



Konzeption und prototypische Implementierung eines Verfahrens zur POI Clusterbehandlung innerhalb einer Augmented Reality Anwendung

Bachelorarbeit an der Universität Ulm

Vorgelegt von:

Julia Müller

julia-2.mueller@uni-ulm.de

Gutachter:

Prof. Dr. Manfred Reichert

Betreuer:

Marc Schickler

2014

"Konzeption und prototypische Implementierung eines Verfahrens zur POI Clusterbehandlung
innerhalb einer Augmented Reality Anwendung"
Fassung vom 11. Mai 2014

Kurzfassung

Augmented Reality ist eine Technologie, die in vielen Anwendungsbereichen zum Einsatz kommt. Vor allem in mobilen Endgeräten wird diese Technologie zunehmend eingesetzt.

Diese Arbeit beschäftigt sich mit einer mobilen Anwendung, die Augmented Reality integriert hat. Dabei wird die Echtzeit-Aufnahme des Sichtfeldes der Geräte-Kamera des Benutzers durch virtuelle Informationen erweitert und auf dem Display angezeigt. Diese virtuellen Informationen stellen dabei Interessenspunkte aus der Umgebung des Benutzers dar.

Die Darstellung und das Interagieren mit diesen Punkten führt jedoch bei Clusterbildungen zu Problemen. Deshalb werden in dieser Arbeit zwei unterschiedliche Konzepte erstellt, die dieses Problem beheben. Daraufhin werden diese beiden Konzepte prototypisch implementiert.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation und kurze Problemstellung	1
1.2. Zielsetzung und Aufbau der Arbeit	2
2. Problemstellung	5
2.1. AREA	5
2.2. Detaillierte Problemstellung	8
3. Problemlösung und Konzepte	9
3.1. Untersuchung bereits existierender Applikationen	10
3.1.1. Vorstellung ausgewählter Applikationen	10
Wikitide	10
yelp	14
junaio	16
mixare	18
3.1.2. Bewertung	21
3.2. Eigene Konzepte	24
3.2.1. Erstes Konzept	24
3.2.2. Zweites Konzept	26
4. Implementierung	31
4.1. Implementierung des ersten Konzepts	32
4.2. Implementierung des zweiten Konzepts	35

5. Vorstellung der POI Clusterbehandlung nach Umsetzung der beiden Konzepte	41
5.1. Vorstellung des ersten Konzepts	41
5.2. Vorstellung des zweiten Konzepts	44
5.3. Bewertung der eigenen Konzepte	47
6. Fazit und Ausblick	49
6.1. Fazit	49
6.2. Ausblick	50
A. Verzeichnisse	53

1

Einleitung

1.1. Motivation und kurze Problemstellung

Die Nachfrage der Technologie *Augmented Reality*¹ stieg in den letzten Jahren stetig. Dies liegt zum Teil daran, dass diese Technologie in den zunehmend gefragten Smartphones eingesetzt wird. Ob Zuhause beim Einrichten der Wohnung (vergleiche [2]), im Auto als Ersatz von Bedienungsanleitungen (vergleiche [3]) oder auch als Hilfestellung in der Umgebung, um beispielsweise Sehenswürdigkeiten oder Restaurants in dieser ausfindig zu machen. Durch die immer leistungsfähigeren Smartphones werden der Augmented Reality Technologie in Bezug auf diese kaum Grenzen gesetzt [4].

Die gerade genannte Hilfestellung in der Umgebung ist bei Nutzern sehr beliebt, um sich zum Beispiel einen Überblick in Städten zu verschaffen, an welcher Stelle sich was

¹Augmented Reality („erweiterte Realität“, kurz: AR) erlaubt dem Nutzer in Echtzeit, die Realität kombiniert mit virtuellen Objekten wahrzunehmen [1].

1. Einleitung

befindet. Auf dem Markt gibt es eine Vielzahl an Applikationen, die diese Art von AR für die Betriebssysteme *iOS* [5] und *Android* [6] anbieten.

Eine dieser Applikationen ist *AREA* [7], die in Abschnitt 2.1 näher beschrieben wird. In dieser wird dem Benutzer unter anderem eine Aufnahme des Sichtfeldes auf dem Display seines mobilen Endgeräts, auf dem die Anwendung installiert ist, angezeigt. Ferner werden auf dem Display *Point Of Interests* („Interessenspunkte“, kurz: POIs) angezeigt.

Bei mehreren POIs im Sichtfeld des Nutzers entsteht das Problem, dass sich Cluster bilden und sich die Interessenspunkte dadurch überschneiden oder vollständig durch andere verdeckt werden. Dies hat zur Folge, dass viele dieser davon betroffenen POIs vom Benutzer nicht mehr angewählt und somit keine weitere Informationen zu diesen aufgerufen werden können. Eine detailliertere, mit einer Abbildung veranschaulichte Problemstellung, wird in Abschnitt 2.2 aufgeführt.

1.2. Zielsetzung und Aufbau der Arbeit

Es ist Ziel dieser Arbeit, zwei Konzepte zu entwerfen und daraufhin prototypisch zu implementieren, die das Problem der durch andere teilweise oder vollständig verdeckten POIs in Clustern lösen. Dies soll die Möglichkeit bieten, dass auch diese anwählbar und für den Benutzer gut sichtbar sind.

Um dieses Ziel zu erreichen, wird zunächst die Applikation *AREA*, in der dieses Problem zu lösen gilt, vorgestellt und anhand dessen das Problem genauer erläutert.

In Kapitel 3 werden daraufhin anfangs mehrere, mit *AREA* vergleichbare, auf dem Markt erhältliche Applikationen detailliert vorgestellt und danach anhand von Kriterien bewertet und miteinander verglichen. Diese Analyse führt die Stärken und Schwächen dieser Applikationen auf. Auf Basis dessen werden zwei eigens erstellte Konzepte beschrieben, die jene Schwächen berücksichtigen und optimieren. Zudem werden die eigenen Konzepte noch um weitere Funktionen ergänzt.

Im nächsten Schritt wird in Kapitel 4 auf das Vorgehen der prototypischen Implementierung der beiden Konzepte eingegangen.

1.2. Zielsetzung und Aufbau der Arbeit

Danach werden im fünften Kapitel die Konzepte mit Hilfe von Screenshots nach der Implementierung vorgestellt und veranschaulicht, wie nun mit der POI Clusterbildung umgegangen wird. Dieses Kapitel endet mit einer Bewertung der eigenen Konzepte.

Das letzte Kapitel beinhaltet das Fazit, das die wesentlichen Ergebnisse dieser Arbeit umfasst. Ferner schließt die Arbeit mit einem Ausblick auf AREA ab.

2

Problemstellung

In diesem Kapitel wird das Problem geschildert, das es in dieser Arbeit zu lösen gilt. Bevor jedoch darauf eingegangen werden kann, muss die dieser zugrunde liegende Anwendung AREA vorgestellt werden.

2.1. AREA

Augmented Reality Engine Application (kurz: *AREA*) ist eine AR Anwendung, die im Zuge einer Bachelorarbeit für das Betriebssystem *iOS* von *Apple* [8] für ein *iPhone* [9] entwickelt wurde [7] [10]. Auf diese Quelle wird im Laufe der Arbeit immer wieder Bezug genommen.

Inzwischen ist diese Anwendung jedoch auch für *Android*-Geräte verfügbar. In dieser Arbeit wird ausschließlich mit der *Android*-Version gearbeitet.

2. Problemstellung

Nach dem Starten von AREA erscheint eine Kartenansicht, in der sowohl der Standort des Benutzers als auch POIs gekennzeichnet sind. Um zur AR Ansicht zu gelangen, kann der Button, der durch ein Kamera-Icon visualisiert ist und sich oberhalb der Kartenansicht befindet, angewählt werden. Dadurch wird eine Aufnahme des Sichtfeldes auf dem Display des Benutzers angezeigt. Das Sichtfeld wird durch virtuelle Informationen erweitert. Diese virtuelle Informationen stellen Point Of Interests aus der Umgebung des Nutzers dar. Zur Berechnung der anzuzeigenden Interessenspunkte wird die genaue Lage derer sowie die Position des Geräts beziehungsweise des Nutzers benötigt.

In Abbildung 2.1 ist ein Screenshot abgebildet, auf dem die Aufnahme des Sichtfeldes, POIs sowie ein Radar zu erkennen sind.

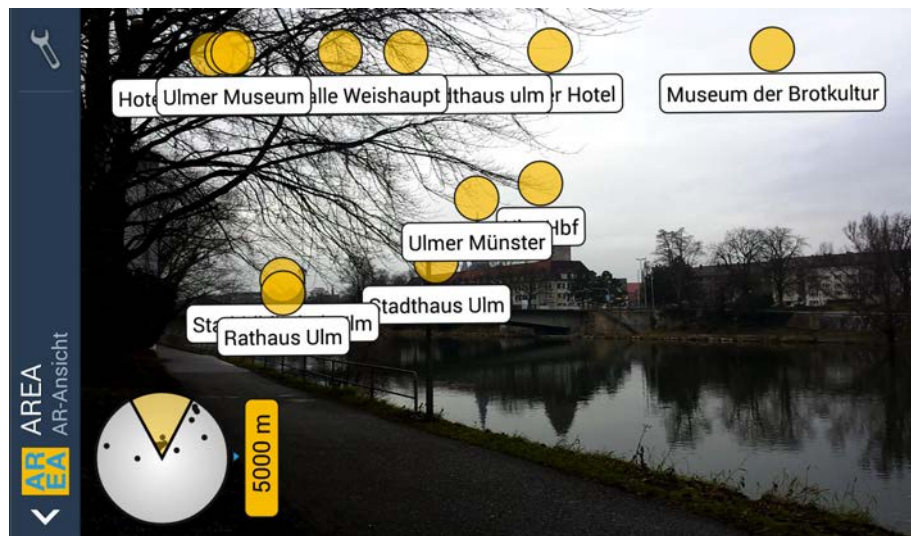


Abbildung 2.1.: Screenshot AREA

Interessenspunkte werden durch gelbe, leicht transparente Kreise dargestellt. Unter jedem dieser virtuellen Objekte ist der dazugehörige Name ablesbar. Jeder einzelne dieser POIs ist anwählbar. Berührt der Benutzer einen von diesen, werden dem Benutzer detaillierte Informationen zu diesem Punkt angezeigt (Abbildung 2.2 (links)).

Befinden sich mehrere POIs neben- oder hintereinander, werden die POIs, die sich näher am Standort des Benutzers befinden, im Vordergrund angezeigt, während weiter entfernte von diesen teilweise oder fast komplett überdeckt werden.

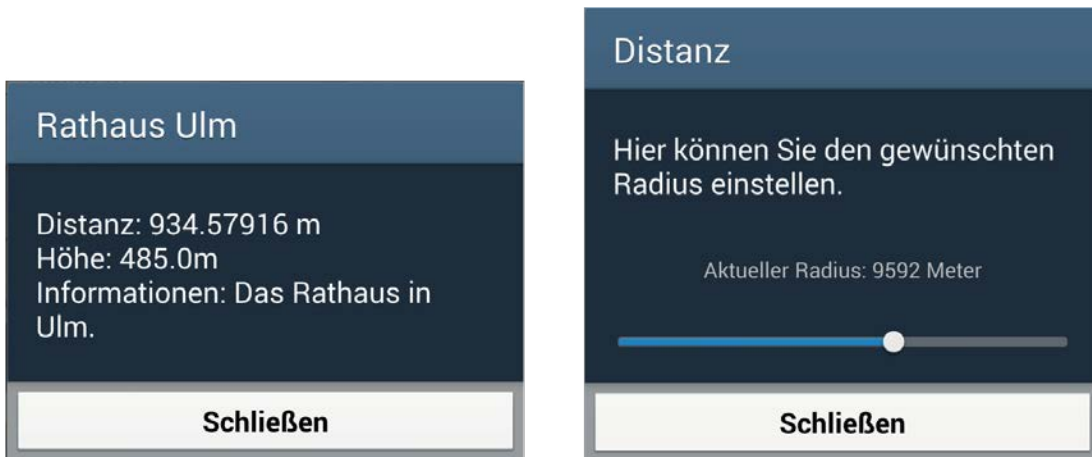


Abbildung 2.2.: Informationen zu einem angewählten POI (links) - Distanzeinstellung (rechts) (Screenshots AREA)

Links unten im Display befindet sich ein Radar (siehe Abbildung 2.1), der kleine schwarze Punkte enthält. Diese Punkte stellen ebenfalls POIs in der Umgebung des Benutzers dar. Im Radar ist das aktuelle Sichtfeld mittels eines gelben Dreiecks veranschaulicht. Das kleine blaue Dreieck, das sich außerhalb des Radars befindet, kennzeichnet den Norden. Sobald sich der Benutzer dreht, dreht sich auch der Radar und die POIs im Sichtfeld werden dementsprechend aktualisiert. Der Benutzer hat die Möglichkeit, die Distanz des Radius der angezeigten POIs einzustellen, in dem dieser entweder das Werkzeug-Symbol oder den Radar anwählt. Dadurch öffnet sich eine Ansicht, in der der Radius zwischen 1000 Meter und 15 Kilometer eingestellt werden kann (Abbildung 2.2 (rechts)). Ändert der Benutzer diese Einstellungen mit Hilfe des Schiebereglers, werden die POIs in der Umgebung neu berechnet. Die aktuell eingestellte Distanz ist unterhalb des Radars ablesbar.

2. Problemstellung

2.2. Detaillierte Problemstellung

Die Anwendung AREA wurde nun detailliert vorgestellt. Bei der Problemstellung werden nun die Interessenspunkte betrachtet, die sich direkt neben- oder hintereinander befinden.

POIs liegen oft in unmittelbarer Nähe oder in gleicher Richtung. Dies hat zur Folge, dass dadurch die virtuellen Objekte auf dem Display des Nutzers sehr nah beieinander liegen, diese sich dadurch überschneiden oder sogar komplett überdecken. Dadurch werden Cluster gebildet.

