Themensteller: Prof. Dr. Oliver Thomas

Betreuer: Dirk Metzger, M.Sc. with Honors

Vorgelegt von: Jannik Hoffjann

Jahnplatz 6 W-169

49080 Osnabrück

Matrikelnummer: 945592  
E-Mail-Adresse: jhoffjann@uni-osnabrueck.de

Abgabetermin: 2015-01-22

Themensteller: Prof. Dr. Oliver Thomas

Betreuer: Vorname Name

Vorgelegt von: Max Mustermann  
Semesteranschrift  
PLZ Wohnort

Matrikelnummer: 00000000  
E-Mail-Adresse: mustermann@uni-osnabrueck.de

Abgabetermin: JJJJ-MM-TT

Einblendung von kontextsensitiven Inhalten auf der Google Glass

Bachelorarbeit

am Fachgebiet Informationsmanagement und Wirtschaftsinformatik,  
Universität Osnabrück

zur Erlangung des Grades

Bachelor of Science (B. Sc.)

im Studiengang

Wirtschaftsinformatik

# Zusammenfassung / Expose

In der Arbeit sollen die Möglichkeiten der Einblendung von kontextsensitiven Inhalten auf einem in das Sichtfeld integrierten Gerät am Beispiel der Google Glass erprobt werden. Dabei sollen nach Einführung und Vorstellung des Geräts und der mitgelieferten Software verschiedene Möglichkeiten der Kontextsensitivität (engl. Context-Awareness) erörtert werden und ihren Nutzbarkeit auf dem Bereich der tragbaren Geräte erfragt werden.

Es sollen die verschiedenen Möglichkeiten der Context-Awareness erläutert werden und dabei versucht werden die Anforderungen der Einzelnen mit den Möglichkeiten der Google Glass abzugleichen. Beispielhaft sollen diese Möglichkeiten an bereits erhältlichen Applikationen erläutert werden. Durch sorgfältige Auswahl sollen die verschiedenen Arten dargelegt werden um so eine Auswahl der am besten Geeigneten zu ermöglichen.

Nach Abwägung der einzelnen Möglichkeiten soll eine der Arten der kontextsensitiven Inhaltsgewinnung beispielhaft auf der Google Glass implementiert werden. Es soll getestet werden inwieweit sich das Medium Google Glass als agierendes Objekt eignet und wo durch gegebene Hard- und Software eventuelle Grenzen entstehen.

Vorstellbar wäre an dieser Stelle zum Beispiel die Implementation einer mobilen Applikation auf Grundlage von Open CV (opencv dev team) und eine der implementierten Keypointerkennungen wie zum Beispiel SURF (Bay et al. 2006), FREAK (Alahi et al. 2012) und BRISK (Leutenegger et al. 2011). Diese bieten durch vielfältige Möglichkeiten des Matchings, Möglichkeiten der Wiedererkennung und Auswertung von Grafiken, welche die Umsetzung einer kontextsensitiven Anwendung auf der Google Glass ermöglichen könnten.

Durch diese abschließende Implementation und eine Auswertung der Ergebnisse soll ein erster Versuch der Einblendung von kontextsensitiven Inhalten auf der Google Glass erbracht werden und Möglichkeiten zu weiteren Nutzung des Geräts aufgezeigt werden.

Inhaltsverzeichnis

Zusammenfassung / Expose II

Abbildungsverzeichnis IV

Tabellenverzeichnis V

Abkürzungsverzeichnis VI

Symbolverzeichnis VII

1 Einleitung / Motivation 1

2 Kontextsensitivität 2

2.1 Definition 2

2.2 Möglichkeiten der Kontextsensitivität 3

2.2.1 Marker / QR Codes 4

2.2.2 Location-Based Services 6

2.2.3 Objekt- und Bilderkennung 7

3 Google Glass 8

3.1 Die Google Glass als Vertreter der Augmented Reality 8

3.2 Spezifikationen und Besonderheiten der Google Glass 8

3.2.1 Hardwarespezifikationen 8

3.2.2 Softwarespezifikationen 8

4 Einblendung von kontextsensitiven Inhalten auf der Google Glass 9

4.1 Idee und Funktionsweise der kontextsensitiven Applikation 9

4.2 Vorstellung von OpenCV und der verwendeten Algorithmen 10

4.2.1 OpenCV 10

4.2.2 SURF & Nearest Neighbor Matching 10

5 Umsetzung einer kontextsensitiven Applikation mit OpenCV 11

5.1 Vorstellung der Implementation / ausgewählter Programmteile 11

5.2 Fallstudie / Auswertung der Applikation 11

6 Fazit und Ausblick 12

Literaturverzeichnis 13

7 Anhang 16

A Unterkapitel des Anhangs 16

B Zweites Unterkapitel des Anhangs 16

# Abbildungsverzeichnis

Abb. 2.1 QR Code für den Titel dieser Arbeit 4

Abb. 2.1 Foursquare auf Apple iOS 8.1 6

Abb. 4.1 Architektur der Applikation 9

# Tabellenverzeichnis

Tab. 1.1 Häufig genutzte physische Sensoren (übernommen aus (Baldauf et al. 2007, S. 266) ) 4

# Abkürzungsverzeichnis

aGPS assisted Global Positioning System

API Application Programming Interface

GPS Global Positioning System

GSM Global System for Mobile Communications

QR Quick Response

REST Representational State Transfer

HUD Head-up Display

# Symbolverzeichnis

# Einleitung / Motivation

Die Nutzung von Geräten der Virtual Reality (VR) und der Augmented Reality (AR) hat in den vergangenen zwei bis drei Jahren mit der Google Glass und der Oculus Rift und den beiden hinter ihnen stehenden Großunternehmen Google und Facebook erneuten Aufschwung erhalten.

