobile:** Total Immersion stellt mit ihrer D'Fusion Serie einige Frameworks zur Erstellung von AR-Applikationen zur Verfügung. Mit D'Fusion Mobile soll vor allem die Entwicklung von nativen Android und iOS AR-Applikationen unterstützt werden. Total Immersion bietet zu diesem Zweck ihre eigene Entwicklungsumgebung, das D'Fusion Studio an. Auch ein Plugin für den Unity Editor ist zumindest in einer Beta-Version verfügbar. Zu den Features zählt Tracking mit und ohne visueller Marker, sowie die Verschlüsselung verwendeter 3D-Objekte und Videos, um Datendiebstahl vorzubeugen. Mehr Infos zu Features und Entwicklung sind auf der Total Immersion Webseite verfügbar (vgl. Total Immersion o.J., [<http://www.t-immersion.com/products/dfusion-suite/dfusion-mobile>]).
- **Open Source Frameworks**
 - **ARToolKit:** Das ARToolKit ist eine Software Bibliothek zur Erstellung von Augmented Reality Applikationen. Sie wurde von Dr. Hirokazu Katu entwickelt und wird derzeit von mehreren Universitäten und der Firma ARToolworks Inc. betreut. Die Library ist ein Open Source Projekt und steht unter der GPL Lizenz (GNU General Public License). Kommerzielle Lizenzen sind jedoch auch verfügbar. ARToolkit ist zwar nicht direkt für mobile Betriebssysteme verfügbar, allerdings gibt es Erweiterungen zur mobilen Nutzung. Nähere Informationen sind auf der offiziellen Webseite des Projekts verfügbar (vgl. HITLab o.J., [<http://www.hitl.washington.edu/artoolkit/>]).
 - **AndAR:** AndAR ist ein Open Source Projekt und steht wie ARToolKit unter der GPL zur Verfügung. Das Projekt ermöglicht mobiles marker-basiertes Tracking durch Portierung der

ARToolkit Library als Android Framework. Genaue Informationen zur Struktur des Projektes sind auf der Projektwebsite verfügbar (vgl. Domhan o.J., [<http://code.google.com/p/andar/>]).

- **mixare:** Bei dem Projekt mixare handelt es sich um eine Open Source Augmented Reality Engine unter der GPLv3 Lizenz. Sie bietet Unterstützung der Betriebssysteme Android und iOS. Es handelt sich dabei grundsätzlich um eine eigenständige Applikation, die entweder von einer eigenen App aus aufgerufen werden kann oder auch komplett integriert werden kann. Nähere Informationen zu dem Projekt befinden sich auf der zugehörigen Webseite der Entwicklerfirma Peer GmbH (vgl. Peer GmbH 2010, [<http://www.mixare.org/>]).

4.4. Evaluierungsergebnisse

Im Folgenden sind die Ergebnisse der Anforderungsbasierten Evaluierung der AR-Frameworks beschrieben. Wobei eine Aufteilung in Sichtweisen der Anwender, Entwickler und Content Provider vorgenommen wurde.

Sicht der Anwender

Die Anforderung der Darstellungsqualität ist grundsätzlich beschränkt durch die Leistungsfähigkeit derzeitiger mobiler Geräte und der darauf verwendeten und unterstützten 3D-Schnittstellen. Auf den mobilen Betriebssystemen Android und iOS kommt etwa die Open Graphics Library for Embedded Systems (OpenGL ES) zum Einsatz. Die Darstellung von 3D-Inhalten muss also konform zu dieser Schnittstelle passieren. Es existieren verschiedene Render Engines für mobile Systeme, metaio bewirbt ihr Mobile SDK mit einer integrierten Render-Engine, es unterstützt aber auch wie andere Frameworks die Unity Game-Engine. Diese Engines unterstützen viele unterschiedliche Darstellungsverfahren von 3D-Inhalten – basierend auf OpenGL ES. Da diese Grafikschnittstelle auch bei Spielekonsolen wie der PlayStation 3 verwendet wird, ist erkennbar, dass der Flaschenhals eher in der vorhandenen Hardware-Leistung als an den verfügbaren Software-Schnittstellen liegt.

Zu Anforderungen an Robustheit und Stabilität können nur schwer Aussagen getroffen werden. Je nach Komplexität und Menge der darzustellenden 3D-Daten können jedenfalls andere Anforderungen an Speicher und Prozessoren entstehen. Die Studierstube Tracker Library verspricht hohe Performanz und Leistungsfähigkeit beim Tracking von visuellen Markern, was auch der Anforderung nach einer ausreichenden Geschwindigkeit nachkommt. Ein Beispiel des gleichzeitigen Trackings einer hohen Anzahl von Markern ist auf Abbildung 13 zu sehen.

Die Analyse der unterschiedlichen AR Frameworks zeigte, dass es im Bereich der Zugänglichkeit für den Anwender keine Unterschiede gibt. Jede der vorgestellten Lösungen erlaubt das Erstellen nativer mobiler Applikationen, welche in gewohnter Weise über die bereits erwähnten App-Marktplattformen verteilt werden können. Dieser Weg des Zugangs zu Programmen ist Smartphone und Tablet Benutzern im Normalfall

bekannt und sollte hier somit zu keinen weiteren Problemen führen. Weitere Installationsprozeduren entfallen somit gänzlich. Externe Plugins und Erweiterungen werden in keinem Fall benötigt.

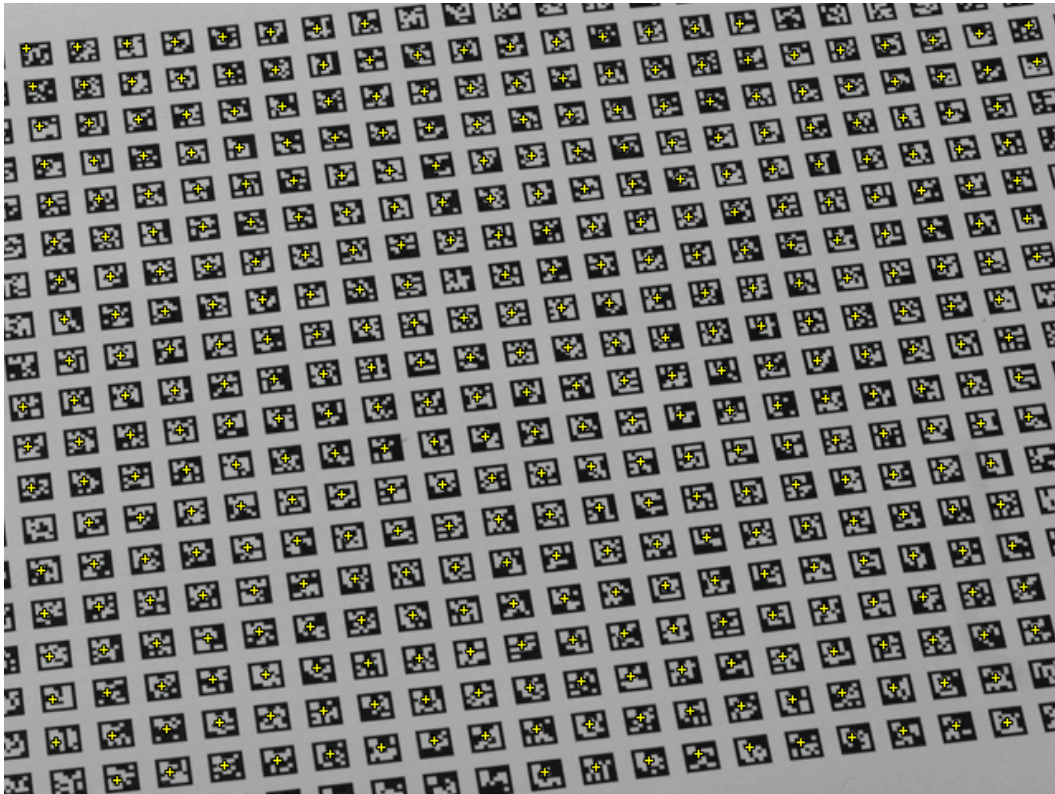


Abbildung 13: Studierstube Tracker – gleichzeitige Erkennung einer hohen Anzahl Marker (vgl. Christian Doppler Laboratory for Handheld Augmented Reality - TU Graz 2011, [<http://handheldar.icg.tugraz.at/stbtracker.php>])

Die Erfüllung der Anforderung nach geeigneter Usability hängt stark vom Typ der AR-Anwendung ab. Eine Spiele-Applikation müsste etwa ganz anders bedienbar sein als eine App zur Gebäudevisualisierung für Architekten. Wie schon unter Punkt 2.3 beschrieben ist ein wichtiger Faktor der Einsatz von bekannten Interaktionsmustern. Da jedes der analysierten Frameworks als reine Programmierschnittstelle in der gewohnten Entwicklungsumgebung für native Applikationen verwendet werden kann, gibt es keine Einschränkung bei der Wahl der Interaktionselemente. Die Frameworks Vuforia Augmented Reality SDK, metaio Mobile SDK und D'Fusion Mobile bieten auch die Option mobile AR-Applikationen im Unity Editor zu erstellen. Da es sich hierbei um eine Umgebung handelt die für die Spieleentwicklung geschaffen wurde, ist es hier schwierig die UI-Patterns mobiler Betriebssysteme abzubilden. Die Wahl der Entwicklungsumgebung sollte also in Hinblick auf die jeweiligen Projektanforderungen mit Bedacht getroffen werden.