In Abbildung 2.3 ist die gerade erläuterte Situation veranschaulicht. Viele Interessenspunkte werden durch andere, die eine geringere Distanz zum Nutzer aufweisen, teilweise oder nahezu vollständig überdeckt. Der Benutzer kann in einem solchen Zustand zumeist nur mit den POIs interagieren, die im Vordergrund liegen. Je mehr die Punkte durch andere verdeckt werden, desto schwieriger wird das interagieren mit diesen. Komplett überdeckte POIs können überhaupt nicht angewählt werden.

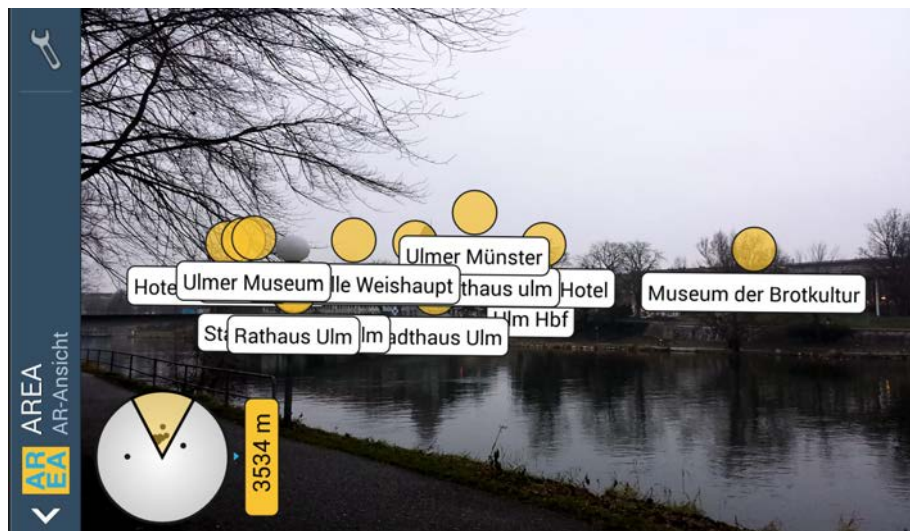


Abbildung 2.3.: Cluster (Screenshot AREA)

In dieser Bachelorarbeit ist es nun Aufgabe, die eben aufgeführte Problematik zu lösen, sodass auch POIs, die durch andere verdeckt werden, problemlos anwählbar sind.

3

Problemlösung und Konzepte

Nachdem im vorherigen Kapitel AREA vorgestellt und die Problemstellung von dieser Anwendung erläutert wurde, ist es Ziel in diesem Kapitel, Konzepte für AREA zu entwerfen, die die POI Clusterbildung verbessern.

Zunächst werden bereits vorhandene Augmented Reality Applikationen detailliert betrachtet, danach anhand einiger Kriterien bewertet und miteinander verglichen. Zudem werden diese Applikationen auf ihren Umgang mit Clustern untersucht. Im Anschluss werden eigens entwickelte Konzepte vorgestellt.

3.1. Untersuchung bereits existierender Applikationen

Im Folgenden werden die Applikationen *Wikitude* [11], *yelp* [12], *junaio* [13] und *mixare* [14] genauer untersucht. Nachdem auf diese eingegangen wurde, werden diese anhand einiger Kriterien bewertet und gegenübergestellt.

3.1.1. Vorstellung ausgewählter Applikationen

Wikitude

Wikitude ist eine beliebte, frei verfügbare AR Applikation. Diese „ist die erste Mobile-Applikation, die auf einem standortbezogenen Ansatz der Augmented Reality (AR) basiert.“ [15]

In dieser Anwendung können Interessenspunkte in einer Liste, einer Karte oder auch in einer Augmented Reality Ansicht angezeigt werden. Wikitude verfügt über viele Bezugsquellen, von denen Daten eingeholt werden. Unter anderem sind hier *Wikipedia* [16] und *flickr* [17], ein Portal in dem zum Beispiel Bilder hochgeladen, bewertet, geteilt oder kommentiert werden können, zu nennen. Dementsprechend existiert in dieser Applikation auch ein breites Spektrum an Interessenspunkten.

In der AR Ansicht werden die virtuellen POIs als graue Rechtecke dargestellt (Abbildung 3.1).

Diese beinhalten Informationen über den Namen sowie die Distanz vom Standort des Benutzers bis zum Interessenspunkt. Zusätzlich zu den genannten Informationen wird auch jeweils ein passendes Icon oder ein Foto des POI angezeigt. Dies ist abhängig davon, von welcher Quelle die Daten zu einem Interessenspunkt stammen. Stammen die Daten beispielsweise von Wikipedia, so erscheint an dieser Stelle das Wikipedia Logo. Werden die Informationen von flickr bezogen, werden entsprechende Bilder angezeigt. Befinden sich mehrere POIs in einem Cluster, sodass diese sich in der Ansicht überschneiden oder komplett durch davor liegende POIs überdeckt werden, werden diese zusammengefasst. Wie dies in Wikitude visualisiert ist, ist in Abbildung 3.2 zu sehen.

3.1. Untersuchung bereits existierender Applikationen



Abbildung 3.1.: Einzelne POIs (Screenshot Wikitude)

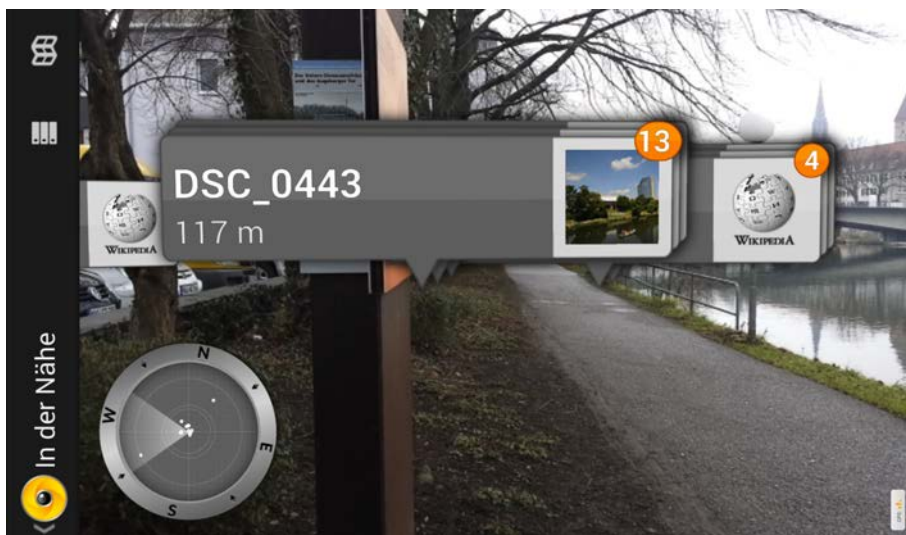


Abbildung 3.2.: Ansicht eines Clusters (Screenshot Wikitude)

Dem Benutzer wird nur der am nächsten liegende Interessenspunkt der zusammengefassten, virtuellen Objekte angezeigt. Wie viele POIs sich in diesem Cluster jedoch tatsächlich befinden, zeigt der kleine gelbe Kreis in der oberen rechten Ecke des Recht-

3. Problemlösung und Konzepte

ecks. Um nicht nur durch die Anzahl einen Cluster zu visualisieren, werden hinter dem angezeigten Interessenspunkt zusätzlich zwei weitere POIs gestapelt angeordnet.

Möchte der Benutzer nun die POIs betrachten, die gebündelt wurden, so werden diese durch Berühren eines virtuellen clusterbildenden Rechtecks angezeigt (Abbildung 3.3).



Abbildung 3.3.: Darstellung der Ansicht der POIs innerhalb eines Clusters (Screenshot Wikitude)

Diese werden in Form eines großzügigen Kreises visualisiert, wobei in diesem nur maximal sechs Interessenspunkte darstellbar sind. Durch leichtes Drehen oder Kippen des Geräts, auf dem sich Wikitude in Ausführung befindet, ist eine Betrachtung aller, der im Kreis angeordneten POIs, möglich.

Sofern sich mehr als sechs POIs in einem Cluster befinden, erscheint zusätzlich ein Button "Weitere", mit welchem die übrigen POIs betrachtet werden können. Detaillierte Informationen über einzelne Interessenspunkte erhält der Benutzer, indem er diese auswählt. Mit Hilfe des „Abbrechen“-Buttons, der sich wie der Button „Weitere“ ebenfalls in der Mitte des Kreises befindet, kann die Ansicht geschlossen werden.

Wikitude geht mit Clusterbildungen gut um. Man kann jeden einzelnen Interessenspunkt auswählen, um mehr Informationen zu erhalten. Durch das Zusammenfassen der POIs besteht auch nicht die Gefahr, dass zu viele von diesen im Display angezeigt werden und

3.1. Untersuchung bereits existierender Applikationen

es dadurch unübersichtlich werden kann. Ebenfalls vorteilhaft ist, dass die Informationen in den virtuellen Rechtecken ausreichen, um sich einen groben Überblick zu verschaffen. Der Benutzer wird nicht durch überflüssige Informationen abgelenkt.

Verbesserungswürdig in dieser Applikation wäre unter anderem das Einführen eines „Zurück“- Buttons in der Cluster Kreisansicht. Momentan kann der Benutzer nur weitere POIs innerhalb der nächsten Ansicht betrachten. Problematisch wird dies, wenn sich sehr viele POIs in einem Zusammenschluss befinden. Möchte der Benutzer zum Beispiel die POIs ansehen, die sich in der Ansicht zuvor befinden, muss dieser so lange den „Weiter“ Button betätigen, bis er schließlich die gewünschte Ansicht angezeigt bekommt. Ein Button, der die Funktion des Zurückschaltens erfüllt, wäre hier für den Benutzer sehr hilfreich.

Zudem hat der Benutzer bei mehrfachem Betätigen des „Weiter“ Buttons keinen Überblick mehr darüber, wie oft dieser die Ansicht bereits gewechselt hat und ob schon alle POIs im Cluster betrachtet wurden. Eine Anzeige der aktuellen Seite sowie die Gesamtzahl an Seiten wäre hier daher denkbar.

Ein weiterer Kritikpunkt ist, dass die Anordnung der POIs innerhalb der Cluster Kreisansicht sehr großflächig ist. Daher ist es dem Benutzer nicht möglich, alle sechs POIs einer Ansicht gleichzeitig zu betrachten. Dies ist nur möglich, wenn das mobile Endgerät, auf dem die Applikation installiert ist, leicht gekippt oder gedreht wird. Des Weiteren ist kein eindeutiges Prinzip der Reihenfolge der in einer Kreisansicht hinzugefügten POIs ersichtlich.

Ein störender Faktor in Wikitude ist, dass in der Augmented Reality Ansicht häufig Werbung in Form eines vorbeifliegenden Zeppelins angezeigt wird und dadurch kurzzeitig Interessenspunkte verdeckt werden.

Im Vergleich zu anderen Engines fällt der maximal einzustellende Radius in der Kameraansicht mit 1,4 Kilometern klein aus. Jedoch ist bei Wikitude zu beachten, dass viele Daten von vielen unterschiedlichen Quellen bezogen werden, weshalb es eine Vielzahl an POIs gibt. Ein größerer Radius würde also bedeuten, dass es noch mehr POIs geben würde, es dadurch zu sehr vielen Clusterbildungen käme und dadurch letztlich alles unübersichtlicher wäre.

3. Problemlösung und Konzepte

yelp

yelp „ist eine Onlineplattform, auf der Menschen sich über Geschäfte, Restaurants, Clubs, Ärzte, Handwerker, Kultureinrichtungen, religiöse Gemeinden, Immobilienhändler, Wellnessrichtungen und Reiseanbieter in ihrer Stadt informieren und austauschen können“ [18]. Die Applikation bietet aber auch die Möglichkeit der erweiterten Realität mittels der sogenannten „Monocle“ Ansicht ¹: Interessenspunkte werden durch virtuelle Objekte - schwarze transparente Rechtecke - in der Kamera-Ansicht angezeigt (Abbildung 3.4).



Abbildung 3.4.: „Monocle“ Ansicht (Screenshot yelp)

Die Einstellung eines Radius, in welchem Interessenspunkte angezeigt werden, ist in dieser Anwendung nicht möglich.

Dafür besteht aber die Möglichkeit, Kategorien in der Monocle Ansicht auszuwählen, wenn das mobile Gerät des Benutzers senkrecht gehalten wird. Zu diesen Kategorien zählen „Restaurants“ und „Bars“. Der Benutzer kann sich aber auch „Alles“ anzeigen lassen. Insgesamt werden bis zu sechs Informationen zu den einzelnen Interessenspunkten in den Rechtecken angezeigt: Immer zu sehen sind der Name des Punktes,

¹In der yelp Applikation bezeichnet "Monocle" die Funktion des Betrachtens der POIs in der virtuellen Sicht.

3.1. Untersuchung bereits existierender Applikationen

eine Bewertungsskala anhand von Sternen, wie viele Beiträge zum jeweiligen Punkt vorhanden sind, die Entfernung vom Standort des Benutzers zum Interessenspunkt in Kilometer und eine einzuordnende Kategorie. Abhängig davon, ob zu einem Interessenspunkt eine Öffnungszeit hinterlegt ist, wird ein Status darüber angezeigt, ob dieser geöffnet oder geschlossen hat. Die Darstellung der Rechtecke hängt von der Position der POIs zum Benutzer ab. Die POIs drehen sich leicht waagrecht in die Richtung, in der sie liegen und zeigen somit die ungefähre Richtung an. Je nachdem, wie die einzelnen virtuellen Rechtecke zueinander stehen, entsteht eine Art Fluchtpunktperspektive.

Befinden sich mehrere POIs in unterschiedlicher Richtung, überschneiden diese sich (Abbildung 3.5).