Dabei hat insbesondere die Google Glass mit ihrer leichten Bauweise und Ungebundenheit zu nahestehenden Computern die Möglichkeit den Bereich des ubiquitous Computing zu verändern. Anders als Smartphones, Tablets oder konkurrierende Wearables bietet die Google Glass dabei die Chance durchgängig in das Sichtfeld des Trägers integriert zu sein. So bieten sich neue Möglichkeiten von Nutzerinteraktion die den Gedanken des anywhere und anytime auf eine neue Ebene bringen könnten, da die benötigte Handlung durch den Nutzer minimiert wird.

Anders als die Oculus Rift ist die Google Glass dabei ein Gerät welches nicht zum Anzeigen virtueller Welten bzw. der virtuellen Darstellung realer Umgebungen entwickelt wurde. Viel mehr bietet sie die Möglichkeit ähnlich einem Interface aus Computerspielen, sich über die Wahrnehmung des Nutzers zu legen und diese mit kontextsensitiver Information anzureichern.

Diese Erweiterung in der Wahrnehmung bietet vielzählige Möglichkeiten. Anders als bei bekannten mobilen Geräten bietet die AR-Brille die Chance dem Nutzer zusätzliche Informationen zu dem von ihm Betrachteten zu liefern ohne dabei seine Handlung zu unterbrechen. Es wäre also zum Beispiel möglich dringend notwendige Information für den Arbeitsfluss zu integrieren ohne dabei die Arbeit unterbrechen zu müssen.

Inwieweit eine solche kontextsensitive Erweiterung mit der heute verfügbaren Google Glass möglich ist soll Bestandteil dieser wissenschaftlichen Arbeit sein.

Diese Arbeit ist ein Teil des Glassroom Projekts welches vom Lehrstuhl für Informationsmanagement und Wirtschaftsinformatik (IMWI) der Universität Osnabrück in Kooperation mit namhaften Partnern durchgeführt wird.

# Kontextsensitivität

## Definition

Kontextsensitivität (engl. Context-Awareness) ist ein in der Wissenschaft langjährig diskutierter Begriff der erstmals von Schilit und Theimer ( 1994, S. 23) erwähnt, aber bereits in den frühen 90er Jahren von Want et al. ( 1992) beschrieben wurde. Die Autoren erdachten damals ein System mit dem, zur Koordination einer Belegschaft, der Aufenthaltsort der einzelnen Mitarbeiter ermittelt wird. Mit dieser Information sollten dann theoretisch die Abläufe verbessert werden, um zum Beispiel das Telefonsystem zu steuern.

Heute wird Kontextsensitivität als Unterfeld des Ubiquitous (bzw. pervasive) Computing gesehen (Baldauf et al. 2007, S. 263–264; Perera et al. 2014, S. 414; Bellavista et al. 2012, S. 2) und spielt in diesem eine wichtige Rolle, da die durch Sensoren gewonnene Datenmenge stetig steigt und die Kontextsensitivität als Chance gesehen wird, die relevanten Informationen hieraus zu gewinnen (Perera et al. 2014, S. 414). Indulska und Sutton gehen sogar soweit Kontextsensitivität als die Voraussetzung für ein „anywhere, anytime“ Computing zu sehen ( 2003, S. 1),

Um zu einer Definition für Kontextsensitivität zu kommen ist es notwendig zunächst dem Kontext Aufmerksamkeit zu schenken.

Die heute in der Wissenschaft weitgehenden anerkannten Definitionen für Kontext, sowie Kontextsensitivität kommen von Abowd und Dey ( 1999; Perera et al. 2014, S. 414), sie beschreiben Kontext wie folgt:

„any information that can be used to characterize the situation of entities (i.e., whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves.“ ( 1999, S. 3)

Anhand dieser Definition mit der die Autoren damals klar den Kontext als Information abgrenzten die für den Nutzer in seiner Interaktion relevant ist, gelang es ihnen dann auch eindeutig eine Definition für kontextsensitive Systeme bzw. Applikationen zu finden.

„A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task.“ (Abowd et al. 1999, S. 6)

Zwar gab es auch danach noch Versuche kontextsensitive Systeme zu definieren (Dey 2001, S. 5), die meisten wissenschaftlichen Arbeiten zu dem Thema bauen aber auf der oben genannten auf (Baldauf et al. 2007, S. 264; Perera et al. 2014, S. 414; Lee et al. 2010, S. 1; Dey et al. 1999, S. 2) und sollen hier genutzt werden.