Die Anforderungen nach Serviceleistung und Wartung hängen mit der Zugänglichkeit der AR-Lösungen zusammen. Da jedes der analysierten Augmented Reality Frameworks eine Auslieferung fertiggestellter Applikationen über die jeweiligen Applikations-Marktplattformen zulässt, können diese Wege auch zur

Auslieferung von Software-Updates genutzt werden. Inwieweit die AR-Softwarelösungen allerdings wirklich gewartet und aktualisiert werden, liegt in der Verantwortung der jeweiligen Entwickler.

Sicht der Entwickler

Die Entscheidung von Entwicklern ein bestimmtes Framework einzusetzen wird neben vielen Anderen Merkmalen sicherlich auch häufig durch die zugrundeliegenden Technologien bestimmt. Welche Programmiersprachen und Scriptingsprachen verwendet werden ist eine wichtige Information, da oftmals auch nicht in allen Sprachen und Umgebungen dasselbe Know-How existiert. Setzt ein Framework nun auf eine populäre, weit verbreitete Programmiersprache, kann das Vorteile für Entwickler mit sich bringen. Entwickler mit einschlägiger Programmiererfahrung sind häufig einfacher zu finden oder vielleicht schon im Development-Team vorhanden. Einige der umfangreicheren proprietären AR-Frameworks unterstützen auch die Verwendung ihrer Klassenbibliotheken und Schnittstellen in mehreren Programmiersprachen. Das Vuforia Augmented Reality SDK beispielsweise kann entweder direkt in der nativen Programmierung von mobilen Applikationen eingesetzt werden, oder aber auch durch ein Unity Plugin über dessen Scripting und Programmiermöglichkeiten. In diesem Fall übernimmt das Unity System die Umsetzung in das native Format des jeweiligen Zielsystems. Eine ähnliche Auswahlmöglichkeit bieten auch das metaio Mobile SDK und D'Fusion Mobile. Das D'Fusion Mobile Framework bietet auch Schnittstellen zum Zugriff auf maschinennahen Code, der zum Beispiel in der Grafik-Programmierung häufig benötigt wird. In Abbildung 14 ist ein schematischer Aufbau des SDK für Android und iOS Programmierung zu sehen. Der Entwickler kann durch reine High-Level Programmierung in Java (für Android) und Objective-C (für iOS) AR-Applikationen entwickeln. Das Wikitude Augmented Reality SDK geht einen anderen Weg und baut ihr Framework auf modernen Webtechnologien auf. 3D-Rendering ist nicht Bestandteil des Systems, da das Framework zur Entwicklung von standortbezogenen AR-Applikationen konzipiert wurde. Somit bedarf es auch keines Zugriffs auf 3D-Grafik Libraries. Technologien wie JavaScript können aber trotzdem für viele Formen der 2D-Visualisierungen, auch im Architekturbereich eingesetzt werden.

Die Anforderungen nach regelmäßiger Wartung und Aktualisierung von Frameworks können je nach Lizenzmodell relativ unterschiedlich ausfallen. Beim Kauf proprietärer Software sollten sich Entwickler ein gewisses Maß an Serviceleistungen erwarten können. Dies ist auch einer der Gründe, warum manche Open-Source Frameworks ein duales Lizenzmodell anbieten. Einer dieser Vertreter ist die vorgestellte Softwarebibliothek ARToolKit. Die frei verfügbare, unter GPL stehende Version der Library wird von einigen universitären Labors betreut. Hier liegt der Fokus eher auf der Weiterentwicklung des Frameworks als auf Kundensupport und speziellen Serviceleistungen. Für diese Anforderungen wurde ein proprietäres Lizenzmodell eingeführt, dass von der Firma ARToolworks, Inc. zur Verfügung gestellt wird. Auf der Webseite des Frameworks wird klar festgestellt, dass eine große Nachfrage nach kommerziellen Verwertungsrechten und erweiterten Support-Leistungen besteht.

Grundsätzlich kann man auch bei den anderen vorgestellten Frameworks von Mehrleistungen im Kundensupport bei proprietären Lösungen ausgehen. Bei quelloffenen Systemen werden Probleme dagegen oftmals in den entsprechenden Communities bearbeitet.

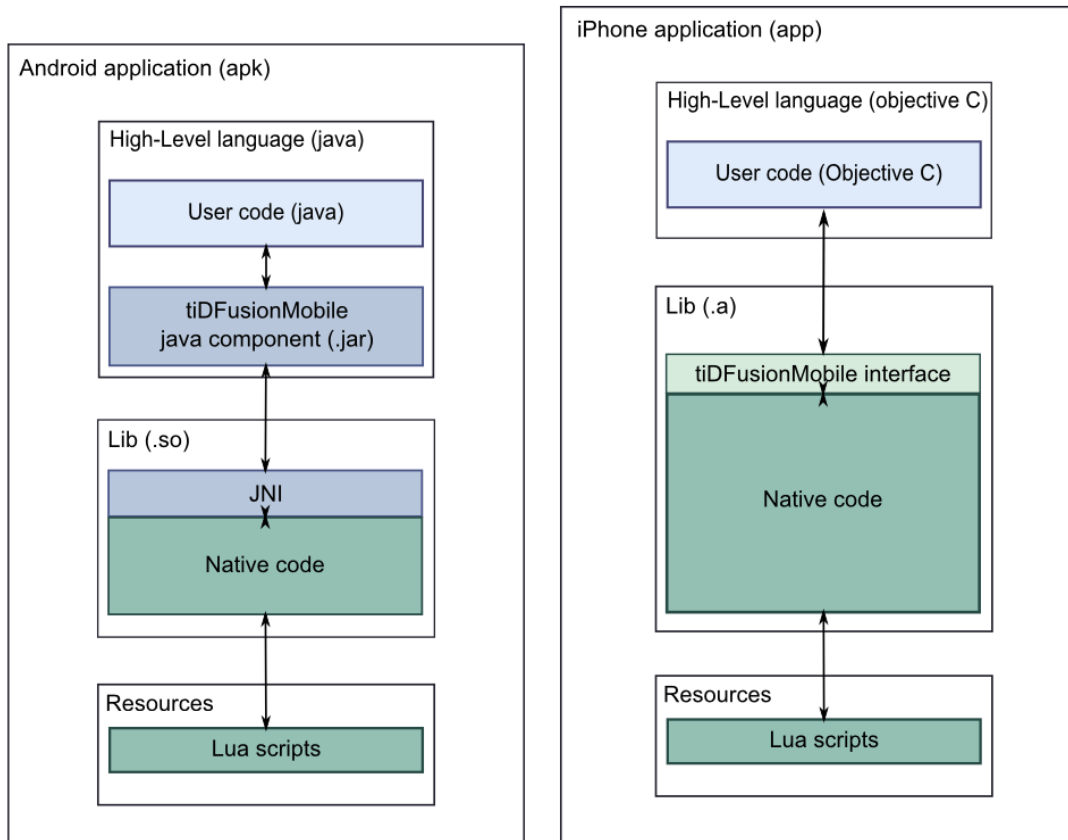


Abbildung 14: D'Fusion Mobile SDK Komponenten-Integration für Android und iOS (Total Immersion 2011, [\[https://community.t-immersion.com/getting-started\]](https://community.t-immersion.com/getting-started))

Die Forderung nach geeigneten Zielplattformen zur Verbreitung der Visualisierungsapplikationen wird zumindest von fast allen der analysierten, proprietären Frameworks unterstützt. Vuforia Augmented Reality SDK, metaio Mobile SDK, Wikitude Augmented Reality SDK, D'Fusion Mobile und auch das offene mixare unterstützen die Veröffentlichung auf Android und iOS Geräte. Es muss jedoch in jedem Fall einzeln berücksichtigt werden, dass gewisse Grenzen in der Unterstützung älterer Geräte und Betriebssystem-Versionen bestehen. Speziell im Bereich der 3D-Unterstützung kann es mit älteren Plattformen Probleme geben. Die entsprechenden Informationen für Entwickler finden sich jedoch in den jeweiligen technischen Informationsbereichen der Hersteller. Für spezielle Visualisierungseinsätze mit AR-Technologie, in denen das verwendete handheld device definiert werden kann ist die Auswahl an Zielplattformen von geringerer Wichtigkeit. In diesen Fällen sind andere Themen wie etwa Tracking-Performanz und Robustheit von größerer Bedeutung. Die Studierstube Tracker Library wurde beispielsweise bewusst nach den Anforderungen für hohe Performanz und niedrigen Speicherverbrauch entwickelt.