Abbildung 3.5.: Deutliche Überschneidung bei mehreren POIs (Screenshot yelp)

Bei Berühren der einzelnen Interessenspunkte öffnet sich eine Ansicht, in der sich eine detaillierte Übersicht befindet. Zum Beispiel können Beiträge, Bewertungen oder Bilder betrachtet werden. In yelp besteht auch die Möglichkeit, eine Liste der POIs in der Umgebung einzusehen. Die Ergebnisse in dieser Liste können mittels ausgewählter Kategorien eingegrenzt werden.

yelp basiert auf einem komplett anderen Ansatz als die zuvor vorgestellte Applikation Wikitude. Es wurde sehr viel Wert auf das Design der Interessenspunkte gelegt.

3. Problemlösung und Konzepte

Der Betrachter erhält auch sofort mehrere detailliertere Informationen in den dargestellten Rechtecken. Dies hat zur Folge, dass die virtuellen Objekte sehr groß erscheinen. Dadurch ist die Ansicht auf dem Display des genutztes Gerätes mit bereits wenigen angezeigten POIs sehr schnell ausgefüllt. Die Idee mit der waagerechten Drehung in Richtung der Position der POIs ist suboptimal umgesetzt. Durch die Überschneidungen, die durch die unterschiedlichen Richtungen der POIs entstehen, ist oftmals nur noch ein kleiner Teil eines virtuellen Rechtecks sichtbar. Das hat zur Folge, dass in vielen Fällen der Name des Interessenspunktes nicht mehr abgelesen werden kann.

Die Applikation yelp ist ein gutes Beispiel dafür, dass ein optisch schönes Design in einer Augmented Reality Applikation nicht nur zu Vorteilen führt.

junaio

junaio ist laut Entwickler der am meist fortgeschrittenste „mobile augmented reality browser“ [19]. Es besteht in dieser Applikationen die Möglichkeit zwischen drei Ansichten zu wählen. POIs können in einer Karte, einer Liste oder in Form einer „Live“ Ansicht betrachtet werden. Interessenspunkte in der Kameraansicht werden bei junaio als hellgraue Rechtecke dargestellt (Abbildung 3.6). In diesen virtuellen Objekten erscheinen dem Benutzer der Name des Interessenspunktes, die Entfernung in Meter oder Kilometer zum Punkt, wo vorhanden eine Bewertungsskala sowie ein Icon für die jeweilige Kategorie des Interessenspunktes - ein Croissant symbolisiert beispielsweise eine Bäckerei. POIs, die sich geografisch weiter weg befinden, werden kleiner angezeigt. Befinden sich mehrere POIs in gleicher Richtung, werden diese bei junaio nach oben verlegt, um mehr Platz für andere POIs auf dem Display zu schaffen [20]. Dadurch entsteht die Möglichkeit, jeden Interessenspunkt zu berühren, um mehr Informationen zu diesem zu erhalten.

Bei geringerer Anzahl von POIs ist dies eine sehr gute Umsetzung die POI Clusterbildung zu behandeln. Befinden sich jedoch sehr viele POIs in der Umgebung, so wird die Ansicht unübersichtlich, wie ansatzweise in Abbildung 3.6 zu erkennen.

3.1. Untersuchung bereits existierender Applikationen

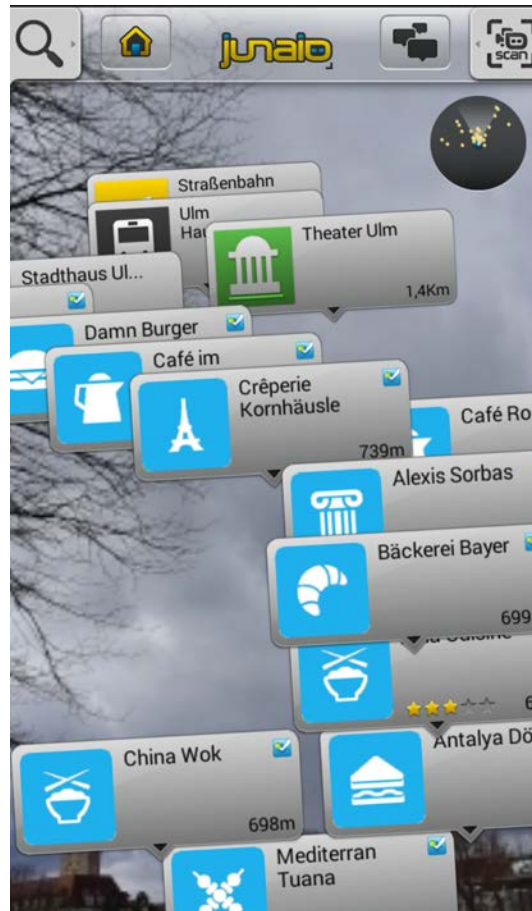


Abbildung 3.6.: Darstellung der POIs (Screenshot junaio)

Näher liegende POIs werden komplett angezeigt, entferntere Punkte werden dahinter und leicht nach oben versetzt angeordnet. Man kann immer noch alle POIs für mehr Informationen anwählen, jedoch kann der Nutzer die Informationen in den virtuellen Objekten nur noch teilweise oder gar nicht mehr erkennen. Die Namen einzelner Interessenspunkte können so nur noch erahnt werden. In junaio kann der Radius nur bis 4,6 Kilometer eingestellt werden. Dies hat allerdings den Vorteil, dass die Ansicht nicht durch zusätzliche POIs mit größeren Distanzen unübersichtlich wird.

junaio steht anderen Augmented Reality Applikationen in nichts nach, auch wenn in dieser Applikation mit weiter entfernten und hintereinander liegenden Interessenspunkten völlig anders umgegangen wird als in Wikitude.

3. Problemlösung und Konzepte

mixare

mixare ist eine kostenlose Open Source AR Engine, die ihre POI Daten von Wikipedia bezieht [14].

Es besteht die Möglichkeit sich POIs in einer Kartenansicht, als Liste oder anhand der Kamera anzeigen zu lassen. Ist die Kameraansicht ausgewählt, so werden die POIs innerhalb des eingestellten Radius als transparente Kreise mit roter Kontur verkörpert (Abbildung 3.7). Unterhalb der angezeigten Kreise befindet sich jeweils der Name und die Entfernung vom Benutzer zum jeweiligen Interessenspunkt. Diese Informationen werden in einem Rechteck dargestellt. Auffällig ist, dass sich die Rechtecke nicht immer in gleichem Abstand zu den zugehörigen POIs befinden - diese sich sogar oftmals überschneiden. Bei den POIs, die weiter entfernt sind, ist zu beobachten, dass das Informationsrechteck unterhalb der dazugehörigen Kreise liegt. Der Abstand wird sogar immer größer je weiter die POIs entfernt sind. Genau das Gegenteil lässt sich beobachten, wenn die POIs umso näher liegen. Zusammenhängende Informationen und Kreise überschneiden sich je geringer die Entfernung vom POI zum Betrachter ist. Ähnlich wie bei junoai werden die POIs, die weiter entfernt liegen, immer kleiner. Dadurch ist sofort erkennbar, welche POIs sich weiter weg befinden. Befinden sich viele virtuelle Objekte direkt neben- oder hintereinander, überschneiden sich diese. Der am nächsten liegende Interessenspunkt wird dementsprechend auf dem Display ganz vorne angezeigt und überdeckt die anderen. Da jedoch das Rechteck mit den Informationen leicht transparent ist, sind die überdeckten POIs für den Betrachter trotzdem noch sichtbar - zumindest die durch die rote Kontur auffälligen Kreise. Die Namen der POIs sowie die Entfernung sind in diesem Falle nur noch schwer lesbar (Abbildung 3.7). Problematisch durch die Überschneidung der POIs wird hierdurch auch das Interagieren mit überdeckten POIs. Liegen mehrere virtuelle Objekte direkt hintereinander, so lässt sich nur mit dem vordersten interagieren. Somit kann der Benutzer keine weiteren detaillierten Informationen zu den dahinter liegenden POIs einholen. Für gewöhnlich öffnet sich nach Berühren eines Kreises die zu dem Interessenspunkt gehörige Wikipedia Seite im Browser.

3.1. Untersuchung bereits existierender Applikationen



Abbildung 3.7.: Darstellung der POIs (Screenshot mixare)

Der Radius hat eine Reichweite von bis zu 80 Kilometern. Dies ist aber nicht auf die POIs selbst bezogen, denn diese haben eine maximale Distanz von einem Kilometer. Worauf sich der hohe Radius bezieht, ist erkennbar, wenn die Kamera auf den Boden gerichtet wird. Es erscheinen nun Pfeile, die in die Richtung der umliegenden Ortschaften zeigen. Zu jedem Pfeil existieren Informationen über den Namen und die Entfernung zum jeweiligen Ort (Abbildung 3.8).

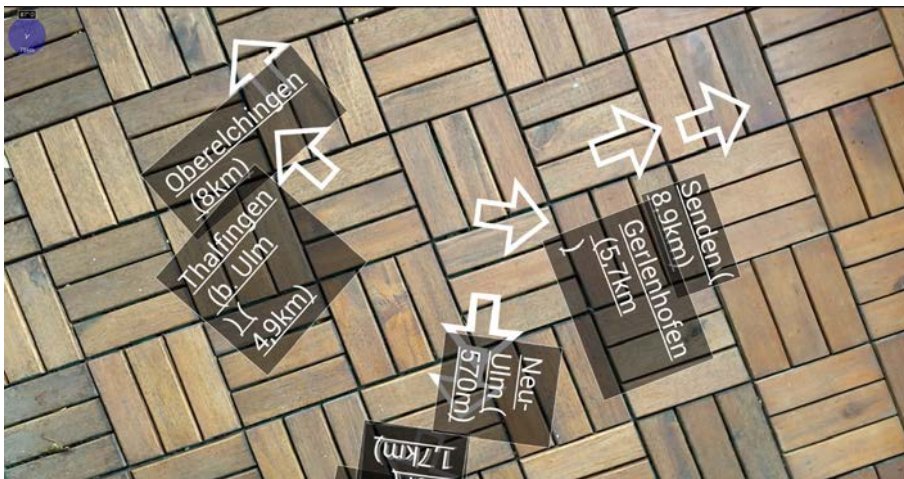


Abbildung 3.8.: Bodenansicht (Screenshot mixare)

3. Problemlösung und Konzepte

Zusammenfassend ist zu mixare zu sagen, dass die Entwickler gute Ideen hatten, diese jedoch nicht befriedigend umgesetzt wurden. Die von der Entfernung abhängige Größendarstellung der POIs ist ein guter Ansatz. Allerdings sind die Größenverhältnisse schlecht gewählt. Ist ein Interessenspunkt nur 150 Meter vom Standort des Nutzers der Applikation entfernt, so bedeckt dieser schon einen großen Teil des Displays (Abbildung 3.9).



Abbildung 3.9.: Interessenspunkt - Entfernung: 150 Meter (Screenshot mixare)

Die Entfernung der POIs ist nicht nur durch die Größe der Kreise definiert. Zusätzlich zum Namen kann der Benutzer die Entfernung zum Interessenspunkt ablesen.

Die Pfeile, die die Richtungen und Entfernungen der umliegenden Ortschaften angeben, sind eine individuelle und gute Idee.

Problematisch wird es bei mixare, wenn sich Cluster bilden. Mit geografisch weiter entfernten Punkten lässt sich hier nicht interagieren und der Benutzer kann somit keine weiteren Informationen über diese POIs erhalten.

Das Design der virtuellen Objekte ist in dieser Applikation suboptimal umgesetzt: Sie sind zweckmäßig, jedoch im Vergleich zu anderen, in dieser Arbeit behandelten Applikationen optisch weniger ansprechend. Zusätzlich lässt sich keine Erklärung dafür finden, weshalb sich bei unterschiedlichen Entfernungen die Abstände der Kreise zu den dazugehörigen Rechtecken ändern. Einheitliche Abstände wären hier von mehr Vorteil.

3.1.2. Bewertung

Beispiele für bereits existierende Engines wurden nun anhand von Wikitude, yelp, junaio und mixare gezeigt und ausführlich diskutiert. Auch auf die Vor- und Nachteile dieser Applikationen wurde eingegangen. Die aufgeführten Engines werden nun unter Zuhilfenahme von Noten in unterschiedlichen Kriterien bewertet und miteinander verglichen. Es stehen die Noten 1, 2, 3, 4 und 5 zur Verfügung, wobei 1 sehr gut und 5 sehr schlecht bedeutet. 3 entspricht in diesem Falle dem Mittelmaß. Ein „-“ verdeutlicht, dass keine Bewertung möglich ist.

Insgesamt gibt es fünf Kategorien, in denen die Applikationen zu bewerten sind. Als erstes wird das allgemeine Design der virtuellen Objekte (**Design**) der jeweiligen Applikation bewertet. Danach die Größe der POIs (**Größe**) und ob Größenunterschiede bei unterschiedlichen Entfernungen existieren (**Größenunterschiede**). Daraufhin folgt die Benotung der Informationen, die in den virtuellen Objekten abzulesen sind (**Informationen**). In diesem Falle wird danach bewertet, ob es ausreichend Informationen sind oder ob zu viele für eine kleine Übersicht angezeigt werden. Als nächstes folgt das Bewerten des Umgangs mit Clustern (**Cluster**): Wurde hier eine Lösung zur POI Clusterbehandlung gefunden und wie gut wurde diese umgesetzt.

In nachfolgender Tabelle 3.1 befindet sich eine Übersicht der einzelnen Benotungen.

	Design	Größe	Größenunterschiede	Informationen	Cluster
Wikitude	2	3	-	2	3
yelp	2	4	-	3	-
junaio	1	1	3	2	2
mixare	5	4	3	2	-

Tabelle 3.1.: Bewertungen ausgewählter, bereits existierender Engines

In der Kategorie Design schneidet junaio am besten und mixare am schlechtesten ab. Wikitude und yelp erhalten jeweils eine gute Benotung.

junaio erhält hier aufgrund des relativ einfachen Designs die Bestnote, da auf die Einzelheiten im virtuellen Objekt und das Zusammenspiel derer viel Wert gelegt wurde. Zudem ist dies ein sehr übersichtliches Design. Jede Kategorie in junaio besitzt eine eigene Farbe. Unterkategorien werden durch einfache, verständliche und dazu passende Icons

3. Problemlösung und Konzepte

dargestellt. Icons sind unabhängig von der Farbe der Kategorie immer weiß, wodurch ein guter Kontrast erzielt wird. Sowohl Name der POIs als auch Entfernungen und Bewertungen sind nicht zu klein oder groß, sondern gut erkennbar.