## Möglichkeiten der Kontextsensitivität

Bei näherer Betrachtung der einzelnen Möglichkeiten fällt auf, dass es sich bei Kontextsensitivität um ein enorm vielschichtiges Thema handelt.

|  |  |
| --- | --- |
| Kontextart | Verfügbare Sensoren |
| Licht | Fotodioden, Farbsensoren, Infrarot und UV-Sensoren etc. |
| Visueller Kontext | Verschiedene Kameras |
| Audio | Mikrofone |
| Bewegung, Beschleunigung | Quecksilberschalter, Neigungssensoren, Beschleunigungssensoren, Bewegungssensoren, Magnetfelder |
| Ort | Freiluft: Global Positioning System (GPS), Global System for Mobile Communications (GSM); Indoor: Active Badge system, etc. |
| Berührung | Berührungssensoren [...] |
| Temperatur | Thermometer |
| Physische Attribute | z.B. Biosensoren zur Hautberührungsmessung oder Blutdruckmessung |

Tab. 1. Häufig genutzte physische Sensoren   
(in Anlehnung an (Baldauf et al. 2007, S. 266) )

Baldauf et al. ( 2007) und Perera et al. ( 2014) haben eine Großzahl wissenschaftlicher Arbeiten der letzten zwei Jahrzehnte (1990-2014) analysiert und dabei eine Vielzahl von Kategorisierungsmöglichkeiten für Kontextsensitive Systeme und Applikationen festgehalten. Da eine Eingrenzung und Betrachtung aller den Rahmen dieser Arbeit deutlich sprengen würde wird im folgenden die Kategorisierung nach der Stelle der Informationsverarbeitung sowie die Art des Sensors zur Informationsgewinnung beispielhaft betrachtet.

Chen ( 2004) und (Indulska und Sutton 2003) [evtl.] schlägt daher eine Einteilung der kontextsensitiven Systeme in drei Kategorien vor, die sich besonders in der Verarbeitung der gewonnen kontextsensitiven Informationen unterscheiden. Die drei von ihm vorgeschlagenen Kategorien sind: „*Direct sensor access*“, „*Middleware infrastructure“* und „*Context server*“, die Verarbeitung der Information geht dabei von direkt im Gerät verankert, über eine Verarbeitung auf implementierten Zwischenebenen, aber immer noch im Gerät hin zu einer Client-Server-Struktur, bei der jegliche Verarbeitung auf einem kontaktierten Server stattfindet. (Baldauf et al. 2007, S. 264–265; Perera et al. 2014, S. 428).

Ein weiterer Ansatz ist die Einteilung der kontextsensitiven Systeme nach Kontextart und den zugehörigen Sensoren (Tabelle 1.1). Die Idee hierhinter ist dass verschiedene Umwelteinwirkungen und Kontexte durch verschiedene Sensoren wahrgenommen werden können. So kann ein Lichtsensor genutzt werden um zu bestimmen, ob es hell oder dunkel ist, eine Uhr kann dem Gerät die Zeit mitteilen und eine GPS-Sensor den Standort auf der Welt. Weiterführen lassen sich diese Informationen dann verknüpfen um so sehr genaue Ergebnisse zu liefern. (Schmidt und van Laerhoven 2001, S. 67–68)

Im Rahmen dieser Arbeit soll eine Kombination der beiden Kategorisierungen zum Einsatz kommen. Dabei wird ein besonderer Schwerpunkt auf Middleware infrastructure und Context Server und auf die visuellen und ortsbasierten Kontextarten gelegt.

(Sensortypen Vergleich. (Baldauf et al. 2007, S. 266; Perera et al. 2014, S. 428))

### Marker / QR Codes



Abb. . QR Code für den Titel dieser Arbeit

Als spezielle Art der visuellen Kontextsensitivität sollten Identifizierung von Objekten und Orten anhand von Markern gesehen werden. Die dafür am häufigsten genutzt Möglichkeit sind die QR (Quick Response) Codes (Quelle benötigt).

QR Codes sind zweidimensionale matrixbasierte Barcodes die Information sowohl vertikaler als auch in horizontaler Richtung enthalten. Sie wurden entwickelt um schnell Informationen durch geeignete Scannersoftware auszulesen und diese dem Nutzer zur Verfügung zu stellen. Im Gegensatz zu eindimensionale Barcodes, wie man sie zum Beispiel aus dem Einzelhandel kennt, haben sie dabei ein stark gesteigertes Speichervermögen und es ist nicht möglich die Codes mit dem menschlichen Auge nachzuvollziehen (Chang et al. 2007, S. 231; Rouillard 2008, S. 52).