Erweiterungen der Funktionalität werden in den analysierten Frameworks hauptsächlich durch Updates des eigentlichen Produktes durchgeführt. Eine Ausnahme stellt Total Immersion dar, für ihr Produkt D’Fusion Mobile werden auf der offiziellen Webseite einige Erweiterungen zum Download bereitgestellt. Diese Plugins erweitern das Hauptprodukt etwa durch die Möglichkeit verschiedener File-Imports aus 3D-Modellierungsprogrammen und auch um Tools zum Einsatz von Partikelsystemen. Weiters wird auch ein Plugin für die Entwicklung mithilfe des Unity Editors von Total Immersion angeboten. Ähnliche Erweiterungen sind auch für die Frameworks metaio Mobile SDK und Vuforia Augmented Reality SDK frei verfügbar. Für die Erweiterbarkeit spielt auch die Systemarchitektur der Plattformen eine Rolle. Hier haben sich modulare Modelle durchgesetzt. Logisch trennbare Teile wie 3D-Rendering, Tracking, und GUI-Darstellung werden in einzelne Module gegliedert und können so leichter aktualisiert oder ausgetauscht werden. Eine schematische Darstellung der modularen Trennung von Verantwortlichkeiten, wie sie im metaio Mobile SDK vorgenommen wurde, ist auf Abbildung 15 zu sehen. Getrennte systemnahe Komponenten kümmern sich um das Tracking der Geräteposition im Raum, die Darstellung von virtuellen 3D-Inhalten und die Marker-Erkennung. Die einzelnen Komponenten werden in einer gemeinsamen High Level API verwendet. Entwickler können ihre konkrete Anwendungsimplementierung auf dieser API aufsetzen.

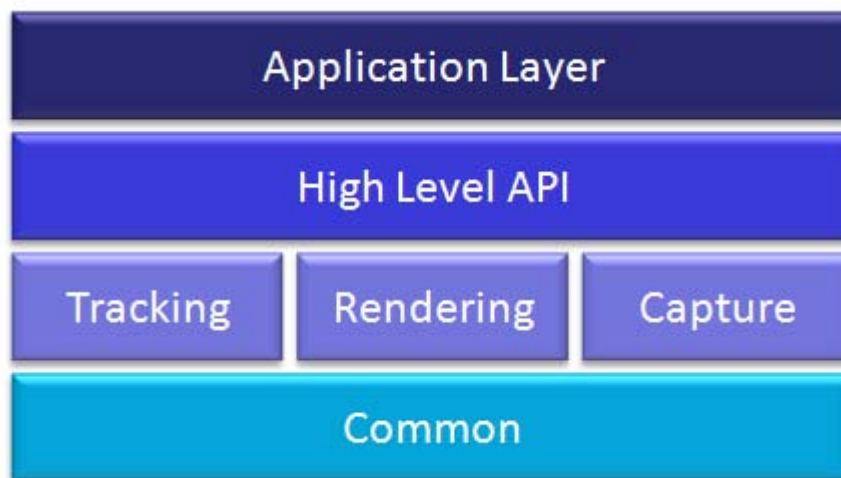


Abbildung 15: metaio Mobile SDK Modularchitektur (metaio GmbH 2012b, [\[http://docs.metaio.com/bin/view/Main/UnifeyeMobileSDK\]](http://docs.metaio.com/bin/view/Main/UnifeyeMobileSDK))

Sicht der Content Provider

Für Content Provider von Gebäudedaten, 3D-Modellen, Plänen und diversen Grafiken ist hauptsächlich interessant ob ihre Arbeit durch den Einsatz von AR-Visualisierungen beeinflusst wird. Ein wichtiger Punkt dabei sind die akzeptierten Eingangsformate für Rohdaten. Gerade für 3D-Modelldaten gibt es eine Vielzahl an Datenformaten. Viele CAD und Modellierungsprogramme verwenden programmspezifische Formate. Darüber hinaus existieren aber auch einige Formate die bewusst für den Datenaustausch gedacht sind und für die auch Import-Funktionen in verschiedenen Programmen bestehen. Die analysierten AR-Frameworks verfolgen unterschiedliche Strategien beim Import von 3D-Daten. Das D’Fusion Mobile Framework bietet zum

Beispiel eigene Tools für den Datenaustausch mit bekannten 3D-Modellierungsprogrammen wie 3ds Max und Maya der Firma Autodesk. Das metaio Mobile SDK sieht hingegen standardmäßig nur den Import von Daten im MD2 und OBJ Format vor. Über das zur Verfügung gestellte Unity Plugin lassen sich, ähnlich wie auch im Vuforia Augmented Reality SDK, Daten in vielen anderen Formaten verwenden. Bei der Wahl eines AR-Frameworks sollten frühzeitig die unterstützten Importformate beachtet werden und mit möglicherweise betroffenen Content Providern abgeklärt werden.

Zu den Grenzen in der Darstellungsleistung gibt es nur wenige Informationen von Seiten der Framework Entwickler. Bei einer Vielzahl von möglichen Zielgeräten, lassen sich darüber aber auch verständlicherweise nur schwer Aussagen treffen. Gerade die Größe des Grafik- und Arbeitsspeichers sind eine wichtige Information darüber, wie viele Polygone gleichzeitig dargestellt werden können. Ob dann auch noch grafische Transformationen flüssig wiedergegeben werden können hängt stark von der Rechenleistung des jeweiligen Geräts ab. Für Content Provider und Entwickler von Visualisierungsalgorithmen wäre es daher wichtig die Grenzen der Zielgeräte durch technische Prototypen zu bestimmen.

Die Art der Datenlieferung kann für Content Provider einen wichtigen Faktor darstellen. Ein direkter Zugriff auf das Visualisierungssystem kann einerseits einen Mehraufwand bedeuten, andererseits aber auch die Flexibilität insgesamt verbessern. Die Verantwortlichkeit für die Funktionalität der AR-App, welche beim Entwickler liegt, kann somit besser getrennt werden von den angezeigten Daten. Dafür muss natürlich auch von Seiten der Datenlieferanten eine gewisse Bereitschaft vorhanden sein. Die Möglichkeit den Unity Editor zur Erstellung von 3D-Szenen für AR-Applikationen zu verwenden wurde bereits erwähnt. Frameworks die ein solches Plugin zur Verfügung stellen, können den Entwickler in der Trennung von Verantwortlichkeiten unterstützen. Durch Import-Funktionen für viele verschiedene 3D-Formate und eine umfangreiche grafische Oberfläche zur Arbeit mit 3D-Daten kann Unity auch einen Zugang für Zulieferer ohne weitere Programmierkenntnisse bieten. Alternativ gibt es natürlich auch programmiertechnische Möglichkeiten die Datenlieferung zu optimieren. Daten könnten etwa über Webinterfaces oder FTP-Zugang geladen und auch analysiert werden. Dies stellt aber wahrscheinlich in den meisten Fällen einen Mehraufwand auf Seiten des Entwicklers dar.

4.5. Fazit

Die Evaluierung der Umsetzungsmöglichkeiten von AR-Applikationen zum Zweck der Architekturvisualisierung zeigt, dass eine Vielzahl unterschiedlicher Augmented Reality Frameworks existiert die bereits zum Großteil für die mobile Nutzung entwickelt wurden. Große Unterschiede bestehen vor allem zwischen proprietären Lösungen die von Firmen entwickelt und vertrieben werden, und quelloffenen Plattformen und Bibliotheken die häufig aus Forschungsgründen entstehen. Für den Anwendungsfall der Architekturvisualisierung bieten kommerzielle Lösungen, ganz unabhängig vom weiteren Vertrieb und der Lizenzierung daraus entstehender Applikationen, einige hilfreiche Funktionalitäten. Durch die Unterstützung einiger Frameworks für alternative Entwicklungswege können auch Personen ohne umfangreiche Programmiererfahrung an der Entwicklung von

AR-Systemen mitwirken. Durch den Einsatz des Unity Editors entstehen zum Beispiel eher Arbeitsprozesse die aus dem Design von Computerspielen bekannt sind als in der Entwicklung mobiler Applikationen. Ein weiteres wichtiges Feature einiger proprietärer Frameworks sind umfassende Möglichkeiten des Trackings. Neben ersten Entwicklungen hin zu Tracking ohne visuelle Marker, sondern durch Erkennung von Objekten der realen Umgebung, gibt es auch im Bereich des marker-basierten Trackings Fortschritte. Frameworks wie das Vuforia Augmented Reality SDK und das metaio Mobile SDK können beispielsweise beliebige 2D-Grafiken erkennen und tracken, wobei sich natürlich nicht alle Grafiken in gleichem Maße eignen. Eine genauere Erklärung sowie Beispiele für gut und schlecht geeignete Grafiken befinden sich in Kapitel 5.2. Tracking-Marker können auf jeden Fall durch die Verwendung von Postern, Gemälden und kontrastreichen Strukturen unauffälliger in die reale Umgebung integriert werden. Die größeren proprietären Frameworks sind normalerweise modular aufgebaut, damit auch nur die Komponenten geladen werden müssen, die in einer Applikation auch wirklich verwendet werden. Trotzdem kann gerade bei sehr speziellen Anforderungen der Einstieg mit einem kompakteren, oft quelloffenen Framework sinnvoller sein. Der Einsatz „leichterer“ Frameworks mit beschränkter Funktionalität kann zwar zu einer besseren Performanz beitragen, allerdings müssen fehlende Funktionalitäten aufwendiger integriert werden. Für Entwickler im Bereich der Architekturdarstellung gilt es also genau zu planen welche Funktionalitäten benötigt werden. Aus dieser Planung lässt sich anschließend einfacher ableiten ob ein umfangreiches proprietäres Framework unbedingt benötigt wird oder auch quelloffene Frameworks mit geringen Anpassungen dem Anwendungszweck gerecht werden können.