In Wikitude ist das Zusammenspiel der einzelnen angezeigten Komponenten in den virtuellen Objekten im Vergleich zu junaio nicht ganz so optimal umgesetzt. Die Icons oder Bilder werden zu klein abgebildet. Dies führt zu der Problematik, dass nicht genau erkennbar ist, was sich auf den Bildern befindet. Außerdem ist hier die Schriftgröße der Namen der Interessenspunkte im Vergleich zu den Icons oder Fotos zu groß.

Wie schon in der Beschreibung von yelp (Abschnitt 3.1.1) erwähnt, wird in dieser Applikation sehr viel Wert auf die Optik der POIs gelegt. Es ist die einzige in dieser Arbeit erwähnte Applikation, die mit dreidimensionalen Effekten arbeitet. Einziger Kritikpunkt ist, dass die Schriftgröße der Namen der POIs und die Größe der Bewertungsskala im Vergleich zur Entfernung oder dem Status der Öffnungszeiten zu groß sind. Aufgrund dessen wird hier nicht die Bestnote vergeben.

Im Gegensatz zu den anderen Applikationen weckt das minimalistische Design der virtuellen Objekte in mixare deutlich weniger Interesse, wodurch es schließlich die schlechteste Note erhält.

In der Kategorie Größe der POIs schneidet ebenfalls wieder junaio mit der Bestnote ab, denn die Größe der POIs ist sehr gut gewählt - nicht zu klein, aber groß genug, um die relevanten Informationen darin übersichtlich anzuzeigen. Außerdem können so viele POIs im Display angezeigt werden.

Wikitude liegt hier im Mittelmaß. Die virtuellen Objekte sind zu lang. Die Länge könnte gekürzt werden, denn so hätte es immer noch genug Platz, um alle Informationen darstellen zu können.

yelp und mixare erhalten beide die Note 4. Bei beiden fallen die virtuellen Objekte sehr groß aus, wodurch nicht sehr viele POIs angezeigt werden können.

Größenunterschiede von POIs existieren nur in den Applikationen junaio und mixare. Beide erhalten die Note 3. Während bei junaio nur minimale Größenunterschiede erkennbar sind, sind diese bei mixare zu groß. Eine Mischung beider Varianten wäre optimal.

3.1. Untersuchung bereits existierender Applikationen

Die nächsten Benotungen erfolgen in der Kategorie Information. Die Applikationen Wikitude, junaio und mixare schneiden dabei gut ab.

Die virtuellen Objekte in Wikitude und mixare enthalten Informationen über den Name der Interessenspunkte sowie die Distanz von Benutzer zu diesen. Eine Anzeige der einzuordnenden Kategorie wäre hierbei als Ergänzung gut, damit der Benutzer POIs sofort einordnen und auch für ihn unbekannte POIs sofort zuordnen kann.

junaio enthält zwar in seinen virtuellen Objekten anhand der Icons Informationen zu den Kategorien, erhält aber dennoch nicht die Bestnote. Dies liegt an der zusätzlichen Information einer Bewertung mancher POIs. Diese Information ist für eine kleine Übersicht zu einem Interessenspunkt nicht relevant, da die Bewertung individuell und nicht auf den Benutzer speziell zugeschnitten ist.

Zusätzlich zu den Information, die in junaio in den virtuellen Objekten angezeigt werden, enthält yelp einen Status über die Öffnungszeiten, falls diese zu dem jeweiligen POI hinterlegt sind. Diese Information ist neben einer Bewertung ebenfalls für eine kleine Übersicht nicht relevant. Aus diesem Grund fällt die Bewertung von yelp in dieser Kategorie im Gegensatz zu den anderen Applikationen schlechter aus.

Zuletzt erfolgt die Bewertung der Clusterbehandlung. Wie Abschnitt 3.1.1 gezeigt hat, erfolgt eine Clusterbehandlung nur in den Applikationen Wikitude und junaio. In dieser Kategorie erhält junaio die Note 2, Wikitude hingegen eine 3.

In Wikitude sind zwei Kritikpunkte zu nennen: Durch die großflächige Anordnung der POIs innerhalb eines Clusters, können nie alle POIs in einer Ansicht gleichzeitig betrachtet werden. Zudem ist kein Schema ersichtlich, anhand dessen die Anordnung der Reihenfolge der zu einer Kreisansicht hinzugefügten POIs festgestellt werden kann.

Bei junaio wiederum wird deshalb eine Note abgezogen, da die Ansicht bei sehr vielen POIs unübersichtlich wird.

Zusammenfassend ist zu sagen, dass junaio die Applikation ist, die nach dieser Bewertung die besten Noten erhalten hat und somit am besten abschneidet. Gleich danach folgen Wikitude und yelp. mixare schneidet hier am schlechtesten ab und ist in vielen Punkten noch zu optimieren.

3.2. Eigene Konzepte

Bislang wurden in diesem Kapitel die bereits existierenden Applikationen beschrieben und anhand von fünf fest definierten Kriterien bewertet. Basierend auf dieser Analyse werden nun in einem nächsten Schritt eigene Konzepte zur Verbesserung der POI Clusterbehandlung in AREA vorgestellt.

Auf die aktuelle Darstellung der POIs in dieser Applikation sowie die dadurch zugrunde liegende Problematik wurde bereits in Kapitel 2 ausführlich eingegangen.

3.2.1. Erstes Konzept

Das erste Konzept, das im Rahmen dieser Bachelorarbeit entworfen wurde, ist wie junaio ein entfernungsbasiertes Konzept.

In AREA wird bisher nur der Name des POI angezeigt. Aufgrund der Analyse der bisherigen Konzepte erscheint das ledigliche Ausweisen des Namens der POIs als zu wenig. Deshalb wird der Name um eine weitere Information, die Anzeige der Distanz, ergänzt. Zudem wird hier das Kriterium der Größenunterschiede miteinbezogen. Die Analyse hat gezeigt, dass bereits vorhandene Applikationen diesen Aspekt berücksichtigen und umsetzen - die Umsetzung jedoch noch optimiert werden kann.

Im Zentrum des neu entwickelten Konzepts steht allerdings die Behandlung von Clustern. Die Idee der neuen Positionierung bei Überdeckung von POIs wurde dabei aus junaio aufgegriffen. Die Darstellung dort hat gezeigt, dass die Ansicht bei zu vielen POIs relativ schnell unübersichtlich werden kann. Eine Optimierung findet daher durch die Einführung einer minimal einstellbaren Distanz statt.

Im Folgenden wird anhand von Abbildungen das erste Konzept detaillierter beschrieben. Die virtuellen Objekte sollen bei größerer Entfernung kleiner werden. Zur Verdeutlichung und zum späteren Vergleich dieser Situation wird die ursprüngliche Situation aus Abbildung 2.3 nachgestellt (Abbildung 3.10).

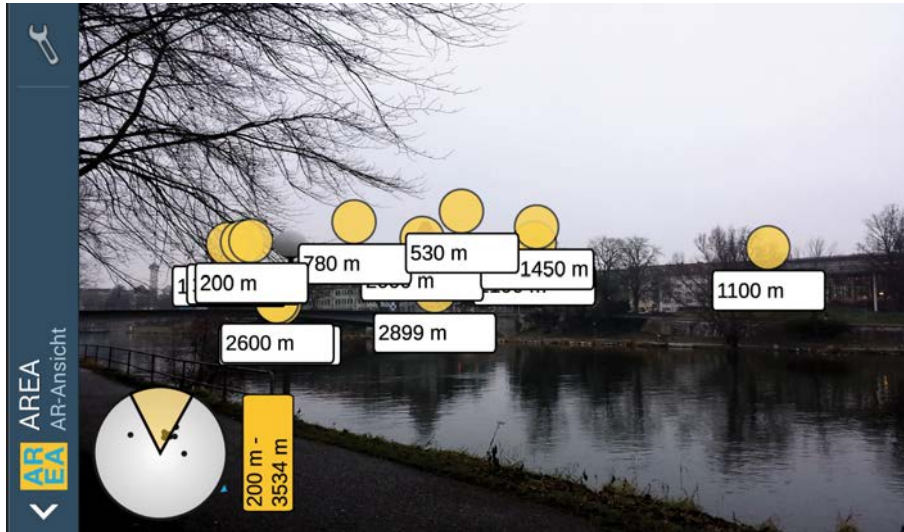


Abbildung 3.10.: Nachgestellte Darstellung der POIs in AREA (erstes Konzept)

Zur besseren Übersicht sind im Namensfeld der einzelnen POIs hier nur frei erfundene Entfernungen abzulesen. Normalerweise ist es in diesem Konzept vorgesehen, dass sowohl der Name als auch die Entfernung von Benutzer zu den einzelnen POIs in diesem Feld angegeben sind. Zu beachten ist, dass POIs, die sich näher am Standort des Benutzers befinden, im Vordergrund liegen, während weiter entfernte teilweise oder sogar komplett überdeckt werden.

In Abbildung 3.11 ist nun veranschaulicht, wie die POIs nach diesem Konzept angeordnet werden. Im Gegensatz zu den POIs in Abbildung 3.10 ist zu erkennen, dass sich jene, die weiter in der Ferne liegen, kleiner werden. Die unterschiedlichen Farben, die je nach Distanz variieren, heben dies zusätzlich hervor. Von nah nach fern erfolgt die Reihenfolge: gelb, orange, rot, grün, blau und zuletzt lila.

Eine weitere Änderung ist die neue Positionierung von POIs, die von anderen verdeckt werden. Diese werden nun leicht nach oben verlegt, damit der Benutzer auch mit diesen interagieren und sich somit nähere Informationen zu diesen POIs einholen kann.

Objekte werden hier als verdeckt betrachtet, wenn der Abstand der Mittelpunkte zweier POIs kleiner ist als eine berechnete Distanz. Die genaue Erläuterung der Berechnung dieser Distanz erfolgt in Abschnitt 4.1.

3. Problemlösung und Konzepte

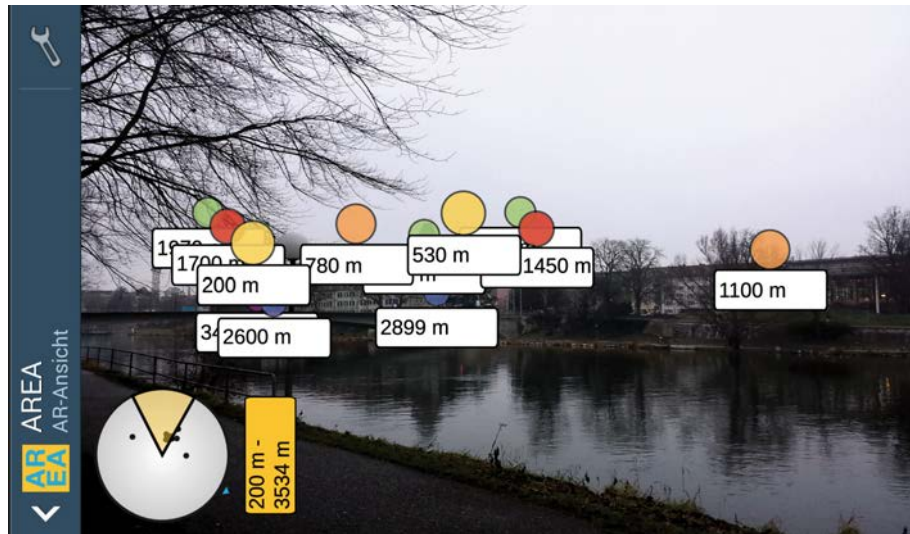


Abbildung 3.11.: Darstellung der POIs (erstes Konzept)

Der letzte Unterschied zum bisherigen Modell von AREA zu diesem Konzept besteht darin, dass noch ein weiterer Parameter bei der Einstellung der Distanz festgelegt werden kann.

Bisher war es dem Benutzer nur möglich, einen maximalen Radius einzustellen, innerhalb dessen die POIs in der Umgebung angezeigt werden (siehe Abbildung 2.2 (rechts)). Es soll nun aber zusätzlich möglich sein, eine minimale Distanz, ausgehend bei 0 Meter, einzustellen. Der Benutzer kann so beispielsweise nur POIs betrachten, die sich innerhalb eines Radius von 200 bis 3534 Meter befinden. POIs, deren Distanz weniger als 200 und mehr als 3534 Meter beträgt, werden nun nicht mehr dargestellt.

3.2.2. Zweites Konzept

Das zweite entworfene Konzept besitzt einen ähnlichen Ansatz wie Wikitude. Bilden viele POIs einen Cluster, so werden diese zusammengefasst. Jedoch hat die Bewertung in Absatz 3.1.2 gezeigt, dass die Darstellung und Anordnung der POIs sowie die Funktionen der Buttons innerhalb eines Clusters in Wikitude noch Verbesserungspotenzial hat. Es wird deshalb in diesem Konzept versucht, diesen Aspekt zu optimieren, damit eine bessere Darstellungsform erzielt werden kann. Des Weiteren wird im Unterschied zu

Wikitude dieses Konzept bei Bedarf noch um eine Anzeige der gesamten Anzahl an Seiten sowie die aktuelle Seitenzahl innerhalb eines Clusters ergänzt. Zuletzt werden, aus dem selben Grund wie im ersten Konzept, die Namen der POIs um die Information der Distanz erweitert.

In folgender Abbildung 3.12 ist veranschaulicht, wie in AREA anhand dieses entworfenen Konzepts nun Cluster dargestellt werden.