QR Codes wurden 1994 von dem japanischen Unternehmen Denso-Wave entwickelt. Es existieren 40 verschiedene Versionen des QR Codes die sich vor allem in ihrer Größe und dadurch in ihrem Speichervolumen, aber auch in ihrer Lesbarkeit trotz teilweiser Beschädigung unterscheiden. (QRcode.com｜DENSO WAVE)

QR Codes finden heute vielerlei Anwendung. Sie sind in der Werbung, auf Visitenkarten, an öffentlichen Orten oder einfach in ihrer ursprünglichen Bestimmung der Industrie aufzufinden. Aber immer haben sie den Nutzen Objekte oder Orte zu identifizieren und weitere kontextsensitive Informationen zum Gescannten anzuzeigen. Sei es die Internetseite eines Unternehmens in der Werbung, oder aber ein informativer Text über ein Ausstellungsstück in einem Museum. Zusätzlich ist es möglich die vom QR-Code zur Verfügung gestellte Information mit weiteren Sensoren des Geräts festgestellte Informationen zu verknüpfen, um so ein ganzheitliches kontextsensitives System zu erreichen. (Rouillard 2008, S. 52–53).

### Location-Based Services

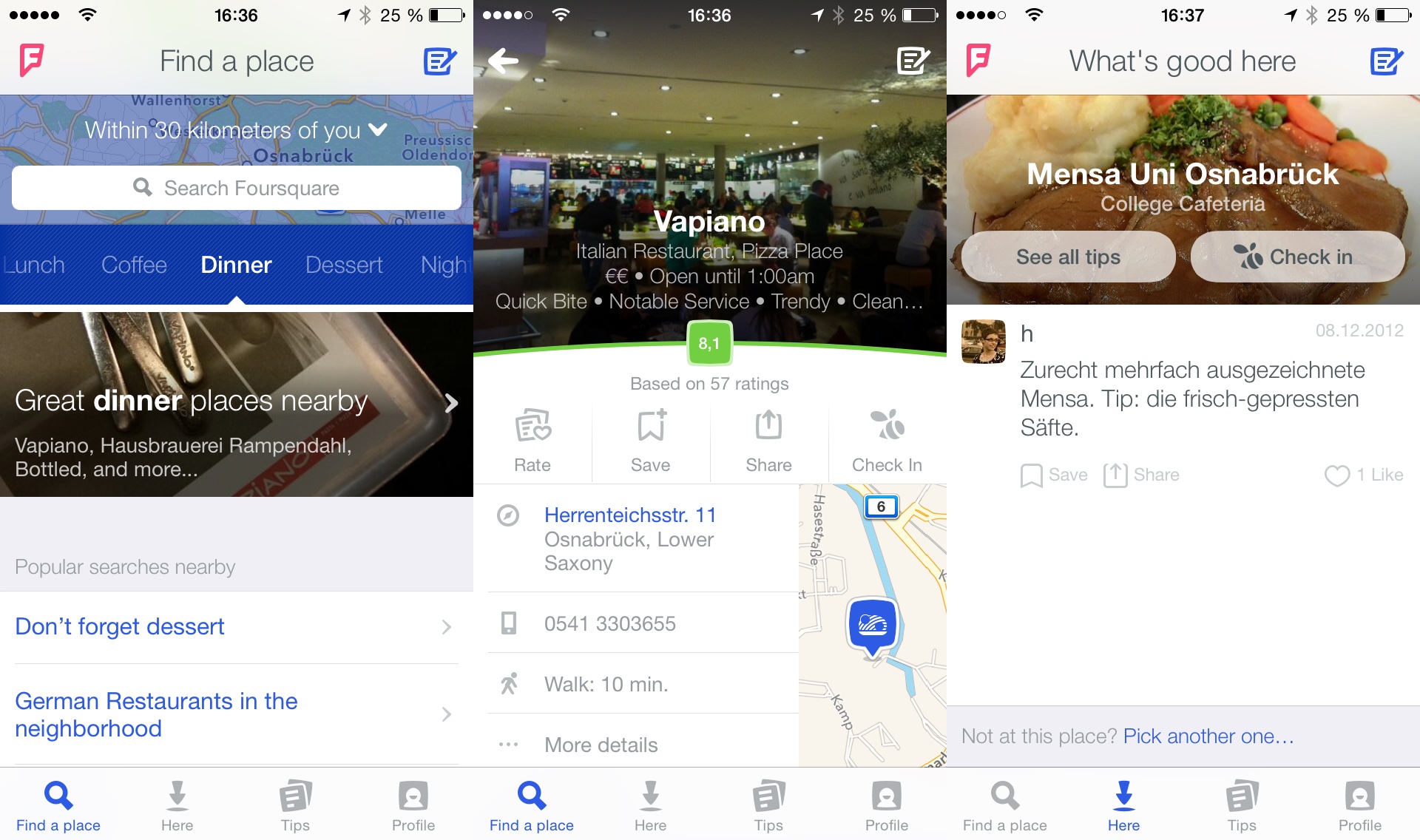


Abb. . Foursquare auf Apple iOS 8.1

Ortsbasierte Dienste (engl. Location-Based Services) sind unter Normalnutzer von Mobilen Endgeräten die heute wohl am weitesten verbreitet und bekanntesten Beispiele unter den kontextsensitiven Anwendungen dafür spricht auch, dass neben einem in den USA registrierten Patent (Portman et al. 2005) auch empirische Studien zu der Akzeptanz dieser Dienste unter Endkunden (Kölmel 2005; Junglas und Watson 2008) zum Thema erschienen sind. Zudem fällt auf, dass ein Großteil der Forschung zum Thema aus dem vergangenen Jahrzehnt stammen, was darauf schließen lässt, dass die Technik heute so etabliert ist, dass keine weitere grundlegenden Forschungsarbeiten benötigt werden. Neuere Veröffentlichungen beschäftigen sich eher mit der Verknüpfung von bestehenden Location-Based Services mit weiteren Technologien. (Gao et al. 2012; Cho et al. 2011).