Die derzeit verfügbaren mobilen Geräte, wie Smartphones und Tablets, bieten genügend Leistung für die Darstellung architektonischer Modelle. Eine hilfreiche Übersicht zur Leistungsfähigkeit mobiler GPU Systeme bietet der NenaMark2 Benchmark. Dabei wird die Darstellungsgeschwindigkeit einer aus durchschnittlich 45.000 Dreiecken bestehenden 3D-Szene als Vergleichswert für die GPU-Leistung herangezogen. Das Samsung Galaxy S III erreicht in diesem Test zum Beispiel 56 fps (Frames per Second) (vgl. Nena Innovations AB o.J., [<http://nena.se/nenamark/view?version=2>]). Die Qualität von stundenlang berechneten Visualisierungen aus 3D-Modellierungsprogrammen mit eigens entwickelten Systemen für die Darstellung von Lichtszenarien und Materialeffekten kann zwar noch nicht live-transformierbar wiedergegeben werden, allerdings zeigt die Entwicklungsgeschwindigkeit der Rechenleistung bei mobilen Geräten, dass auch der Bereich von fotorealistischen Darstellungen in naher bis mittlerer Zukunft erreicht werden könnte. Jedenfalls kann auch schon mit häutiger Hardware ein sehr guter räumlicher Eindruck von 3D-Daten vermittelt werden, wie auch an der in Kapitel 5 beschriebenen prototypischen Entwicklung einer AR-Anwendung zu erkennen ist.

5. Implementierung einer Architekturvisualisierungs-Applikation

5.1. Einleitung

Im Rahmen der Forschung nach Umsetzungsmöglichkeiten für mobile Augmented Reality Applikationen im Bereich der Architekturvisualisierung wurde neben der Durchführung der anforderungsbasierten Evaluierung auch eine Prototyp-Applikation entwickelt. In diesem Kapitel werden der Entstehungsprozess sowie die implementierten Funktionalitäten der mobilen Anwendung vorgestellt. Weiters soll auch ein Überblick über die Entscheidungsgrundlagen zur Wahl der verwendeten Entwicklungsumgebung gegeben werden. Die während der Entwicklung entstandenen Erfahrungen, etwa zu technischen Grenzen und dem Umgang mit Problemen werden schlussendlich in Abschnitt 5.4 behandelt.

Die Applikation *Furniture Visualizer* soll prototypisch einen möglichen Anwendungsfall für AR-Technologie im Bereich der Architektur demonstrieren. In der Innenarchitektur und der Raumgestaltung wäre es oftmals von Vorteil Haushaltstechnik wie etwa Heizung, Kühlung, Küchengeräte und besonders Einrichtungsgegenstände an ihren zukünftig geplanten Positionen sehen zu können, um Größenverhältnisse und Farbkombinationen besser beurteilen zu können. Große Möbelhäuser bieten zu einem ähnlichen Zweck oft Schauräume an, um ihre Artikel in ihrem angedachten Umfeld präsentieren zu können. Die *Furniture Visualizer* App bietet durch Augmented Reality Technologie genau für diesen Anwendungsfall eine mobile, unkomplizierte Lösung. Möbelstücke können dadurch in einem beliebigen Umfeld dargestellt werden. Mithilfe der Überlagerung virtueller 3D-Objekte auf eine reale Umgebung kann ein direkter, visueller Eindruck über geplante Änderungen in der Raumgestaltung gewonnen werden. Als Basis für mögliche Erweiterungen könnte die AR-Applikation um eine Dokumentationsfunktion per Foto oder Video ergänzt werden, um die virtuell veränderte Umgebung auch längerfristig festhalten zu können. Der entwickelte Prototyp sollte aber nicht als fertiges Produkt verstanden werden, vielmehr sollten vor allem die Möglichkeiten der verwendeten AR-Technologie anhand einiger Beispielmodelle aufgezeigt werden. Sämtliche Projektdateien und die kompilierte Applikation sind verfügbar unter: http://mt091100.students.fhstp.ac.at/bakk2/furniture_visualizer/

5.2. Technische Umsetzung

Zur technischen Umsetzung des Prototyps wurde das Vuforia Augmented Reality SDK in Verbindung mit der Unity 3 IDE eingesetzt. Qualcomm, der Entwickler des Vuforia Frameworks bietet sein SDK (Software Development Kit) für die native Applikationsentwicklung unter Android und iOS an. Außerdem ist ein Plugin zur Softwareentwicklung mithilfe des bereits erwähnten Unity Editors verfügbar. Um das Unity Plugin verwenden zu können wird Unity in der Version 3 benötigt. Für die Entwicklung der *Furniture Visualizer* App wurde die Unity Free Version verwendet. Da die App für das mobile Betriebssystem Android entwickelt wurde, musste zusätzlich das Unity Android Plugin verwendet werden. Dieses Plugin ermöglicht erst das Deployment der Applikation auf entsprechende mobile Geräte. In der folgenden Liste sind alle verwendeten Tools für die Unity Entwicklung mithilfe des Vuforia Augmented Reality SDK zusammengefasst:



1. Java Development Kit (JDK)
2. Android Development Kit
3. Unity IDE
4. Unity Android Plugin
5. Unity Vuforia Extension

Um Android Geräte während der Prototyp Entwicklung für Debugging Zwecke einsetzen zu können mussten diese immer erst manuell installiert werden. Die benötigten Treiber werden oftmals von den jeweiligen Geräteherstellern zum Download angeboten. Ist dies nicht der Fall kann der mitgelieferte USB-Treiber des Android Development Kit eingesetzt werden. Damit das direkte Kompilieren des Unity Projekts auf das Android Testgerät funktioniert, muss am Gerät die Option USB-Debugging aktiviert sein. Nach dem Abschluss aller Vorbereitungsschritte konnte ein neues Projekt im Unity Editor angelegt werden. Nach dem Import der Vuforia Erweiterung wurde mit der eigentlichen Entwicklung der AR-Applikation begonnen. Die wichtigsten Funktionalitäten der Augmented Reality Darstellung sind durch das Vuforia Plugin einfach zu integrieren. Dieser rasche Einstieg in die Entwicklung von AR-Systemen war einer der ausschlaggebenden Gründe für die Entwicklung mithilfe des Vuforia Augmented Reality SDK. Gerade der Umgang mit 3D-Daten wird durch den Unity Editor stark vereinfacht. Wie auch in Kapitel 4.4 erwähnt bietet Unity Import-Funktionen für viele 3D-Datenformate sowie umfangreiche Möglichkeiten des Scriptings in den Sprachen JavaScript und C#. Abbildung 16 zeigt die Unity Projektstruktur der *Furniture Visualizer* Anwendung.

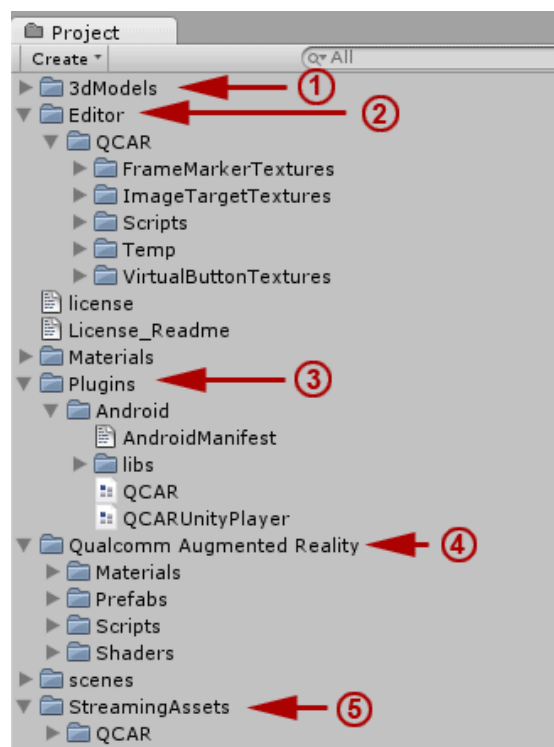


Abbildung 16: Furniture Visualizer Projektstruktur

1. **3dModels:** Der Ordner 3dModels enthält die 3D-Objekte des Projekts, also im Fall dieses Prototyps die Modelle der Einrichtungsgegenstände.
2. **Editor:** Dieser Ordner enthält die benötigten Scripts um dynamisch mit den Trackable Daten interagieren zu können (vgl. Qualcomm Incorporated o.J.b, [<https://ar.qualcomm.at/qdevnet/sdk/android/Downloads%20-%20Android>]). Auf Trackables bzw. deren Subkategorie ImageTargets, wird im weiteren Verlauf dieses Kapitels noch näher eingegangen.
3. **Plugins:** Der Plugins Ordner enthält Java- und native Programme die das Vuforia Augmented Reality SDK in die Unity Android Applikation integrieren (ebd.)
4. **Qualcomm Augmented Reality:** In diesem Order befinden sich die Prefabs (Unity Bausteine) und Skripte des Vuforia SDK die in jeder damit erstellten Unity AR-Applikation verwendet werden (ebd.). Auf die im *Furniture Visualizer* Projekt verwendeten Prefabs wird im weiteren Verlauf des Kapitels noch näher eingegangen.
5. **StreamingAssets:** Dieser Ordner enthält die heruntergeladenen Daten des online Target Management Systems (ebd.).