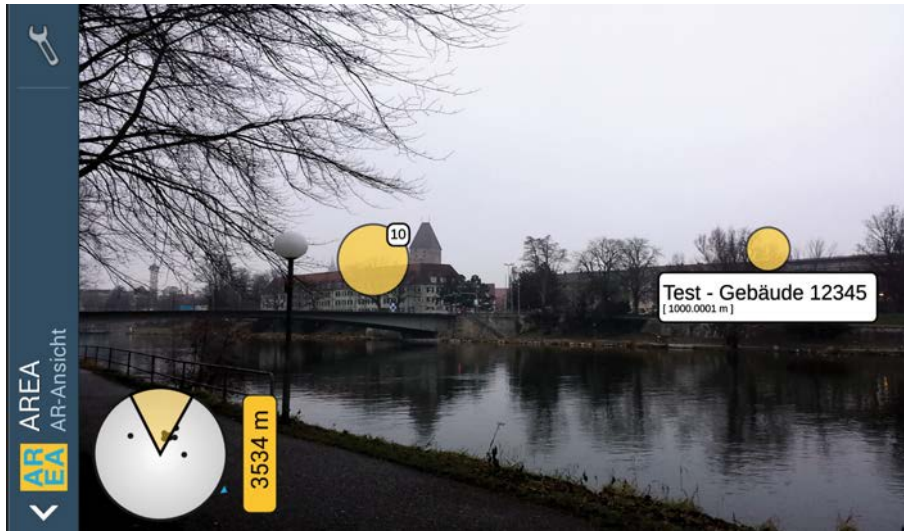


Abbildung 3.12.: Darstellung eines Clusters (zweites Konzept)

Ein gelber Kreis, der deutlich größer als die auf dem Display dargestellten POIs ist, symbolisiert hier einen Cluster. Ebenso wie in Wikitude wird die Anzahl der POIs, die sich innerhalb eines Clusters befinden, mit angezeigt. Interessiert sich der Benutzer nun für die POIs, die sich in diesem Cluster befinden, kann dieser den Kreis auf dem Bildschirm seines Geräts berühren. Dem Benutzer werden durch diese Interaktion die gewünschten POIs angezeigt (Abbildung 3.13).

Die Interessenspunkte werden innerhalb eines leicht transparenten, gelben Kreises kreisförmig angeordnet. In der Mitte dieses Kreises befindet sich ein „X-Knopf“, den der Benutzer betätigen kann, um diese Ansicht zu schließen.

3. Problemlösung und Konzepte

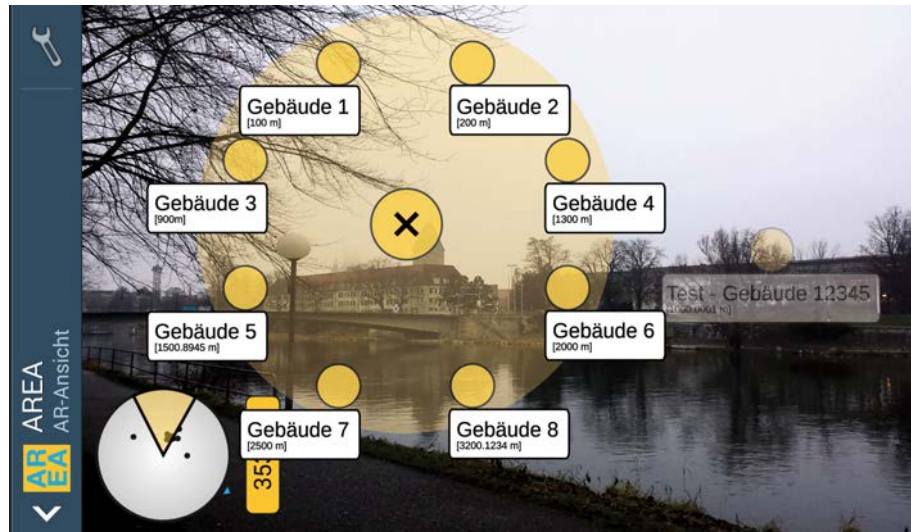


Abbildung 3.13.: Darstellung von acht POIs innerhalb eines Clusters (zweites Konzept)

Die Anordnung der POIs innerhalb des Kreises erfolgt im Gegensatz zu Wikitude immer nach einem fest vorgegebenen Prinzip, unabhängig davon, wie viele POIs sich in einem Cluster befinden. Diese werden der Ansicht von links oben nach rechts unten zugefügt, wobei links oben der Interessenspunkt angezeigt wird, der die geringste Entfernung zum Benutzer aufweist. In Abbildung 3.13 wird dies sowohl durch die fiktiven Entfernungen unterhalb der Namen der einzelnen POIs als auch durch die aufsteigende Nummerierung der Gebäude visualisiert. „Gebäude 8“ wäre in diesem Fall der Interessenspunkt, der ausgehend vom Standort des Nutzers, am weitesten entfernt liegt.

Innerhalb einer geöffneten Ansicht können maximal acht POIs dargestellt werden. Befinden sich jedoch mehr als acht POIs in einem Cluster, werden diese in mehrere Seiten unterteilt, sodass maximal sechs Interessenspunkte auf einer Seite zu sehen sind. Um die restlichen POIs anzuzeigen, werden Buttons mit darin entsprechend angezeigten Pfeilen eingeblendet, mit denen der Benutzer die Seiten wechseln kann (Abbildung 3.14). Im Vergleich zu Wikitude enthält dieses Konzept einen Button, der die Funktion des Zurückschaltens erfüllt. Diese beiden Buttons ersetzen dann im Vergleich zu Abbildung 3.13 „Gebäude 7“ und „Gebäude 8“. Bei Anzeige der ersten Seite ist der Button, der den in links Richtung zeigenden Pfeil enthält, deaktiviert. Dasselbe geschieht mit dem

3.2. Eigene Konzepte

rechten Button, mit welchem der Benutzer zur nächsten Seite gelangen kann, sobald die letzte Seite erreicht ist. Dies verhindert das direkte Wechseln von der ersten zur letzten Seite oder umgekehrt. Außerdem kann der Benutzer zu einer besseren Übersicht zwischen den Pfeil-Buttons die aktuelle Seitenzahl sowie die Gesamtzahl aller Seiten ablesen.

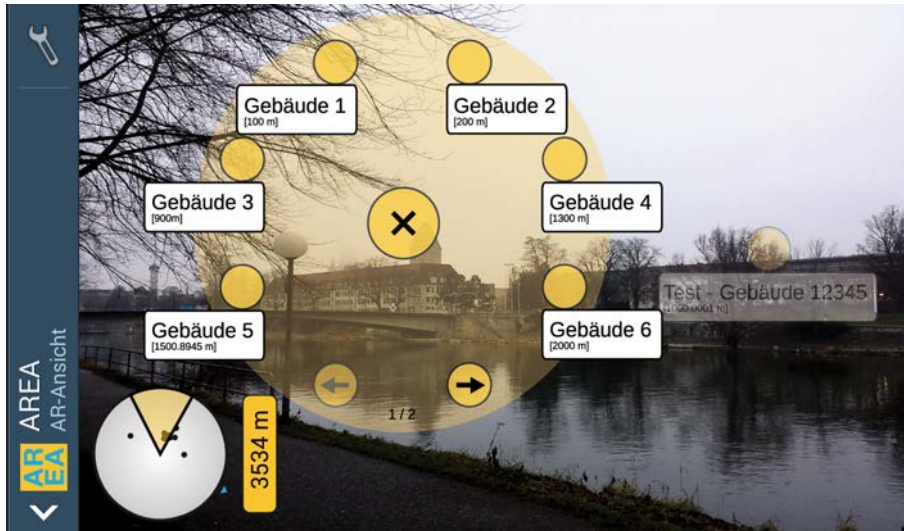


Abbildung 3.14.: Buttons mit Pfeilen und Darstellung der Seitenzahl (zweites Konzept)

Sobald sich die kreisförmige Ansicht öffnet, werden die restlichen POIs auf dem Display nahezu ausgeblendet, damit der Fokus auf die POIs innerhalb des Clusters gerichtet wird. Mit fast ausgeblendeten POIs kann der Benutzer nicht mehr interagieren. Durch das Schließen der Clusteransicht werden alle POIs wieder wie zuvor dargestellt.

4

Implementierung

AREA wird in diesem Kapitel um die prototypische Implementierung beider zuvor vorgestellter Konzepte (Unterkapitel 3.2) erweitert. Entwickelt wird in der Programmiersprache Java von Oracle [21]. Das Betriebssystem ist Android¹. Darüber hinaus wird die entsprechende Android API [23], auch „application programming interface“ genannt, verwendet. Auf den Architekturentwurf, die Klassenstruktur, den Kommunikationsablauf und die Datenhaltung wird hier nicht eingegangen. Dies war Bestandteil einer anderen Bachelorarbeit und kann dort nachgelesen werden [7]. In den beiden folgenden Unterkapiteln wird auf die wichtigsten Änderungen und Neuerungen von AREA eingegangen.

¹Die Prototypen wurden beide auf einem Samsung Galaxy S4 [22] mit der Android-Version 4.4.2 erfolgreich getestet und ausgeführt.

4. Implementierung

4.1. Implementierung des ersten Konzeptes

Die Hauptfunktionen des ersten Konzeptes wurden bereits in Absatz 3.2.1 erläutert.

Die Implementierung dieses entfernungsbasierten Konzeptes besteht im Grunde aus drei Schritten: Zuerst wird AREA um die Einstellung einer minimalen Distanz erweitert. Danach erfolgt die Anpassung der Größen und Farben der Kreise der POIs je nach Distanz. Zuletzt werden POIs, die durch andere verdeckt sind, neu positioniert, sodass der Benutzer auch mit diesen interagieren kann.

Damit eine minimale Distanz eingestellt werden kann, wird bei der Einstellung der Distanz eine neue `SeekBar` [23], ein Schieberegler, hinzugefügt. Diese wird in der Methode `showDistancePopup()` in der Klasse `AREAMainActivity` erstellt, in welcher auch der bereits vorhandene `distanceSlider` zur Einstellung des aktuellen Radius implementiert wurde. Damit eine minimale Distanz ab 0 Meter eingestellt werden kann, wird der Wert der Variable `kMinDistance` in der Klasse `AREAConstants` von 1000 auf 0 gesetzt. Durch das Einstellen einer minimalen Distanz muss beachtet werden, dass nur die Interessenspunkte angezeigt werden, die sich innerhalb der minimalen und maximalen Distanz befinden. Auch im Radar sollen nur noch die Punkte innerhalb der eingestellten Distanz angezeigt werden. Dazu wird unter anderem in der `setRadarPoints`-Methode in der Klasse `AREARadarView` eine `if`-Anweisung eingeführt, durch die nur noch die POIs im Radar angezeigt werden, die sich innerhalb der minimalen und maximalen Distanz befinden (Abbildung 4.1).

```
1 if (poi.getDistance() >= minDistance  
2    && poi.getDistance() <= distance) {...}
```

Abbildung 4.1.: `if`-Anweisung zur Eingrenzung der angezeigten POIs im Radar

Damit nur noch die POIs auf dem Display des Benutzers angezeigt werden, die sich innerhalb der eingestellten minimalen und maximalen Distanz befinden, müssen in der Klasse `AREALocationController` Änderungen vorgenommen werden. Um festzustellen, ob sich Interessenspunkte innerhalb der eingestellten Entfernung befinden, wird die `if`-Anweisung in der Methode `onLocationChanged` geändert (Abbildung 4.2). In

4.1. Implementierung des ersten Konzepts

der Kommentarzeile befindet sich die ursprüngliche `if`-Anweisung vor Einführung der minimalen Distanz.

```
1 if (distance >= minDistance && distance <= radius) {...}
2 //if (distance <= radius && distance >= 0) {...}
```

Abbildung 4.2.: `if`-Anweisung zur Eingrenzung der angezeigten POIs im Display

Zudem werden, analog zu den vorhandenen Methoden `setRadius` und `getRadius`, die Methoden `setMinDistance` und `getMinDistance` implementiert.

Das Zeichnen der POIs erfolgt in der Klasse `AREAPointOfInterestView`. Um die Größe und Farbe der Kreise zu ändern, werden `circleSize` und `circleFill` je nach Distanz verändert. In Abbildung 4.3 ist ein Teil der Methode `circleFillAndSize` abgebildet, welche für das Berechnen der Größe und Farbe der POI-Kreise verantwortlich ist.

```
1 public void circleFillAndSize() {
2     int size = (int) maxRadius - (int) minRadius;
3     int stepSize = size / 5;
4     int poiDistance = (int) poi.getDistance();
5     int circleSizePixel = 30;
6     if (poiDistance < ((int) minRadius + stepSize)) {
7         // 100% - gelb
8         circleSize = dpToPixel(circleSizePixel);
9         circleFill.setColor(Color.argb(179, 249, 196, 49));
10    } else if (poiDistance < ((int) minRadius + stepSize * 2)
11    && poiDistance >= ((int) minRadius + stepSize)) {
12        // 90% - orange
13        circleSize = dpToPixel((int) (circleSizePixel * 0.9));
14        circleFill.setColor(Color.argb(179, 238, 154, 0));
15    } else if (poiDistance < ((int) minRadius + stepSize * 3)
16    && poiDistance >= ((int) minRadius + stepSize * 2)) {
17        // 80% - rot
18        circleSize = dpToPixel((int) (circleSizePixel * 0.8));
19        circleFill.setColor(Color.argb(179, 238, 64, 000));
20    } ...
21 }
```

Abbildung 4.3.: Ausschnitt der Methode zur Berechnung von Größe und Farbe der Kreise

4. Implementierung

Es gibt insgesamt sechs Größen und Farben, in welchen POIs gezeichnet werden können. Interessenspunkte werden je nach eingestellter minimaler und maximaler Distanz in einen von sechs Bereichen eingeteilt, in welchen dann auch die Größe und Farbe gesetzt wird.