Auch die immense Fülle an reinen ortsbasierten Applikationen (Foursquare, Yelp) und Applikationen die ihrem Kerndienst Extrainformationen durch ortsbasierte Dienste hinzufügen sprechen für die Annahme, dass Location-Based Services akzeptierte Anwendungen in der Welt der Mobile Devices sind.

Das UMTS-Forum beschreibt Location-Based Services bereits in seinem 13. Report, als einen Service der es Nutzern oder Geräten ermöglicht andere Personen, Fahrzeuge, Ressourcen, Dienste oder Maschinen anhand ihrer Personen zu ermitteln. Zudem ermöglicht es den Nutzer seine eigene Position zu ermitteln. (UMTS Forum 2001, S. 35).

Diese Lokalisierung geschieht dabei durch verschiedene Techniken, die sich sehr in ihrer Funktionsweise unterscheiden. Beispiele hierfür sind wie bereits in der Tabelle 1.1 genannt physische Sensoren wie GSM, GPS und aGPS, aber auch eine Ermittlung anhand von Nutzereingaben oder durch andere Applikationen ist denkbar (Indulska und Sutton 2003, S. 1). Hierbei ist allerdings zu unterscheiden ob die Lokalisierung unter freien Himmel und so über Satelliten oder Sendemasten geschehen kann, oder ob sich um eine Lokalisierung in einem Gebäude handelt, welche dann zum Beispiel über die bereits genannten QR-Codes geschehen könnte, wie sie zum Beispiel in Museen denkbar ist (Quelle benötigt).

### Objekt- und Bilderkennung

Im Bereich der Objekt- und Bilderkennung wird im Gegensatz zu den beiden vorher genannten Bereichen der kontextsensitiven Anwendungen insbesondere im Bereich der Applikationen für den Konsumentenmarkt die Auswahl sehr dünn. Zwar gab es einige Ansätze wie beispielsweise Google Goggles. Aber wirklich nutzenswerte Ansätze gab es hier in der Vergangenheit eher weniger.

Objekt- und Bilderkennung nutzt Licht- und Bildsensoren zur Erkennung und Wiedererkennung von Objekten.

# Google Glass

## Die Google Glass als Vertreter der Augmented Reality

Schon seit Mitte der 90er Jahre ist die erweiterte Realität (engl. Augmented Reality) ein intensiv beachteter Forschungsbereich. Anders als in der virtuellen Realität, die aktuell mit der Oculus Rift einen neuen starken Vertreter hat geht es bei der Augmented Realiy nicht um die künstliche Darstellung von Räumen und Objekten, sondern um die Integration von virtuellen Elementen in die Realität (Azuma 1997, S. 2). In dieser Weise wird die Augmented Reality genutzt um dem Nutzer die Möglichkeit zu bieten mit der ihm umgebenden Umwelt zu interagieren, während diese gleichzeitig um virtuelle Elemente erweitert wird.

Allerdings gibt es Uneinigkeit darüber ab welchem Zeitpunkt eine Erweiterung als Augmented Reality anerkannt werden kann. So bezeichnet Azuma ( 1997, S. 2) Augmented Reality noch als Systeme die Virtualität und Realität kombinieren, in Echtzeit interaktiv agieren und dreidimensional dargestellt werden. Er schließt zu dem 2D Einblendungen konsequent als Teil der Augmented Reality aus. Spätere Definitionen wie die von Huang et al. ( 2013, S. 1) heben diese Beschränkung auf und akzeptieren mobile Geräte als Vertreter der Augmented Reality, wenn sie mit dieser interagieren und um Elemente, gleich welcher Natur erweitern. Sie bezeichnen Vertreter dieser Geräte Kategorie als Mobile Augmented Reality (MAR) und unterteilen diese weiter in 6 Unterkategorien:

1. Notebooks
2. Personal Digital Assistants (PDAs)
3. Tablet Personal Computer (Tablets)
4. Ultra Mobile PCs (UMPCs)
5. Mobile Phones
6. AR Brillen

Die AR Brille Google, die Google Glass, wurde erstmals im Februar 2012 (Google to Sell Heads-Up Display Glasses by Year’s End - NYTimes.com) erwähnt und im April offiziell vorgestellt (Quelle benötigt). In den Verkauf ging das Gerät dann im Frühjahr 2013 für Entwickler und anerkannte Tester. Bis heute ist sie nur in sehr kleiner Auflage und vor allem für Entwickler und Forschungszwecke verfügbar.