Neben der allgemeinen Projektstruktur gibt es in der Unity IDE auch eine Szenenhierarchie. Diese Ansicht liefert eine Übersicht über die verwendeten Prefab Instanzen einer Szene. Das *Furniture Visualizer* Projekt ist in einer Szene organisiert, eine Übersicht der darin verwendeten Prefabs ist auf Abbildung 17 zu sehen.

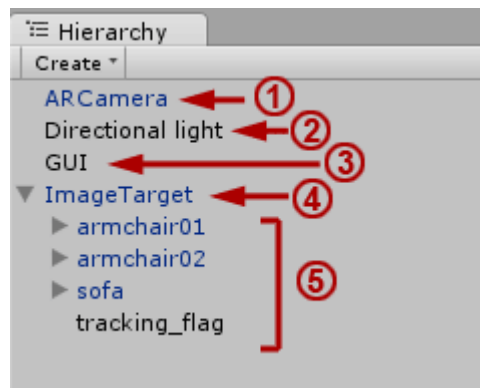


Abbildung 17: *Furniture Visualizer* Szenenhierarchie

1. **ARCamera:** Die ARCamera Prefab Instanz sorgt für die Darstellung des Live-Kamerabildes im Hintergrund der Augmented Reality Ansicht. Sie stellt die reale Umgebung dar, die in weiterer Folge um die virtuellen Einrichtungsgegenstände erweitert wird. Außerdem sorgt die ARCamera dafür, dass diese Szenenobjekte auf Positionstrackingdaten des mobilen Zielgerätes reagieren können. Weiters können in den ARCamera Einstellungen Pakete von Visuellen Markern (Data Sets) geladen und aktiviert werden. Nur hier aktivierte Marker können vom AR-System erkannt werden.

- 2. Directional Light:** Das Directional Light Objekt dient als Lichtquelle für die 3D-Objekte in der Szene. Schattenwurf und andere erweiterte Lichtoptionen wurden deaktiviert um eine möglichst neutrale und performante Beleuchtung zu erreichen.
- 3. GUI:** Dieses Objekt verwaltet das Graphical User Interface (GUI) des *Furniture Visualizer*. Es generiert alle Elemente der Anwendung die zur Interaktion mit dem Benutzer verwendet werden. Im Falle des Prototyps handelt es sich noch um ein einfaches Menü zur Auswahl von virtuellen Objekten (siehe Abbildung 18), das verwendete UnityGUI System bietet aber auch Möglichkeiten für die Entwicklung komplexerer Interfaces.



Abbildung 18: Screenshot *Furniture Visualizer* mit Auswahlmenü

Zur Erstellung des Menüs wurde eine leere `GameObject` Instanz in die Szenenhierarchie eingefügt und um ein Skript erweitert welches die eigentliche GUI-Logik durchführt. Der folgende Code-Teil zeigt die grundlegende Struktur des Menüs.

```
public class GUIMenu : MonoBehaviour {
    GameObject armchair01;
    GameObject armchair02;
    GameObject sofa;
    GameObject tracking_flag;
    //more declarations of constants and member variables..

    //the Awake() method is called after the game objects have been
    //initialized but before the application starts
    void Awake() {
        //get 3d-modells
        armchair01 = GameObject.Find("armchair01");
        armchair02 = GameObject.Find("armchair02");
        sofa = GameObject.Find("sofa");
    }
}
```



```

tracking_flag = GameObject.Find("tracking_flag");
BUTTON_WIDTH = (Screen.width/3)-WIDTH_SPACING;
}

//the OnGUI() method is called every frame to update the screen
//and respond to gui events
void OnGUI () {
    BUTTON_WIDTH = (Screen.width/3)-WIDTH_SPACING;
    // Make a background box
    GUI.Box(new Rect(SPACING,(Screen.height-110),
    (Screen.width-20),100), "Selection Menu");

    // Make the first button and respond to press events
    if(GUI.Button(new Rect(SPACING*2,(Screen.height-70),
    BUTTON_WIDTH,50), "armchair 01")) {
        tracking_flag.renderer.enabled = false;
        //toggle display of first 3d-modell
        Renderer[] renderers =
        armchair01.GetComponentsInChildren<Renderer>();
        foreach(Renderer render in renderers){
            render.enabled = !render.enabled;
        }
        armchair01.renderer.enabled =
        !armchair01.renderer.enabled;
    }
    //similar code for other buttons and 3d-modells
}
}
}

```

Unity Scripts können in JavaScript und C# programmiert werden. Für beide Sprachen sind dieselben Bibliotheken verfügbar. Für die Entwicklung des Prototyps wurde C# aufgrund der strengeren Typisierung bevorzugt. Diese Eigenschaft führte zu einem besseren Verständnis der neu kennengelernten Unity Library.

- 4. ImageTarget:** Die ImageTarget Instanz repräsentiert einen bestimmten visuellen Marker. Ein Vuforia Projekt kann mehrere visuelle Marker in sogenannten Data Set Paketen beinhalten. In den Eigenschaften des ImageTargets muss ein bestimmter Marker ausgewählt werden um die Registrierung virtueller Objekte zu ermöglichen. Visuelle Marker können nach der kostenlosen Registrierung eines Benutzerkontos auf der offiziellen Webseite des Vuforia Augmented Reality SDK verwaltet werden. Über das Tool lassen sich Marker erstellen und in Form von Data Sets für das eigene Projekt herunterladen. Das System analysiert hochgeladene Bilder auch in Hinsicht auf spätere Erkennbarkeit und liefert eine Einschätzung ob eine Verwendung als Registrierungsmarker sinnvoll ist. Dabei sind die wichtigsten Punkte: hoher Detailreichtum, hoher Kontrast und die Vermeidung gleichförmiger Muster. Abbildung 19 und 20 zeigen Beispiele für unterschiedlich gut erkennbare Marker-Bilder. Das ImageTarget enthält außerdem noch das DefaultTrackableEventHandler Script, mit dessen Hilfe man programmatisch auf das Erkennen und Verlieren der Marker-Registrierung reagieren kann. Die folgenden zwei Methoden zeigen die konkrete Implementierung dieser Einstiegspunkte.

```

//this method is called on ImageTarget registration
private void OnTrackingFound()
{
    GameObject tracking_flag = GameObject.Find("tracking_flag");

    //show placeholder object when marker was registered
}

```

```
tracking_flag.renderer.enabled = true;

Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + "
found");
}

//this method is called when visual connection is lost
private void OnTrackingLost()
{
    Renderer[] rendererComponents =
    GetComponentInChildren<Renderer>();

    // Disable rendering of 3d components:
    foreach (Renderer component in rendererComponents) {
        component.enabled = false;
    }

    Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + "
lost");
}
```

5. **3D-Daten:** Alle 3D-Objekte deren Darstellung an die Registrierung des ImageTarget gebunden werden sollen, werden hierarchisch darunter eingefügt. Im Fall des *Furniture Visualizer* handelt es sich hierbei um die Modelle der Einrichtungsgegenstände