Um zuletzt das Problem der Clusterbildung zu lösen, werden die POIs in diesem Falle neu positioniert. Dies erfolgt in der Methode `onHeadingWithLocationsChanged` der Klasse `AREAMainActivity`. In Abbildung 4.4 ist ein Teil der Berechnung der Neupositionierung zu sehen.

```
1 // Abstand der Mittelpunkte berechnen
2 float distance = (float) Math.sqrt((Math.pow((x - secondX), 2))
3                               + (Math.pow((y - secondY), 2)));
4 if (distance < (radius + secondRadius + dpToPixel(20))) {
5     secondView.repositionPOI();
6 }
```

Abbildung 4.4.: Überprüfung auf verdeckte POIs

Es werden die auf dem Display geografischen Positionen zweier Interessenspunkte miteinander verglichen. Dazu werden je die x- und y-Koordinaten dieser Punkte benötigt. Danach wird der Abstand der Mittelpunkte berechnet. Ist dieser echt kleiner als die Radien der beiden Punkte zusammen addiert plus einem festen Wert, wird ein POI durch einen anderen, näher liegenden verdeckt. In diesem Fall wird der verdeckte Punkt neu positioniert. Dies geschieht durch die neue Methode `repositionPOI` in der Klasse `AREAPointOfInterestView` (Abbildung 4.5).

```
1 public void repositionPOI() {
2     Point newPoint = new Point(poi.getPoint().x - dpToPixel(40),
3                               poi.getPoint().y - dpToPixel(40));
4     poi.setPoint(newPoint);
5     this.setX(poi.getPoint().x);
6     this.setY(poi.getPoint().y);
7 }
```

Abbildung 4.5.: Neues Positionieren eines POI

4.2. Implementierung des zweiten Konzepts

Bei dem Interessenspunkt, der neu positioniert wird, werden die x- und y-Koordinaten neu gesetzt. Dem x- und y-Wert wird jeweils ein fester Wert abgezogen, sodass der Interessenspunkt leicht nach links oben versetzt wird.

Dies ist so allerdings nur auf die vertikale Ausrichtung des Gerätes, auf dem die Applikation AREA installiert ist, ausgerichtet.

4.2. Implementierung des zweiten Konzepts

In der Implementierung des zweiten Konzepts werden Änderungen und Erweiterungen nur in der View-Schicht sowie in der Klasse `AREAMainActivity` vorgenommen.

Zuerst wird in der Klasse `AREAMainActivity` eine `ArrayList` [23] erstellt, die nur Objekte vom Typ `AREAPointOfInterest` enthalten darf. Dieser Liste werden alle POIs, die sich in einem Cluster befinden, hinzugefügt. Dazu werden, wie in der Implementierung des ersten Konzepts, die geografischen Positionen zweier POIs auf dem Display miteinander verglichen. Ist der Abstand der Mittelpunkte kleiner als ein fest vorgegebener Wert, werden beide Punkte der `ArrayList` zugefügt. Danach wird überprüft, ob es noch weitere Punkte gibt, deren Abstand der Mittelpunkte zum ersten der Liste hinzugefügten Interessenspunkt echt kleiner als der vorgegebene Wert ist. Ist dies der Fall, werden diese POIs ebenfalls der `ArrayList` hinzugefügt. Nachdem alle POIs mit dem ersten in der Liste verglichen wurden, wird ein neuer `AREAClusterPointView` mit dem geografischen Punkt des ersten Elements in der Liste und der Anzahl der POIs in dieser erzeugt. Daraufhin wird der `AREAClusterPointView` der `locationView` hinzugefügt und die Cluster-Suche wird dabei beendet. Die Begründung hierzu erfolgt am Ende dieses Unterkapitels. Die `AREAClusterPointView` - Klasse ist dafür zuständig, einen großen Kreis mit der Anzahl der POIs im Cluster zu zeichnen (siehe Darstellung in Abbildung 3.12). Eine neue Methode `clusterMenu` in der Klasse `AREAMainActivity` wird aufgerufen, sobald mit diesem Kreis interagiert wird. Zu Beginn dieser Methode wird eine neue Instanz der neu erstellten Klasse `AREACircleView` erzeugt. Die genaue Erläuterung der Funktion dieser Klasse erfolgt an späterer Stelle.

4. Implementierung

Danach werden zu jedem Interessenspunkt im Cluster Objekte der Klasse `AREAPointOfInterestViewForClusterView` erstellt. Diese Klasse ist bis auf kleine Teile identisch der `AREAPointOfInterestView`-Klasse von AREA. Beide Klassen werden um die Anzeige der Entfernung der POIs zum Standort des Nutzers zusätzlich zum Namen ergänzt und die einzigen Unterschiede bestehen darin, dass die Breite des Textfeldes fest vorgegeben wird und zwei Methoden entfernt werden. Die neuen Objekte werden einer `ArrayList` hinzugefügt, die daraufhin so sortiert wird, dass sich der Interessenspunkt, der die geringste Distanz aufweist, an erster Stelle befindet. Dementsprechend befindet sich der POI mit größter Distanz an letzter Position in dieser Liste. Dies hat zur Folge, dass später die POIs in der Kreisansicht nach den einzelnen Entfernungen sortiert dargestellt werden. Befinden sich weniger oder genau acht Objekte in dieser Liste, werden diese der Instanz von `AREACircleView` zugefügt und angezeigt. Bei mehr als acht Objekten in der Liste wird die neu implementierte Methode `showDistinctViews` aufgerufen. Diese ist für die Berechnung der anzuzeigenden Interessenspunkte und das Hinzufügen derer auf der richtigen Kreisansicht-Seite verantwortlich. Außerdem werden in dieser Methode noch zwei `Buttons`, inklusive Berücksichtigung der Aktivierung oder Deaktivierung derer, und ein `TextView` [23] für das Anzeigen der aktuellen Seite zusammen mit der gesamten Seitenzahl erstellt. Beide `Buttons` sind jeweils Instanzen der hinzugefügten Klasse `AREAforwardBackButton`. In dieser werden Kreise erstellt, die identisch mit denen der POIs sind, zusätzlich aber noch ein „v“ für „vor“ oder ein „z“ für „zurück“ beinhalten.

In Abbildung 4.6 ist ein Teil des `if-else`-Konstrukts der gerade erläuterten Methode `showDistinctViews` abgebildet, das für das Hinzufügen der richtigen POIs auf der passenden Seite verantwortlich ist. Zu beachten ist hierbei, dass nicht immer genau sechs POIs in einer Ansicht vorhanden sind. Angenommen es befinden sich zehn POIs im Cluster: Auf der ersten Seite würden dann sechs POIs angezeigt werden, auf der zweiten dementsprechend vier. Daher muss berechnet werden, auf welcher Seite sich der Nutzer befindet und wie viele POIs darin abgebildet werden.

4.2. Implementierung des zweiten Konzepts

```
1 if (poiInClusterList.size() < pageNumber * 6) {
2     for (int i = (pageNumber - 1) * 6;
3         i < poiInClusterList.size(); i++) {
4         poiInClusterList.get(i).setOnClickListener(
5             ...
6         menu.addView(poiInClusterList.get(i));
7     }
8     for (int i = poiInClusterList.size();
9         i < pageNumber * 6; i++) {
10        View emptyView = new View(this);
11        menu.addView(emptyView);
12    }
13 } else {
14     for (int i = (pageNumber - 1) * 6; i < pageNumber * 6; i++){
15        poiInClusterList.get(i).setOnClickListener(
16            ...
17        menu.addView(poiInClusterList.get(i));
18    }
19 }
```

Abbildung 4.6.: Teil der Berechnung der POIs auf aktueller Seite

Die vorher bereits erwähnte Klasse `AREACircleView` ist für das Anordnen der Objekte im neuen Fenster zuständig (Abbildung 4.7). Diese Klasse erbt von der Klasse `ViewGroup`, welche eine besondere `View`-Klasse ist, die andere `Views` enthalten kann [23]. Die Methode `onLayout` ist für das Positionieren der `children` im `Layout` verantwortlich. Jeder `View` wird abhängig davon, wann er dem `Layout` hinzugefügt wurde, an der richtigen Stelle platziert. Die Positionierung erfolgt kreisförmig von links oben nach rechts unten, wie dies in Absatz 3.2.2 erläutert wurde.

4. Implementierung

```
1 public class AREACircleView extends ViewGroup{
2     ...
3
4     @Override
5     protected void onLayout(boolean changed, int l, int t,
6                             int r, int b) {
7         int width = r/16;
8         int height = b/12;
9
10        int childCount = getChildCount();
11        for (int i = 0; i < childCount; i++) {
12            View child = getChildAt(i);
13
14            if (i == 0) {
15                child.layout(5*width-150, 3*height-90,
16                             5*width+150, 3*height+120);
17            } else if (i == 1) {
18                child.layout(11*width-150, 3*height-90,
19                             11*width+150, 3*height+120);
20            } else if (i == 2) {
21                child.layout(3*width-150, 5*height-90,
22                             3*width+150, 5*height+120);
23            } else if (i == 3) {
24                child.layout(13*width-150, 5*height-90,
25                             13*width+150, 5*height+120);
26            } ...
27        }
28    }
29 }
```

Abbildung 4.7.: Zuständige Klasse für die Anordnung der Objekte

Die Implementierung des Prototypen dieses Konzepts berücksichtigt nicht die exakte Berechnung von Clustern. Es wird lediglich ein Cluster, wie anfangs in diesem Unterkapitel beschrieben, gebildet. Zudem werden jene POIs, die sich in diesem Cluster befinden, nicht von der `locationView` entfernt, sondern werden immer noch angezeigt. Die Position des der `locationView` hinzugefügten, clusterbildenden Kreises wird nicht wie die POIs auf dem Display neu berechnet - dieser bleibt an der Position, an der er hinzugefügt wurde, bestehen.

4.2. Implementierung des zweiten Konzepts

Im Zuge des Implementieren eines Algorithmus zur exakten Berechnung von Clustern, der nicht Teil dieser Arbeit ist, können die eben genannten Punkte mit eingebracht und optimiert werden. Es existieren bereits dichte-basierte Clusteralgorithmen, die im Stande sind, Cluster zu erkennen [24] .

5

Vorstellung der POI Clusterbehandlung nach Umsetzung der beiden Konzepte

In diesem Kapitel wird die Applikation AREA nach der Implementierung der beiden Prototypen vorgestellt. Diese werden mit der Hilfe von Screenshots erläutert. Danach erfolgt eine Bewertung der eigens erstellten und implementierten Konzepte.

5.1. Vorstellung des ersten Konzepts

In Abbildung 5.1 sind zwei Screenshots abgebildet. Diese Abbildung dient der direkten Gegenüberstellung der ursprünglichen Applikation ohne die POI Clusterbehandlung (links) und der Applikation nach der Implementierung des ersten Konzepts (rechts).

5. Vorstellung der POI Clusterbehandlung nach Umsetzung der beiden Konzepte

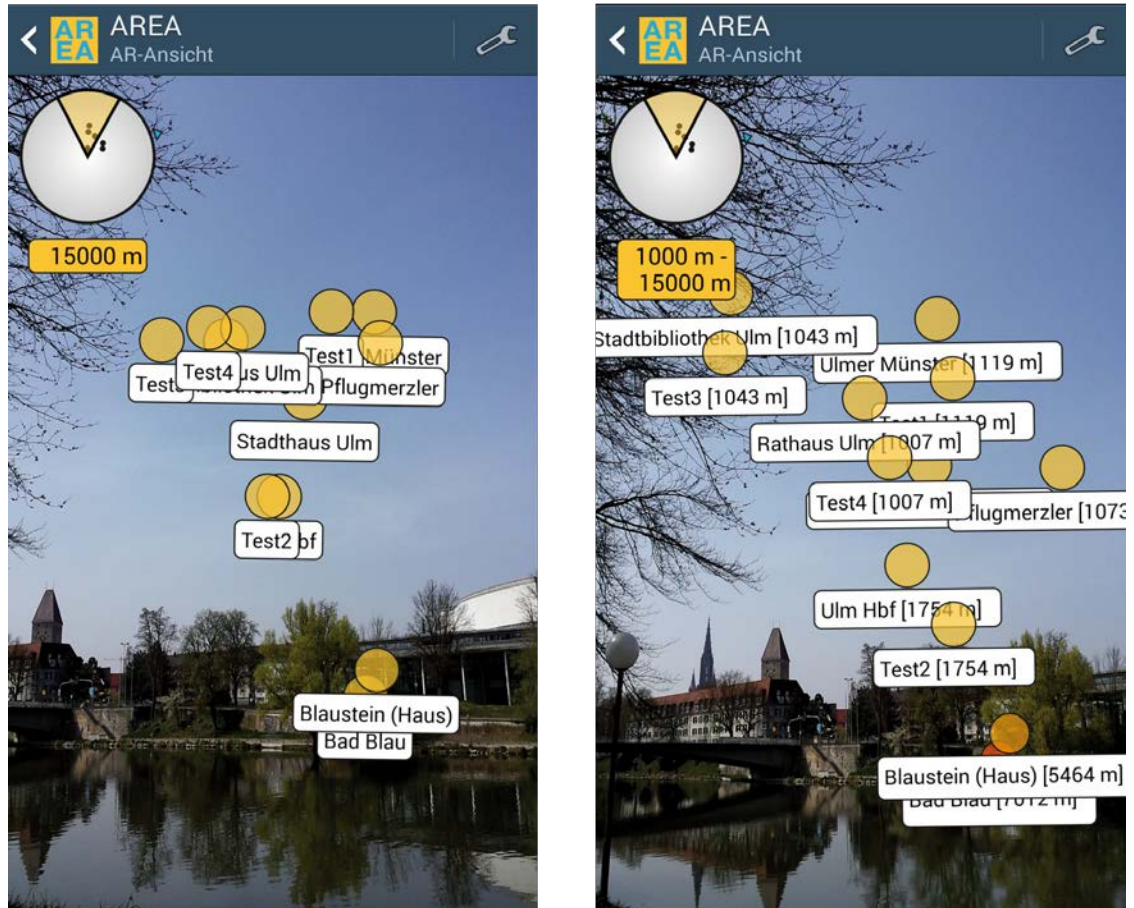


Abbildung 5.1.: Screenshots AREA (Konzept 1, Vorher - Nachher)

Zu erkennen ist, dass einzelne Interessenspunkte nach Umsetzen des ersten Entwurfs anders angeordnet sind. Durch die Neupositionierung von verdeckten POIs kann der Benutzer dieser Augmented Reality Applikation nun auch mit denjenigen POI interagieren, die zuvor durch andere verdeckt wurden. Wie bereits in Absatz 4.1 werden diese POIs leicht nach links oben versetzt.