Allerdings stellt sie ihre Inhalte nicht durch Projektion auf Brillengläsern dar, sondern integriert ein kleines Display in die obere rechte Ecke des Sichtfeldes, es handelt sich hier also um ein Head-up-Display (HUD).

## Spezifikationen und Besonderheiten der Google Glass

### Hardwarespezifikationen

(Torborg und Simpson 2012)

### Softwarespezifikationen

Die Glass benutzt Google hauseigenes Betriebssystem Android in einer speziellen Version, dem Glass Development Kit (GDK), zum Zeitpunkt dieser Arbeit lag diese in der Version XE22.0 vom 14. Oktober 2014 vor. Android basiert auf Java und bietet daher die Möglichkeit bestehende Bibliotheken anzusprechen und bei der Entwicklung zu nutzen.

Android besitzt einige spezielle Spezifikationen die bei der Entwicklung eine Applikation beachtet werden müssen, so verhindert das Betriebssystem beispielsweise automatisch rechenaufwendige Aufgaben auf dem UI Thread auszuführen, da dies zum Einfrieren der Benutzeroberfläche führen könnte.

Zur Darstellung

Android und GDK 🡪 Versionsnummern, deprecated Methoden, Live Cards, Cardbuilder, Besonderheiten.

# Einblendung von kontextsensitiven Inhalten auf der Google Glass

## Idee und Funktionsweise der kontextsensitiven Applikation

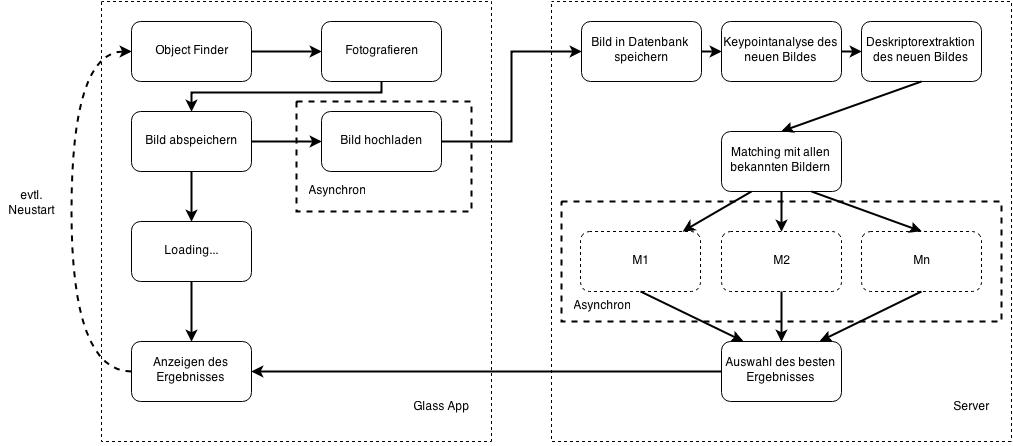


Abb. . Architektur der Applikation

Bringt man die Google Glass als Vertreter eines Geräts der Augmented Reality und Kontextsensitivität zusammen, fällt offensichtlich der hohe Überschneidungsgrad der beiden Themenbereiche auf. Selbst eine kleine, auf der Glass vorinstallierte, Anwendung wie der Kompass erfüllen streng genommen bereits die Anforderungen um als kontextsensitive Applikation akzeptiert werden zu können. Hier wird anhand der Position des Nutzers über GPS die Himmelsrichtungen ermittelt, es werden also ortsbasierte Informationen angezeigt und somit die Wahrnehmung des Nutzer um, in seinem Kontext relevante Informationen ergänzt, was sowohl der in 2.2.1 angesprochenen Definition eines Location-Based Services, als auch der in 2.1 genannten eines Kontextsensitiven Systems genügt.

Auch einige von freien Entwicklern veröffentlichte Anwendungen gehen den Weg der Kontextsensitivität und entwickelten bereits Apps die, die Möglichkeiten der Google Glass nutzen, um die über Sensoren ermittelten Außenwelteinflüsse auszuwerten und so dem Nutzer kontextsensitive Informationen anzubieten.

Die Idee der Applikation entstand direkt aus dem angeschlossenen Forschungsprojekt Glassroom. Durch Nutzung einer AR-Brille, in diesem Fall der Google Glass, soll es einem Mitarbeiter im technischen Kundendienst ermöglicht werden allein durch Interaktion mit der AR-Brille zusätzliche Information zu der vor ihm liegenden Aufgabe und den ihm eventuell bisher unbekannten Komponenten zu erhalten. Durch einfaches Fotografieren mit der Brille soll ermöglicht werden zusätzliche Informationen zu bereits bekannten Objekten anzubieten und diese dem Nutzer über Texteinblendung zur Verfügung zu stellen.

Die Brille schickt dafür das Foto an einen Server auf dem die Übereinstimmung des Bildes mit allen bereits bekannten Bildern ermittelt wird. Bei Erfolg werden die Informationen zu dem am besten übereinstimmenden Bild an die Google Glass zurückgesendet. Erreicht der Matchingprozess mit keinem der bekannten Bilder eine vorher festgelegte Akzeptanzgrenze wird der Prozess abgebrochen und dem Nutzer eine entsprechende Fehlermeldung angezeigt.