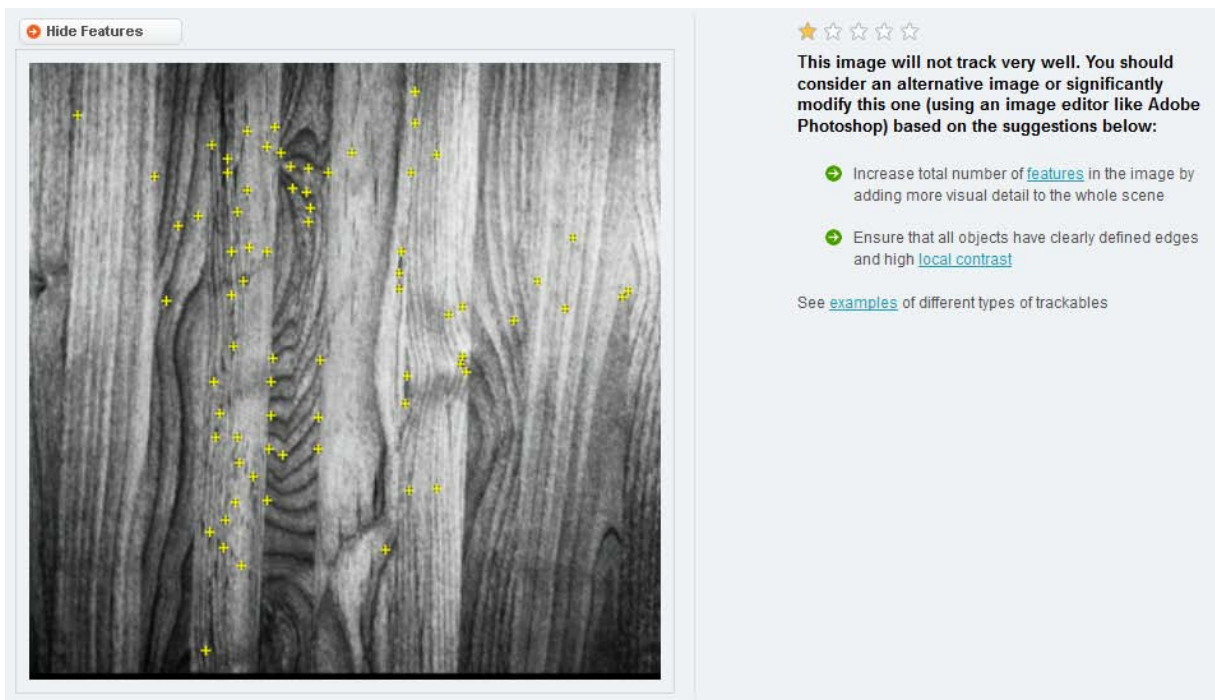


Abbildung 19: Ungeeignetes Bild für visuelles Tracking

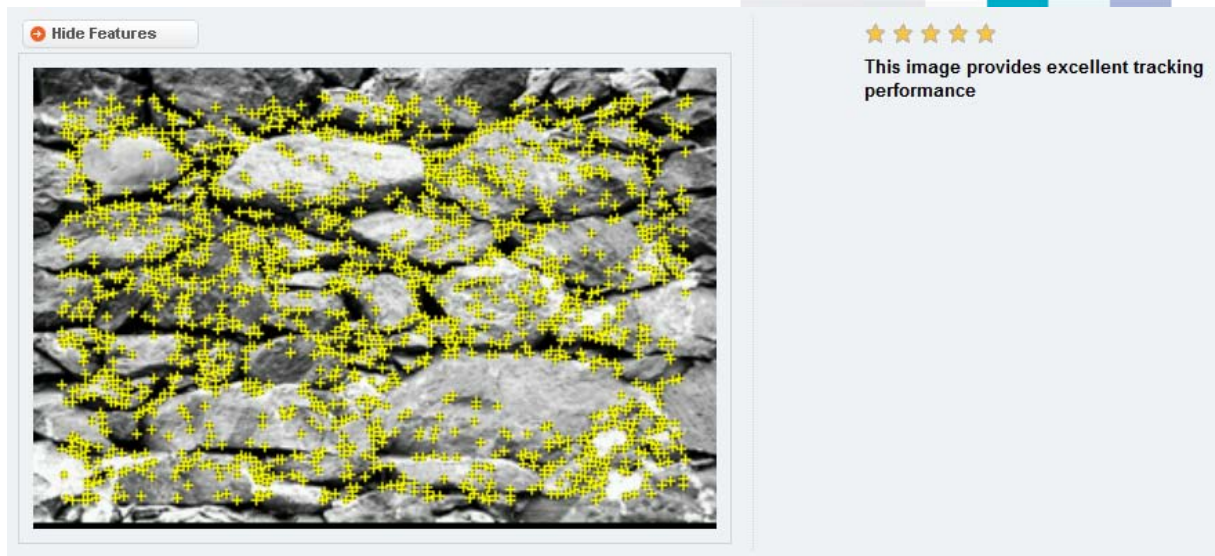


Abbildung 20: Optimal geeignetes Bild für visuelles Tracking

5.3. Vorstellung der Funktionalitäten

In diesem Kapitel sollen die Funktionalitäten des *Furniture Visualizer* demonstriert werden. Grundsätzlich können vordefinierte visuelle Marker im realen Umfeld erkannt werden und durch virtuelle Einrichtungsgegenstände überlagert werden. Durch die Überlagerung mittels Platzhalterobjekt wird dem Benutzer die Erkennung eines virtuellen Markers dargestellt. Der Benutzer erhält somit Feedback über die erfolgreiche Marker-Registrierung. Ein Screenshot der Applikation mit angezeigtem Platzhalterobjekt ist in Abbildung 21 zu sehen. Über das Auswahlménü am unteren Bildschirmrand kann der Benutzer nun Einrichtungsgegenstände zur Darstellung auswählen. Je nach Positionierung des Markers im Raum können die Gegenstände an verschiedenen Plätzen dargestellt werden. Abbildung 22 und 23 zeigen Screenshots mit Beispieldarstellungen. Durch Verwendung verschieden großer Marker können unterschiedliche Betrachtungsentfernungen unterstützt werden. Einrichtungsgegenstände lassen sich so auch in verschiedenen Maßstäben darstellen. Ein Beispiel ist in Abbildung 24 zu sehen. Da außerdem keine manuelle Kalibrierung und Konfiguration notwendig ist um 3D-Inhalte auf das reale Umfeld zu überlagern, kann die Anwendung sehr einfach, schnell und flexibel eingesetzt werden.