Ein weiterer Unterschied besteht im Namensfeld der POIs. Es ist nicht mehr nur der Name der einzelnen POIs abzulesen, sondern auch die Distanz von Benutzer zu den Interessenspunkten in Meter. Außerdem ist zu erkennen, dass sich die Größe und Farbe der Kreise der beiden sich im Screenshot rechts unten befindlichen Interessenspunkte geändert hat, da diese weiter entfernt sind. Die Größe und Farbe ändert sich nach

5.1. Vorstellung des ersten Konzepts

eingestellter minimaler und maximaler Distanz (siehe Absatz 4.1). Deutlicher ist dies in Abbildung 5.2 durch zwei weitere Screenshots veranschaulicht.

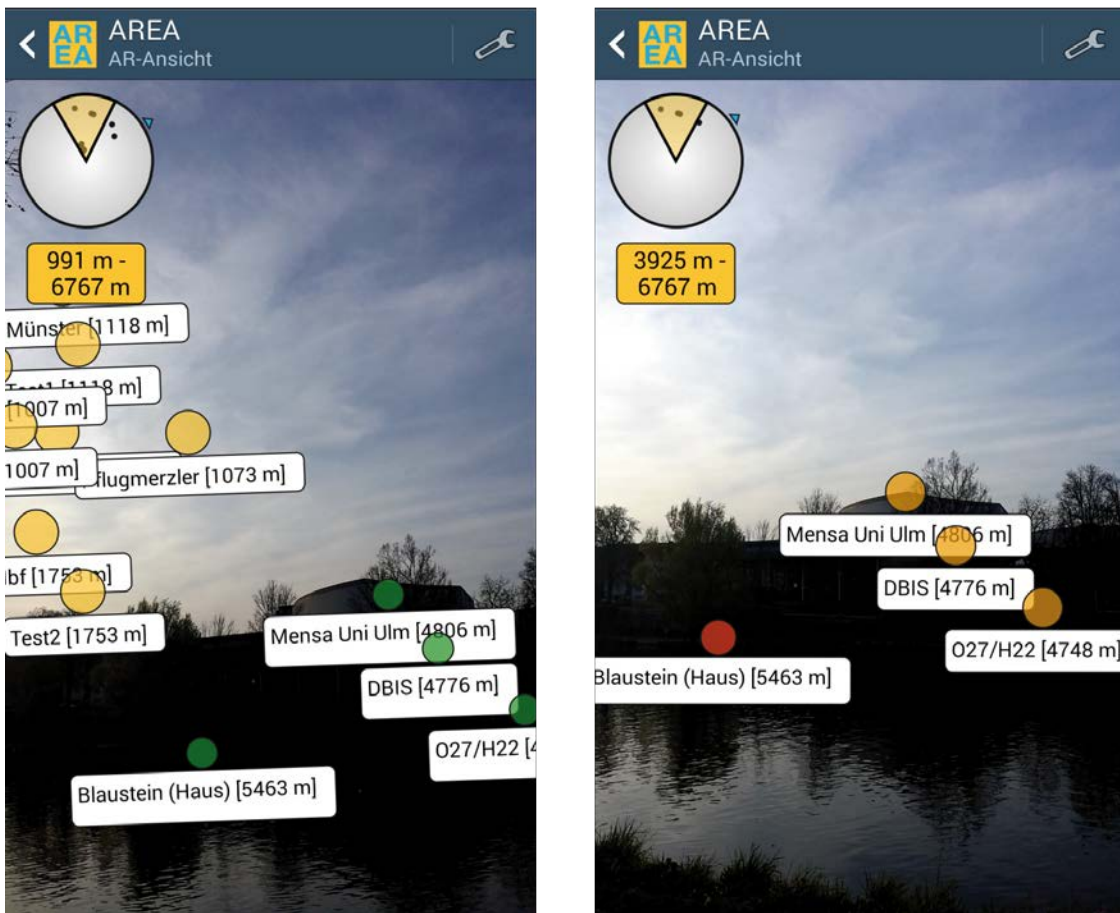


Abbildung 5.2.: Screenshots AREA (Konzept 1, unterschiedliche Distanzen)

Es sind jeweils zwei unterschiedliche Minimaldistanzen eingestellt, welche unterhalb des Radars abzulesen sind. Auf dem linken Bild sind mehr POIs als auf dem rechten erkennbar, da auf dem rechten die POIs, deren Distanz geringer als 3925 Meter ist, nicht mehr angezeigt werden.

In Abbildung 5.3 ist das in diesem Konzept veränderte Fenster zur Einstellung der Distanzen veranschaulicht. Es kann hier anhand von Schiebereglern die minimale und maximale Distanz eingestellt und verändert werden.

5. Vorstellung der POI Clusterbehandlung nach Umsetzung der beiden Konzepte

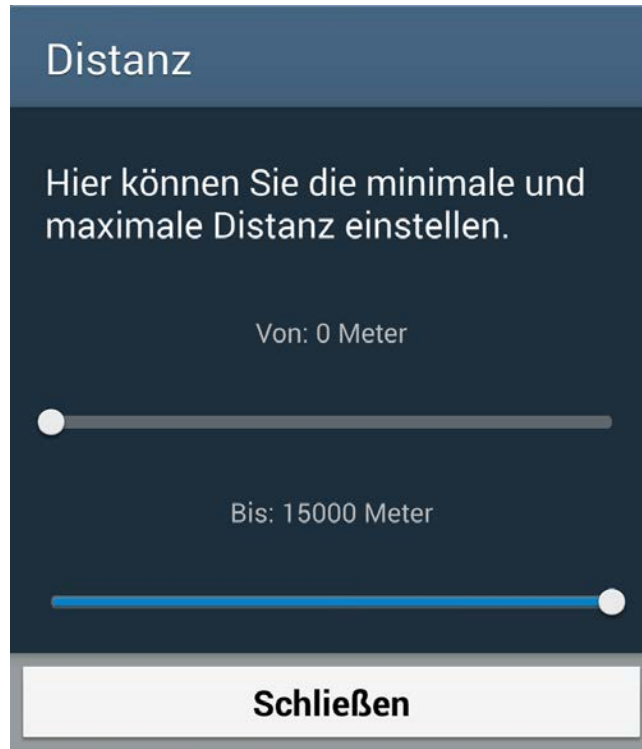


Abbildung 5.3.: Einstellung der Distanzen (Konzept 1)

5.2. Vorstellung des zweiten Konzepts

Nachdem nun das erste Konzept vorgestellt wurde, wird in diesem Abschnitt das zweite umgesetzte Konzept veranschaulicht.

In Abbildung 5.4 ist die AR Ansicht abgebildet. Darauf ist ein großer dunkelroter Kreis hinter diversen POIs zu sehen. Dieser Kreis verdeutlicht einen Cluster, der mehrere POIs zusammenfasst. Dieser wird hier rot dargestellt, damit er sich von den davor liegenden POIs farblich abhebt und dadurch besser zu erkennen ist. Zusätzlich zu diesem Kreis wird die Anzahl der POIs, die sich in diesem Cluster befinden, mit angezeigt. In Abschnitt 4.2 wurde bereits erwähnt, dass die POIs, die sich in einem Cluster befinden, nicht entfernt werden und insgesamt nur ein Cluster gebildet wird, welches eine feste Position einnimmt.

5.2. Vorstellung des zweiten Konzepts

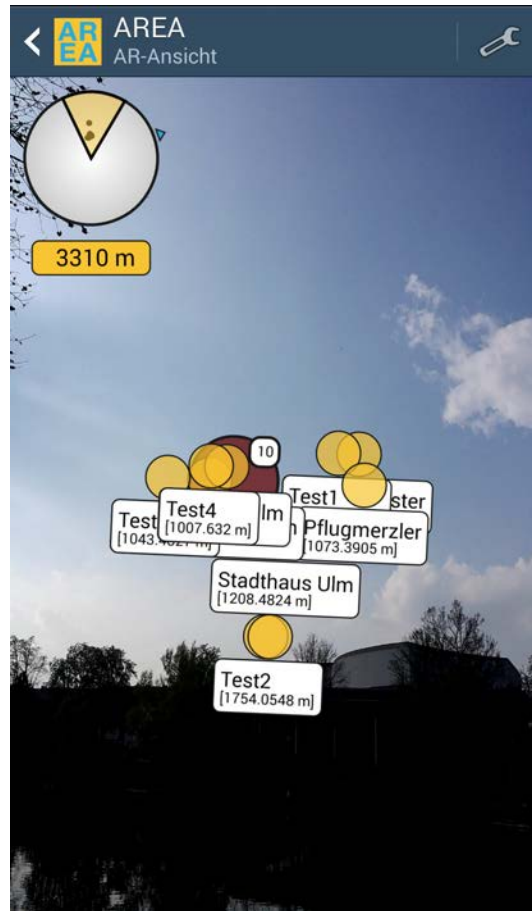


Abbildung 5.4.: Screenshot AREA (Konzept 2)

Insgesamt werden im Screenshot zehn POIs zusammengefasst. Berührt der Benutzer den dunkelroten Kreis, öffnet sich eine neue Ansicht. Diese Ansicht ist auf den Screenshots in Abbildung 5.5 zu sehen.

Da sich mehr als acht POIs im Cluster befinden, werden in einer Ansicht nur maximal sechs POIs dargestellt und es erscheinen zwei Buttons, mit denen man die Seiten wechseln kann, wobei hier je einer dieser beiden deaktiviert ist (siehe Abschnitt 3.2.2). Hier entspricht der runde, gelbe Button mit dem darin befindlichen „z“ dem Zurück-Button und der andere entsprechend dem Vorwärts-Button („v“). Zudem ist zwischen diesen Buttons die aktuelle Seitenzahl sowie die Gesamtzahl aller Seiten abzulesen.

5. Vorstellung der POI Clusterbehandlung nach Umsetzung der beiden Konzepte

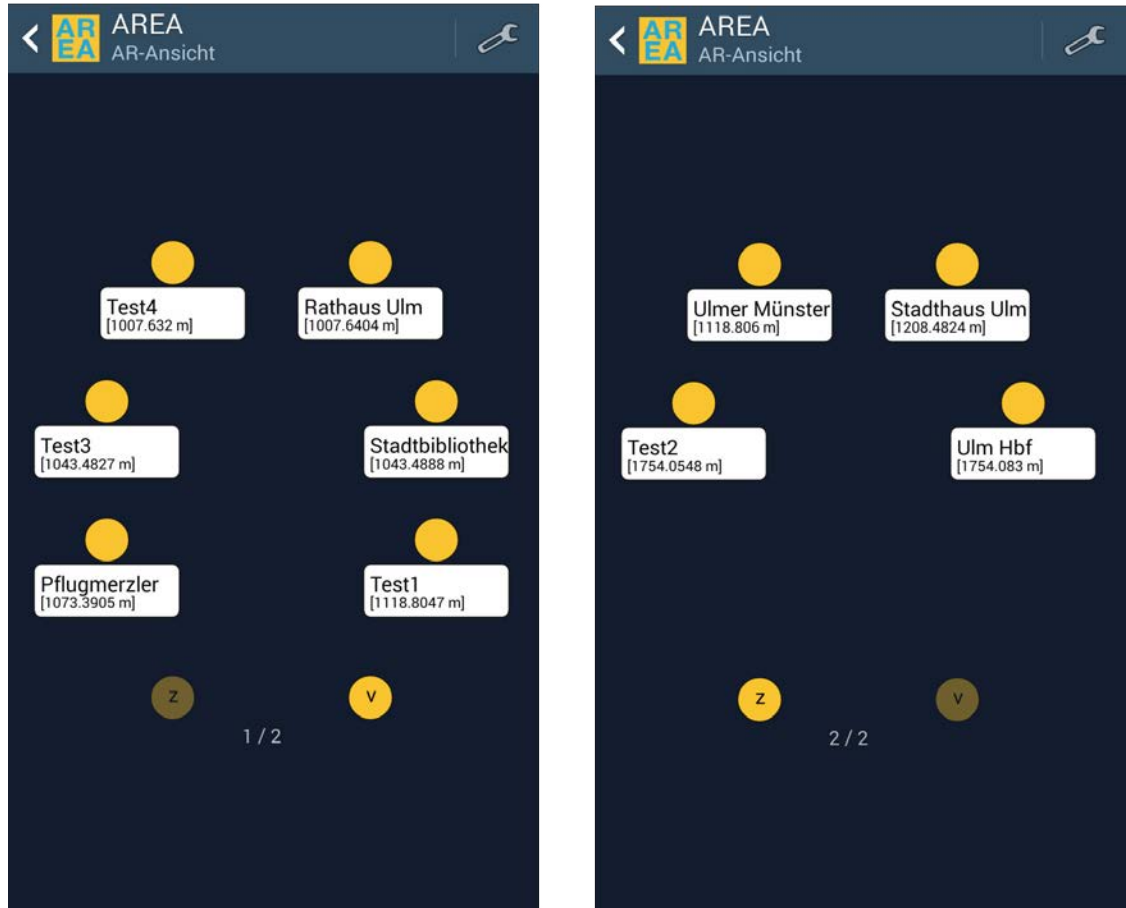


Abbildung 5.5.: Screenshots AREA - Cluster Kreisansicht (Konzept 2)

Die Interessenspunkte sind, wie in Abschnitt 3.2.2 erläutert und auch in diesen Screenshots erkenntlich, nach der Entfernung zum Nutzer sortiert.

5.3. Bewertung der eigenen Konzepte

Neben der Clusterbehandlung, welche den Schwerpunkt der Arbeit bildete, wurden auch die Kriterien „Größenunterschiede“ und „Information“ betrachtet. Die Kriterien „Design“ und „Größe“ können hier nicht betrachtet werden, da im Rahmen dieser Arbeit eine Optimierung nicht behandelt wurde.

Die zu Konzept 1 vergleichbare Applikation stellt junaio dar.