Um diese rechenintensiven und aufwendigen Prozesse von der Glass zu nehmen, wurde jegliche Rechenarbeit auf einen Server in der Cloud ausgelagert. Auf der Glass selber verbleiben nur die den User direkt betreffenden Prozesse wie UI-Darstellung (User Interface), Fotografieren, Upload des Bildes und Anzeigen der Ergebnisse. Eine weitere Einschränkung entstand durch die Auslagerung der benötigten Methoden in der angewandten Computer Vision Bibliothek in ein Nonfree-Modul.

Der Matchingprozess erfolgt dabei in 4 Schritten:

1. Keypointerkennung im hochgeladenen Bild
2. Extraktion der Deskriptoren die das Bild beschreiben
3. Matching der Deskriptoren gegen bereits ermittelte, abgespeicherte Deskriptoren
4. Auswahl des besten Ergebnisses und Rückgabe des Ergebnisses an die Glass

Diese Vorgehensweise hat besonders im Bereich der Skalierbarkeit enorme Vorteile. Es erleichtert aber auch den nachträglichen Austausch der AR-Brille oder sogar die Integration eines weiteren mobilen Gerätes, da der Server über eine einfache REST-API (Representational State Transfer - Application Programming Interface) verfügt. Zudem verschwindet durch die serverseitige Pflege der Vergleichsdaten das Problem der inkonsistenten und redundanten Datenhaltung.

## Vorstellung von OpenCV und der verwendeten Algorithmen

### OpenCV und JavaCV

Zur Implementation der Kernlogik der Applikation wurde OpenCV beziehungsweise der Wrapper JavaCV von Bytedeco genutzt. Bei OpenCV (http://opencv.org) handelt es sich um eine in C++ und C implementierte, frei erhätlliche Bibliothek die gängige Methoden der Computer Vision in sich vereint und so Forschern und interessierten Nutzern einen Eintritt bietet. OpenCV wird aktiv für C++, Java, Python, Ruby, Matlab und einige weiter Programmiersprachen entwickelt (Bradski und Kaehler 2008). Auch für Android findet man Implementationen. Die Umsetzungen für die verschiedenen Programmiersprachen sind allerdings verschieden stark umgesetzt und so kann es vorkommen, dass wichtige Grundfunktionen in der genutzten Sprache noch gar nicht zur Verfügung stehen und aus diesem Grund auf die Kernimplementierung in C++ zurück gegriffen werden müsste.

Um solche Probleme zu umgehen wurde für die Implementierung des hier vorgestellten Prototypen auf JavaCV (<https://github.com/bytedeco/javacv>) und JavaCPP (<https://github.com/bytedeco/javacpp>) zurückgegriffen. Bei den beiden Bibliotheken handelt es sich um Interfaces zu gängigen C++ Bibliotheken. Umgesetzt wurden diese von Bytedeco, einer Gemeinschaft von Entwicklern, die sich dem Ziel verschrieben haben C++ Bibliotheken in Java zugänglich zu machen. Durch Nutzung dieser konnte ein gut konfigurierbarer Zugang zu OpenCV geschaffen werden. Zur Implementierung dieser Arbeit wurde OpenCV in der Version 2.4.9, sowie JavaCPP in der Version 0.9 genutzt.

### SURF & Nearest Neighbor Matching

Wissenschaftliche Arbeit zu Surf, Sift, Brisk und Freak. Wie sie grob funktionieren, warum welches genutzt wurde etc.

# Umsetzung einer kontextsensitiven Applikation mit OpenCV

## Vorstellung der Implementation / ausgewählter Programmteile

**private** ArrayList<opencv\_features2d.DMatch> getGoodMatches

(opencv\_features2d.DMatchVectorVector matches) {

ArrayList<opencv\_features2d.DMatch> goodMatches = **new** ArrayList<opencv\_features2d.DMatch>();

**for** (int j = 0; j < matches.size(); j++) {

double mRatio = matches.get(j, 0).distance() /

matches.get(j, 1).distance();

**if** (mRatio <= RATIO)

goodMatches.add(matches.get(j, 0));

}

**return** goodMatches;

}

## Fallstudie / Auswertung der Applikation

# Fazit und Ausblick

# Literaturverzeichnis

Abowd, Gregory D.; Dey, Anind K.; Brown, Peter J.; Davies, Nigel; Smith, Mark; Steggles, Pete (1999) Towards a better understanding of context and context-awareness. In Gellersen, Hans-W. (Hrsg) Handheld and ubiquitous computing. Berlin, Heidelberg, Springer Berlin Heidelberg, 304–307.

Alahi, a.; Ortiz, R.; Vandergheynst, P. (2012) FREAK: Fast Retina Keypoint. 2012 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, 510–517.

Azuma, Ronald T (1997) A Survey of Augmented Reality. 4 (August):355–385.

Baldauf, Matthias; Dustdar, Schahram; Rosenberg, Florian (2007) A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing, 2 (4):263–277.