Abbildung 21: Screenshot mit Platzhalterobjekt



Abbildung 22: Screenshot mit virtuellem Stuhl 1

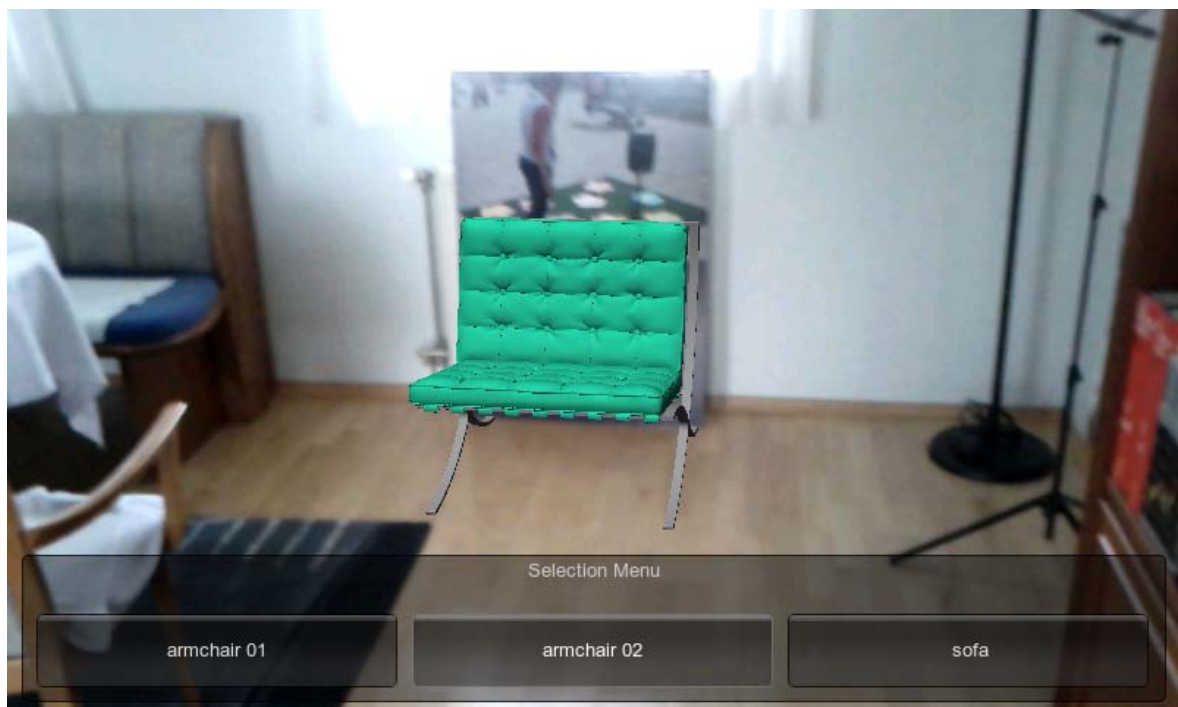


Abbildung 23: Screenshot mit virtuellem Stuhl 2

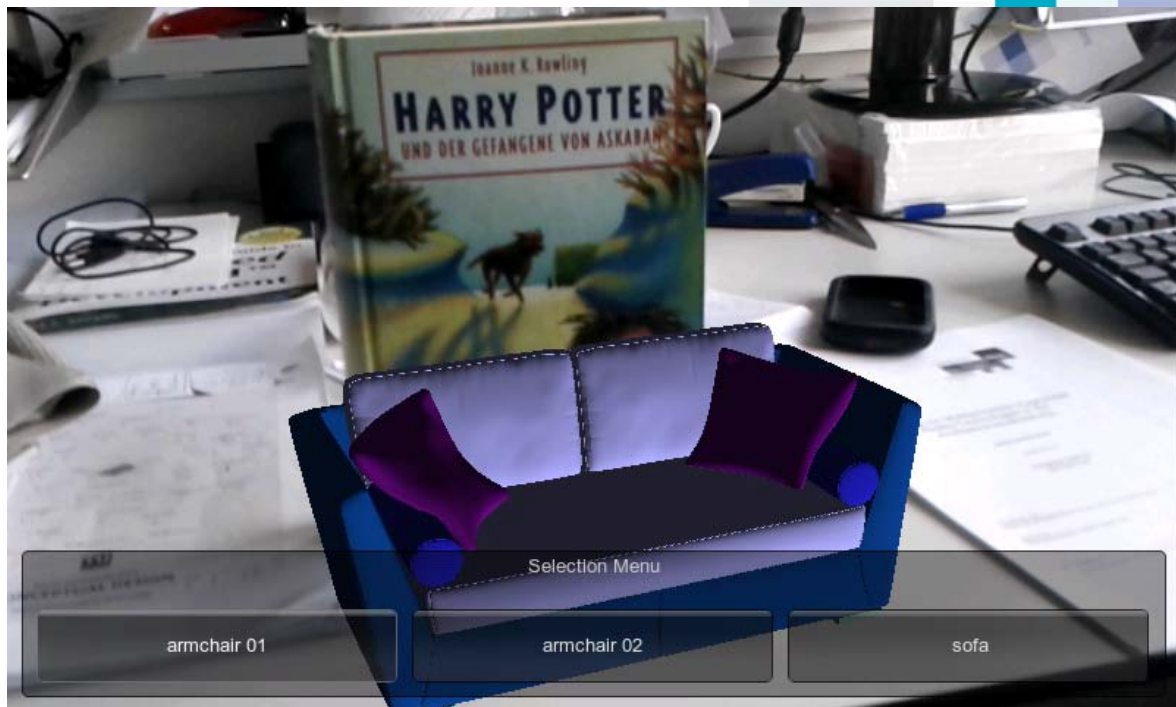


Abbildung 24: Screenshot mit verändertem Modellmaßstab

5.4. Erkenntnisse aus der Anwendungsentwicklung

Durch die Entwicklung des *Furniture Visualizer* Prototyps konnten einige Erkenntnisse über die Entwicklung von mobilen AR-Applikationen gewonnen werden. Wie Kapitel 5.3 veranschaulicht konnte die Implementierung des Prototyps mit interessanten Ergebnissen abgeschlossen werden. Im Umgang mit der fertiggestellten Applikation zeigen sich die weitreichenden Möglichkeiten der mobilen Augmented Reality Technologie. Speziell für den in Abschnitt 5.1 beschriebenen Anwendungsfall ergibt sich eine hohe potentielle Nutzbarkeit. Die folgende Auflistung gibt eine Übersicht über die wichtigsten Erfahrungen aus der Anwendungsentwicklung.

- **Marker Qualität:** Die Qualität, und damit die Erkennbarkeit der verwendeten visuellen Marker ist ein entscheidender Faktor für eine möglichst geringe Fehleranfälligkeit. Wie in Kapitel 5.2 beschrieben ist nicht jede Grafik gleich gut registrierbar durch das AR-System. Weniger geeignete Marker führen somit zu häufigerem Ausfall der Anzeige der überlagerten Objekte, speziell unter nicht optimalen Lichtverhältnissen.
- **Trackingfehler:** Auch unter guten Lichtverhältnissen, und der Verwendung gut geeigneter Marker kann es zu Trackingfehlern kommen. Die Häufigkeit ist in diesem Fall jedoch deutlich geringer als bei qualitativ minderwertigen Markern. Objekte werden in manchen Fällen falsch auf den Markern positioniert. Durch Positionsänderungen des mobilen Geräts oder des Markers verschwinden diese Fehler im Normalfall wieder. Durch die inkorrekte Positionierung der virtuellen Objekte in der realen Umgebung wird allerdings der Eindruck der Verschmelzung von realen und virtuellen Inhalten geschmälert.

- **Geschwindigkeit:** Wie schon in der Auflistung derzeitiger Problemfelder der AR-Technologie in Kapitel 2.3 angeführt, wirken sich zu langsame Bildwiederholraten störend auf die Benutzung von Augmented Reality Applikationen aus. Dieses Problem wurde auch im *Furniture Visualizer* Prototyp sichtbar. Sehr rasche Bewegungen des mobilen Geräts führen in der Applikation nicht nur zu einem verwischten Eindruck des Live-Kamerabildes sondern auch zu häufigem Verlust der Marker-Registrierung.
- **GUI Entwicklung:** Der Punkt GUI-Entwicklung hängt speziell mit dem Einsatz der Unity IDE in der Entwicklung zusammen. Das UnityGUI System, welches zur Implementierung des Auswahlmenüs im Prototyp eingesetzt wurde verfügt über andere Interaktionselemente und –muster als es bei der nativen Programmierung mit dem Android SDK der Fall ist. Trotz seiner Mächtigkeit kann dies in der Entwicklung von nativen, mobilen Applikationen einen Nachteil darstellen. Wie in Kapitel 2.3 erwähnt sind Benutzer mobiler Apps gewisse GUI-Patterns gewohnt, ein Einsatz ähnlicher Muster in AR-Applikationen könnte daher Einstiegsschwierigkeiten für unerfahrene Anwender reduzieren.
- **Debugging:** Entwicklern nativer Android Applikationen steht mit dem Dalvik Debug Monitor Server (DDMS) ein mächtiges Debugging Tool zur Verfügung. Dessen Integration in die häufig verwendete Eclipse IDE erleichtert die zeitnahe Fehlerbehebung erheblich. Viele Informationen über den aktuellen Gerätezustand und Speicherverbrauch lassen Rückschlüsse auf mögliche Fehlerquellen zu. Das DDMS Tool kann zwar auch unabhängig von der Eclipse IDE verwendet werden, allerdings verläuft das Debugging mit Unity durch längere Kompilierungszeiten oft schwerfälliger.
- **Hardware Voraussetzungen:** Die Unity Entwicklungsumgebung und das Vuforia SDK können nur für Geräte eingesetzt werden die, die Systemanforderungen beider Plattformen erfüllen. Dazu ist Android ab Version 2.2 (Froyo) und zumindest eine ARMv7 CPU notwendig. Die GPU sollte aus Performancegründen OpenGL ES in der Version 2.0 unterstützen (vgl. Qualcomm Incorporated 2012, [https://ar.qualcomm.at/qdevnet/support_file/download/Android/android%20supported%20devices/Supported_Devices_Android.pdf]; Unity Technologies o.J., [<https://store.unity3d.com/>]). Während der *Furniture Visualizer* Entwicklung konnte etwa ein älteres Android Tablet (Samsung Galaxy Tab) nicht als Testgerät verwendet werden.



6. Fazit

Die Evaluierung der aktuell verfügbaren Umsetzungsmöglichkeiten von mobilen AR-Anwendungen sowie die Implementierung eines konkreten Prototyps führten zu zahlreichen Erkenntnissen über die derzeitigen Möglichkeiten mobiler Augmented Reality Technologie. Die in Kapitel 4 beschriebene, anforderungsbasierte Analyse von AR Frameworks und Libraries liefert einen Überblick über aktuell verfügbare Funktionalitäten und deren Nutzbarkeit für Anwender, Entwickler und Content Provider. Hier zeigten sich große Unterschiede zwischen umfangreichen proprietären Lösungen und quelloffenen Systemen. Während Open-Source Frameworks häufig einen bestimmten Teilbereich einer AR-Anwendung, wie etwa das Tracking, sehr gut implementieren, müssen andere Bereiche wie das 3D-Rendering selbst entwickelt werden. Kommerzielle Produkte bieten zur Umsetzung eigener Anwendungen meist einen größeren Funktionsumfang und durch diverse Erweiterungen eine hohe Flexibilität für den Entwickler. Natürlich können für diesen Komfort auch hohe Kosten anfallen, sodass eine Abschätzung über benötigte Funktionalitäten sinnvollerweise vor Projektbeginn durchgeführt werden sollte.

In Kapitel 5 wird die Eigenentwicklung des AR-Prototyps *Furniture Visualizer* beschrieben. Aufgrund eines Anwendungsfalls für Augmented Reality Technologie aus dem Bereich der Innenarchitektur, wird hier die konkrete Umsetzung einer entsprechenden mobilen Applikation dargestellt. Aus dem Funktionsumfang der fertiggestellten Anwendung und den festgestellten Problemen konnten einige Erkenntnisse über den Einsatz von AR-Technologie in der Architekturvisualisierung gewonnen werden. Es ist beispielsweise überhaupt keine manuelle Kalibrierung und Konfiguration notwendig um virtuelle Gegenstände in einem realen Umfeld anzuzeigen. Daraus resultiert eine sehr flexible Nutzbarkeit in unterschiedlichen Umgebungen. Der Darstellungsmaßstab kann so durch Änderungen der Markergrößen beeinflusst werden. Aufgrund der technischen Grenzen derzeitiger Smartphones und Tablets wird die mobile Augmented Reality Technologie, die unterschiedlichen, gängigen Formate der Architekturvisualisierung nicht vollends ersetzen, aber funktional und medial erweitern können. Einige Vorteile dieser relativ jungen Darstellungstechnologie, könnten sie zu einer interessanten Ergänzung der klassischen Visualisierungsformen werden lassen. Gerade für Anwendungsfälle in denen mobile Verfügbarkeit und ein realer Kontext eine wichtige Rolle spielen, können AR-Applikationen sinnvoll eingesetzt werden. Außerdem bieten die Freiheiten des 6DOF Tracking eine einfache und direkte Form räumliche Verhältnisse darzustellen, was in vielen Bereichen der Architektur einen Mehrwert bieten kann. Die Screenshots des *Furniture Visualizer* in Kapitel 5.3 vermitteln bereits einen Eindruck über derartige Vorteile. In Verbindung mit den aufgezeigten, allgemeinen Beweggründen für Augmented Reality im Architekturbereich lassen sich somit auch weitere potentielle Anwendungsszenarien ausmachen. Denkbar wären etwa auch Visualisierungen im Bereich der Planung von Gebäuden und Gebäudekomplexen.