Im Vergleich zu dieser werden in Konzept 1, dem entfernungsbasierten Entwurf, die Größenunterschiede der einzelnen POIs stärker wahrgenommen. Verstärkt wird dieser Effekt durch die Hinzuziehung von Farben: Auf einen Blick kann der Nutzer anhand der Farbgebung erahnen, welche Interessenspunkte sich in welcher Entfernung befinden. Konzept 1 präsentiert weniger Informationen als junaio - lediglich Name und Entfernung werden angezeigt. In einem weiteren Optimierungsschritt könnte hier die Einführung von Kategorien in Betracht gezogen werden.

Durch das Einführen eines Schiebereglers für die minimale Distanz kann im Vergleich zu junaio das Problem der Unübersichtlichkeit bei zu vielen POIs entzerrt werden. Die Anzahl von POIs wird durch den eingestellten Distanzbereich minimiert und eingeschränkt.

Als Pendant zu Konzept 2 kann die Applikation Wikitude gesehen werden. Größenunterschiede spielen sowohl in Wikitude als auch in Konzept 2 keine Rolle.

Konzept 2 enthält exakt die gleichen Informationen wie Konzept 1 - Name und Entfernung. Wikitude enthält im Vergleich eine Information mehr.

In Konzept 2 ist ebenfalls wie im entfernungsbasierten Konzept die zusätzliche Einführung von Kategorien denkbar.

Hinsichtlich der Clusteransicht bietet Konzept 2 gegenüber Wikitude mehrere Vorteile: Die Anzeige der aktuellen Seitenzahl zusammen mit der Gesamtheit aller Seiten wurde eingeführt. Außerdem sind in Konzept 2 alle im Kreis angeordneten POIs zugleich abgebildet und obendrein nach den Entfernungen sortiert. Zusätzlich enthält Konzept 2 im Gegensatz zu Wikitude einen „Zurück“ Button, mit dem der Benutzer auch die POIs der vorherigen Seiten betrachten kann. Die ersten drei der hier aufgeführten Optimierungen in der Clusteransicht dienen einer besseren Übersicht und helfen dadurch dem Benutzer, sich in dieser Ansicht besser zurechtzufinden. Das Einführen eines „Zurück“ Buttons

5. Vorstellung der POI Clusterbehandlung nach Umsetzung der beiden Konzepte

erleichtert es dem Nutzer zu den Ansichten der vorherigen Seiten zu wechseln, ohne dass dieser wie in Wikitude dazu ständig den Button zum Wechseln auf die nächste Ansicht betätigen muss.

Wie ersichtlich, haben beide erstellten Konzepte völlig andere Ansätze. Während in dem ersten viel Wert auf Distanzen gelegt wird, liegt der Fokus im zweiten Konzept auf dem Zusammenfassen der POIs und die Darstellung derer innerhalb eines Clusters.

6

Fazit und Ausblick

Zum Abschluss dieser Arbeit erfolgt eine Zusammenfassung der Ergebnisse der einzelnen Kapitel. Die Arbeit wird mit einem Ausblick abgeschlossen.

6.1. Fazit

Ziel dieser Arbeit war es, zuerst zwei Konzepte zu entwerfen, die zu einer Verbesserung von POI Clusterbildungen führen, indem auch mit den POIs interagiert werden kann, die durch andere teilweise oder vollständig verdeckt werden. Ein weiteres Ziel war es zudem, diese Konzepte prototypisch zu implementieren.

Zu diesem Zweck wurden auf dem Markt frei erhältliche Applikationen untersucht und betrachtet, wie diese mit Clustern aus POIs umgehen. Wie sich herausgestellt hat, bieten nicht alle Applikation eine Lösung zur POI Clusterbehandlung. Bei denjenigen, die

6. Fazit und Ausblick

ein Verfahren zur Verbesserung dieses Problems aufweisen, ergaben sich durch eine Bewertung Schwächen dieser Verfahren. Es wurden daraufhin anhand dieser Analyse zwei eigene Konzepte erstellt. Dabei wurde auf die Schwächen der untersuchten Applikation geachtet, um diese in den eigenen Konzepten zu vermeiden oder zu verbessern. Zusätzlich fand eine Optimierung durch die Einführung neuer Eigenschaften statt.

Im nächsten Schritt wurden die eigens erstellen Konzepte implementiert und das Vorgehen dabei in Kapitel 4 erläutert. Eine Hürde stellte dabei das Einarbeiten in den vorhandenen Programmcode von AREA dar, damit Änderungen an passenden Stellen vorgenommen werden konnten. Jedoch wurde dies durch die verständliche und umfassende Kommentierungen im Quelltext vereinfacht.

Die eigenen Konzepte wurden daraufhin in Kapitel 5 vorgestellt und mittels Screenshots veranschaulicht. Eine darauf folgende Bewertung der eigenen entworfenen und prototypisch implementierten Konzepte hat die wesentlichen Optimierungen gegenüber anderen Applikationen aufgezeigt. Ebenso wurden in dieser Bewertung auch die neuen Eigenschaften mit berücksichtigt.

6.2. Ausblick

Die prototypischen Implementierung beider neu erstellten Konzepte lassen sich an manchen Stellen weiterhin optimieren und erweitern.

Ein Punkt stellt dabei die richtige Neuberechnung und Positionierung der POIs im ersten Konzept bei Drehung des mobilen Endgeräts dar.

Im zweiten implementierten Konzept gibt es mehrere Punkte, die verbessert werden könnten.

Als erstes ist eine genaue Berechnung von Clustern anhand dichtebasierter Clusteralgorithmen denkbar. Darauf aufbauend könnten die Cluster-Punkte auf dem Display aktualisiert und die darin enthaltenen POIs aus der Kameraansicht entfernt werden, damit diese nur noch in der Cluster Kreisansicht angezeigt werden. Auch eine andere Clusteransicht, die optisch der im erstellten Konzept gleicht (siehe Abschnitt 3.2.2), wäre

denkbar, da die prototypische Implementierung dieses Konzepts optisch noch nicht ausgereift ist, sondern lediglich vor Augen führt, dass eine Clusteransicht mit den darin enthaltenen POIs, Buttons und der Seitenzahl realisierbar ist.

In Abschnitt 5.3 wurde erwähnt, dass die Einführung von Kategorien in beiden Konzepten denkbar sei, da sich dies als Vorteil in den untersuchten Applikationen erwies. Es existieren bereits Augmented Reality Anwendungen, in denen AREA eingebettet ist, die das Kriterium der Kategorien bereits berücksichtigen [25]. Für AREA selbst könnte dies aber auch in Betracht gezogen werden, damit für den Nutzer sofort erkenntlich ist, um was für eine Art Interessenspunkt es sich handelt.

Abschließend fehlt bisher eine Bewertung darüber, welches der eigenen Konzepte sich besser in der Praxis bewährt. Beide Konzepte haben unterschiedliche Herangehensweisen, Clusterbildungen zu verbessern. Zudem sind beide optimierbar und haben jeweils Vorzüge gegenüber dem anderen. Im ersten Konzept hat sich herausgestellt, dass die Einführung eines Schiebereglers zur Einstellung einer minimalen Distanz von Vorteil ist. Das zweite Konzept bietet dagegen eine gute Darstellung der Cluster sowie die Darstellung innerhalb dieser. Aufgrund dessen könnte eine Kombination beider Konzepte in Betracht gezogen werden.

Insgesamt lässt sich hieraus der Schluss ziehen, dass manche Stellen noch zu optimieren und erweiterbar sind. Dem Ziel, zwei Konzepte für Verfahren zur POI Clusterbehandlung zu entwerfen und diese daraufhin prototypisch zu implementieren, wurde in dieser Arbeit vollständig nachgegangen.

A

Verzeichnisse

Abbildungsverzeichnis

2.1. Screenshot AREA	6
2.2. Informationen zu einem angewählten POI (links) - Distanzeinstellung (rechts) (Screenshots AREA)	7
2.3. Cluster (Screenshot AREA)	8
3.1. Einzelne POIs (Screenshot Wikitude)	11
3.2. Ansicht eines Clusters (Screenshot Wikitude)	11
3.3. Darstellung der Ansicht der POIs innerhalb eines Clusters (Screenshot Wikitude)	12
3.4. „Monocle“ Ansicht (Screenshot yelp)	14
3.5. Deutliche Überschneidung bei mehreren POIs (Screenshot yelp)	15
3.6. Darstellung der POIs (Screenshot junaio)	17
3.7. Darstellung der POIs (Screenshot mixare)	19
3.8. Bodenansicht (Screenshot mixare)	19
3.9. Interessenspunkt - Entfernung: 150 Meter (Screenshot mixare)	20
3.10. Nachgestellte Darstellung der POIs in AREA (erstes Konzept)	25
3.11. Darstellung der POIs (erstes Konzept)	26
3.12. Darstellung eines Clusters (zweites Konzept)	27
3.13. Darstellung von acht POIs innerhalb eines Clusters (zweites Konzept)	28
3.14. Buttons mit Pfeilen und Darstellung der Seitenzahl (zweites Konzept)	29
4.1. if-Anweisung zur Eingrenzung der angezeigten POIs im Radar	32
4.2. if-Anweisung zur Eingrenzung der angezeigten POIs im Display	33

Abbildungsverzeichnis

4.3. Ausschnitt der Methode zur Berechnung von Größe und Farbe der Kreise	33
4.4. Überprüfung auf verdeckte POIs	34
4.5. Neues Positionieren eines POI	34
4.6. Teil der Berechnung der POIs auf aktueller Seite	37
4.7. Zuständige Klasse für die Anordnung der Objekte	38
5.1. Screenshots AREA (Konzept 1, Vorher - Nachher)	42
5.2. Screenshots AREA (Konzept 1, unterschiedliche Distanzen)	43
5.3. Einstellung der Distanzen (Konzept 1)	44
5.4. Screenshot AREA (Konzept 2)	45
5.5. Screenshots AREA - Cluster Kreisansicht (Konzept 2)	46

Tabellenverzeichnis

3.1. Bewertungen ausgewählter, bereits existierender Engines	21
--	----

Literaturverzeichnis

- [1] AZUMA, R.; BAILLOT, Y.; BEHRINGER R.; FEINER S. K.; JULIER S.; MACINTYRE B.: *Recent Advances in Augmented Reality*. In: IEEE Computer Graphics and Applications. 21, Nr. 6, 2001, S. 34.
- [2] *IKEA Katalog App*. http://www.ikea.com/ms/de_DE/campaigns/services/kataloge_und_apps.html?icid=de|itl|fy14_service|teaser|kw3|83#IKEA_App. abgerufen am 07.04.2014.
- [3] *Audi eKurzinfo*. http://www.audi.de/de/brand/de/erlebniswelt/audi_multimedial/audi_apps/audi-connect_und_mobilitaet/audi_ekurzinfo.html. abgerufen am 07.04.2014.
- [4] SCHOBEL, J.; SCHICKLER, M.; PRYSS R.; NIENHAUS H.; REICHERT M.: *Using Vital Sensors in Mobile Healthcare Business Applications: Challenges, Examples, Lessons Learned*. 9th Int'l Conference on Web Information Systems and Technologies (WEBIST 2013), Special Session on Business Apps, 509–518, Mai 2013.
- [5] *Apple iOS*. <https://www.apple.com/de/ios/>. abgerufen am 07.04.2014.
- [6] *Android*. <http://www.android.com/>. abgerufen am 07.04.2014.
- [7] GEIGER, P.: *Entwicklung einer Augmented Reality Engine am Beispiel des iOS*, 2012. Universität Ulm, Bachelorarbeit.
- [8] *Apple*. <http://www.apple.com/>. abgerufen am 09.04.2014.
- [9] *Apple iPhone*. <http://www.apple.com/de/iphone/>. abgerufen am 09.04.2014.

Literaturverzeichnis

- [10] GEIGER, P.; SCHICKLER, M.; PRYSS R.; SCHOBEL J.; REICHERT M.: *Location-based Mobile Augmented Reality Applications: Challenges, Examples, Lessons Learned. 10th Int'l Conference on Web Information Systems and Technologies (WEBIST 2014), Special Session on Business Apps*, 383–394, April 2014.
- [11] *Wikitude*. www.wikitude.com. abgerufen am 30.01.2014.
- [12] *yelp*. www.yelp.de. abgerufen am 30.01.2014.
- [13] *junaio*. <http://www.junaio.com>. abgerufen am 09.02.2014.
- [14] *mixare*. www.mixare.org. abgerufen am 30.01.2014.
- [15] *Artikel zu Wikitude auf Wikipedia*. <http://de.wikipedia.org/wiki/Wikitude>. abgerufen am 08.02.2014.
- [16] *Wikipedia*. <http://www.wikipedia.de/>. abgerufen am 06.02.2014.
- [17] *flickr*. <http://www.flickr.com/>. abgerufen am 08.02.2014.
- [18] *Artikel über yelp*. <http://www.finanzen.net/aktien/Yelp-Aktie>. abgerufen am 20.02.2014.
- [19] *Metaio Developer Portal*. <http://dev.metaio.com/junaio/>. abgerufen am 09.02.2014.
- [20] *Displaying location based content in junaio*. <http://dev.metaio.com/junaio/documentation/channels/location-based-channels/#c2486>. abgerufen am 09.02.2014.
- [21] *Oracle Java*. <http://www.oracle.com/de/technologies/java/overview/index.html>. abgerufen am 09.04.2014.
- [22] *Samsung Galaxy S4*. <http://www.samsung.com/de/consumer/mobile-device/mobilephones/smartphones/GT-I9505ZKADBT#>. abgerufen am 09.04.2014.
- [23] *Android API*. <http://developer.android.com/reference/packages.html>. abgerufen am 01.04.2014.

- [24] AGGARWAL, C. C.; REDDY, C. K.: *Data Clustering: Algorithms and Applications*. Chapman and Hall/CRC, 2013, S. 111-126.
- [25] *AREA - Augmented Reality Engine Application*. <http://www.area-project.info/>. abgerufen am 08.04.2014.

Name: Julia Müller

Matrikelnummer: 696645

Erklärung

Ich erkläre, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

Julia Müller