Bay, Herbert; Tuytelaars, Tinne; Gool, Luc Van (2006) Speeded Up Robust Features. Computer Vision–ECCV 2006, 3951 (September):346–359.

Bellavista, Paolo; Corradi, Antonio; Fanelli, Mario; Foschini, Luca (2012) A survey of context data distribution for mobile ubiquitous systems. ACM Computing Surveys, 44 (4):1–45.

Bradski, Gary; Kaehler, Adrian (2008) Learning OpenCV Computer Vision with the OpenCV Library. 1. Auflage, Sebastopol, CA, O’Reilly, Inc.

Chang, Yao-Jen; Tsai, Shih-Kai; Chang, Yao-Sheng; Wang, Tsen-Yung (2007) A novel wayfinding system based on geo-coded qr codes for individuals with cognitive impairments. Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility - Assets ’07. New York, New York, USA, ACM Press, 231.

Chen, HL (2004) An intelligent broker architecture for pervasive context-aware systems. University of Maryland, .

Cho, Eunjoon; Myers, Seth A.; Leskovec, Jure (2011) Friendship and mobility. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’11. New York, New York, USA, ACM Press, 1082.

Dey, A.K.; Salber, D.; Abowd, G.D.; Futakawa, M. (1999) The Conference Assistant: combining context-awareness with wearable computing. Digest of Papers. Third International Symposium on Wearable Computers. IEEE Comput. Soc, 21–28.

Dey, Anind K. (2001) Understanding and Using Context. Personal and Ubiquitous Computing, 5 (1):4–7.

Gao, Huiji; Tang, Jiliang; Liu, Huan (2012) Exploring Social-Historical Ties on Location-Based Social Networks. ICWSM. 114–121.

Huang, Zhanpeng; Hui, Pan; Peylo, Christoph; Chatzopoulos, Dimitris (2013) Mobile augmented reality survey: a bottom-up approach. Arxiv.Org, .

Indulska, J; Sutton, P (2003) Location management in pervasive systems. Workshop on Wearable, Invisible, Context-Aware, Ambient, Pervasive and Ubiquitous Computing. Adelaide, Australia, Australian Computer Society, 143–151.

Junglas, Iris A.; Watson, Richard T. (2008) Location-based services. Communications of the ACM, 51 (3):65–69.

Kölmel, Dr. Bernhard (2005) Location-based services.

Lee, Sangkeun; Chang, Juno; Lee, Sang-goo (2010) Survey and Trend Analysis of Context-Aware Systems. Information-An International Interdisciplinary Journal, 14 (2):527–548.

Leutenegger, Stefan; Chli, Margarita; Siegwart, Roland Y. (2011) BRISK: Binary Robust invariant scalable keypoints. 2011 International Conference on Computer Vision. Ieee, 2548–2555.

opencv dev team OpenCV API Reference — OpenCV 2.4.9.0 Documentation. http://docs.opencv.org/modules/refman.html, abgerufen am 09.10.2014.

Perera, Charith; Zaslavsky, Arkady; Christen, Peter; Georgakopoulos, Dimitrios (2014) Context Aware Computing for The Internet of Things: A Survey. IEEE Communications Surveys & Tutorials, 16 (1):414–454.

Portman, Eric A.; Gailey, Michael L.; Holmes, Chad S.; Burgiss, Michael J.; Smith, Angela King; Pitts III, Ashton F.; Dempsen, Stephen L.; Che, Vinny Wai-yan (2005) Location-based services. USA, .

Rouillard, José (2008) Contextual QR Codes. 2008 The Third International Multi-Conference on Computing in the Global Information Technology (iccgi 2008), :50–55.

Schilit, B.N.; Theimer, M.M. (1994) Disseminating active map information to mobile hosts. IEEE Network, 8 (5):22–32.

Schmidt, A; Laerhoven, K. van (2001) How to build smart appliances? IEEE Personal Communications, 8 (4):66–71.

Torborg, Scott; Simpson, Star (2012) Google Glass Teardown. http://www.catwig.com/google-glass-teardown/, abgerufen am 16.10.2014.

UMTS Forum (2001) Report No. 13. London, .

Want, Roy; Hopper, Andy; Falcão, Veronica; Gibbons, Jonathan (1992) The active badge location system. ACM Transactions on Information Systems, 10 (1):91–102.

QRcode.com｜DENSO WAVE. http://www.qrcode.com/en/index.html, abgerufen am 16.10.2014.

Google to Sell Heads-Up Display Glasses by Year’s End - NYTimes.com. http://bits.blogs.nytimes.com/2012/02/21/google-to-sell-terminator-style-glasses-by-years-end/?ref=technology, abgerufen am 20.11.2014.

Anhang

Unterkapitel des Anhangs

Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text.

Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text.

Zweites Unterkapitel des Anhangs

Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text.

Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text.

Abschließende Erklärung

Ich versichere hiermit, dass ich diese Bachelorarbeit Einblendung von kontextsensitiven Inhalten auf der Google Glass selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel und Quellen angefertigt habe, sowie den benutzten Quellen wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Osnabrück, den 21. November 2014

Jannik Hoffjann