Die in den letzten Jahren rasant angestiegene Rechenleistung mobiler CPUs lässt auch für die nahe Zukunft eine ähnliche Entwicklung vermuten. Für die mobile AR-Technologie könnte dies in Verbindung mit verbesserten Kamerasensoren und Optiken zu schnelleren, leistungsfähigeren Darstellungen von 3D-Objekten führen. Betrachtet man zusätzlich die hohen Absatzzahlen am Smartphone und Tablet Markt, scheinen mobile Augmented Reality Lösungen in Zukunft ein verstärktes Anwendungspotential erlangen zu können. Diese Umstände lassen sie außerdem prädestiniert erscheinen für den Einsatz im Architekturbereich.



Literaturverzeichnis

- Apple Inc. (2012). ARBasketball - Augmented Reality Basketball Game. [<http://itunes.apple.com/us/app/arbasketball/id393333529>] (ausgehoben 28.05.2012).
- Azuma, R. / Baillo, Y. / Behringer R. / Feiner S. / Julier S. / MacIntyre B. (2001). Recent advances in augmented reality. In: Computer Graphics and Applications, IEEE 21. Jg., H. 6, S. 34-47.
- Beuchert, P. (2010). Augmented Reality in der Architektur. Unveröffentlichte Diplomarbeit, Karlsruher Institut für Technologie.
- BlueLime Studio Inc. (2008). Architectural Rendering, Savannah River Landing, 900 E. broughton St, Savannah GA. [<http://commons.wikimedia.org/wiki/File:Bluelime002.jpg>] (ausgehoben 28.05.2012).
- Christian Doppler Laboratory for Handheld Augmented Reality - TU Graz (2011). Studierstube Tracker. [<http://handheldar.icg.tugraz.at/stbtracker.php>] (ausgehoben 30.05.2012).
- Colborne, G. (2010). Simple and Usable Web, Mobile, and Interaction Design. Berkeley, California, United States of America: New Riders Press.
- Domhan, T. (o.J.). AndAR - Android Augmented Reality. [<http://code.google.com/p/andar/>] (ausgehoben 30.05.2012)
- Biocca, F. A. / Rolland, J. P. (o.J.). Virtual Eyes Can Rearrange Your Body: Adaptation to Visual Displacement In See-Through Head-Mounted Displays. [<http://mindlab.msu.edu/presence/presence.html>] (ausgehoben 02.06.2012)
- Hincapié, M. / Caponio, A. / Rios, H. / Mendivil, E. (2011). An Introduction to Augmented Reality with Applications. In: 13th International Conference on Transparent Optical Networks (ICTON), 2011, S. 1-4.
- HITLab (o.J.). ARToolKit Home Page. [<http://www.hitl.washington.edu/artoolkit/>] (ausgehoben 30.05.2012)
- Höhl, W. (2006). Augmented Reality (AR) Welche Potentiale gibt es für die Architekturvisualisierung? In: Computer Spezial 1/2006, S. 12-15.
- IDC Corporate USA (2012). Android- and iOS-Powered Smartphones Expand Their Share of the Market in the First Quarter, According to IDC. [<http://www.idc.com/getdoc.jsp?containerId=prUS23503312>] (ausgehoben 27.05.2012)
- Mullet, K. / Sano, D. (1994). Designing Visual Interfaces: Communication Oriented Techniques. Upper Saddle River, New Jersey, United States of America: Prentice Hall PTR.
- Leibl, M. (o.J.). Architekturmodellbau für Wettbewerbe und Präsentationen. [<http://www.architekturmodelle.com/architekturmodelle/wettbewerbsmodelle.html>] (ausgehoben 31.05.2012)
- metaio GmbH (2012a). metaio Mobile SDK. [<http://www.metaio.com/software/mobile-sdk/>] (ausgehoben 30.05.2012)
- metaio GmbH (2012b). UnifeyeMobileSDK. [<http://docs.metaio.com/bin/view/Main/UnifeyeMobileSDK>] (ausgehoben 31.05.2012)
- Nena Innovation AB (o.J.). NenaMark2 results. [<http://nena.se/nenamark/view?version=2>] (asugehoben 02.06.2012)
- Peer GmbH (2010). mixare Free Open Source Augmented Reality Engine. [<http://www.mixare.org/>] (ausgehoben 30.05.2012)

Qualcomm Incorporated (2012). Android Supported Devices.

[https://ar.qualcomm.at/qdevnet/support_file/download/Android/android%20supported%20devices/Supported_Devices_Android.pdf] (ausgehoben 02.06.2012)

Qualcomm Incorporated (o.J.a). Augmented Reality (Vuforia) - Qualcomm Developer Network.

[<http://developer.qualcomm.com/mobile-development/mobile-technologies/augmented-reality>] (ausgehoben 30.05.2012)

Qualcomm Incorporated (o.J.b). SDK | Qualcomm Augmented Reality.

[<https://ar.qualcomm.at/qdevnet/sdk/android/Downloads%20-%20Android>] (ausgehoben 01.06.2012)

Sakr, S. (2012). Engadget - Those suave Google glasses are now patent-protected.

[<http://www.engadget.com/2012/05/15/google-glasses-design-patent/>] (ausgehoben 25.05.2012)

Salzburg Research (o.J.). Peak.AR. [<https://peakar.salzburgresearch.at>] (ausgehoben 27.05.2012)

Steiner, S. (2011). User Interface Design for Media Remote Control Applications on Mobile Touch Devices. Unveröffentlichte Bachelorarbeit, FH St. Pölten.

Tarovik, K. (o.J.). simiotica. [<http://simiotica.com/index.php>] (ausgehoben 28.05.2012)

Total Immersion (o.J.). D'Fusion Mobile | Augmented Reality Software and Solutions by Total Immersion.

[<http://www.t-immersion.com/products/dfusion-suite/dfusion-mobile>] (ausgehoben 30.05.2012)

Total Immersion (2011). Getting Started - Total Immersion Community Website. [<https://community.t-immersion.com/getting-started>] (ausgehoben 31.05.2012)

Unity Technologies (o.J.). UNITY: Store. [<https://store.unity3d.com/>] (ausgehoben 02.06.2012)

Wagner, D. (2007). Handheld Augmented Reality. Unveröffentlichte Dissertation, Graz University of Technology.

Wikitude GmbH (o.J.a). The Wikitude Augmented Reality SDK. [<http://www.wikitude.com/developer/wikitude-augmented-reality-for-developers>] (ausgehoben 30.05.2012)

Wikitude GmbH (o.J.b). Wikitude World Browser. [<http://www.wikitude.com/tour/wikitude-world-browser>] (ausgehoben 26. 05 2012)



Abbildungsverzeichnis

Abbildung 1: HWD Prototyp.....	5
Abbildung 2: Mehrere Marker in einer AR Anwendung.....	7
Abbildung 3: Google Project Glass.....	8
Abbildung 4: Wikitude Screenshot.....	10
Abbildung 5: Peak.AR Überlagerung von virtuellen Daten.....	12
Abbildung 6: ARBasketball Marker.....	13
Abbildung 7: ARBasketball Spielszene	13
Abbildung 8: Physisches Architekturmodell	16
Abbildung 9: Architektur 3D-Visualisierung	17
Abbildung 10: SnapShop Showroom Interface.....	20
Abbildung 11: SnapShop Showroom „AR“ Montage.....	20
Abbildung 12: 3DOn ARchitecture Visualisierung eines 3D-Modells	21
Abbildung 13: Studierstube Tracker – gleichzeitige Erkennung einer hohen Anzahl Marker	28
Abbildung 14: D’Fusion Mobile SDK Komponenten-Integration für Android und iOS.....	30
Abbildung 15: metaio Mobile SDK Modulararchitektur	31
Abbildung 16: Furniture Visualizer Projektstruktur	35
Abbildung 17: Furniture Visualizer Szenenhierarchie	36
Abbildung 18: Screenshot Furniture Visualizer mit Auswahlmenü.....	37
Abbildung 19: Ungeeignetes Bild für visuelles Tracking	39
Abbildung 20: Optimal geeignetes Bild für visuelles Tracking	40
Abbildung 21: Screenshot mit Platzhalterobjekt.....	41
Abbildung 22: Screenshot mit virtuellem Stuhl 1.....	41
Abbildung 23: Screenshot mit virtuellem Stuhl 2.....	41
Abbildung 24: Screenshot mit verändertem Modellmaßstab.....	42



Tabellenverzeichnis

Tabelle 1: Marktanteile mobile Betriebssysteme 1. Quartal 2012..... 11