Analyse von Bewegungsdaten zur Identifizierung von potentiellen Freundschaftsbeziehungen

Bachelorarbeit

Jan Burmeister
Nelly-Sachs Str. 1 – Sankt Augustin
jan.burmeister@uni-bonn.de

Bonn, den 13. 09. 2012

Rheinische Friedrich-Wilhelms-Universität Bonn
Institut für Informatik III
Professor Dr. Armin B. Cremers

universitätbonn

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind.

Bonn, den 13. September 2012

Überblick

Soziale Netzwerke und mobile Geräte wie Smartphones oder Tablets sind aus dem Alltag vieler Menschen nicht mehr wegzudenken. Ziel solcher moderner Technologien ist es möglichst nahtlos mit dem natürlichen Leben des Nutzers zu verschmelzen. Die Fähigkeit der Positionsbestimmung, über die der Großteil aller modernen Smartphones verfügt, ermöglicht es den sozialen Kontext eines Nutzers über Verfolgung der Positionsverläufe zu bestimmen. Diese Arbeit soll ein neues Verfahren zur Bestimmung von Freunden in sozialen Netzwerken vorstellen, das anhand des Vergleichs von Positionsverläufen arbeitet. Dabei soll mit ungenauen Positionen gearbeitet werden um die Privatsphäre der Benutzer zu berücksichtigen. Es wird ein Verfahren zur Darstellung unscharfer Positionen vorgestellt, und die Eignung für den gegebenen Anwendungskontext evaluiert.

Inhaltsverzeichnis

Abbildungen und Tabellen

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einführung

1.1 Einleitung

Smartphones haben in den letzten Jahren unser Leben verändert. Fähigkeiten wie mobiles Internet, Bewegungssensoren und Positionsbestimmung haben völlig neue Möglichkeiten eröffnet Anwendungen zu erstellen, die sich in das natürliche Leben des Benutzers integrieren. Anstelle von sperrigen Computern und Eingabegeräten sind Touch-Displays, Bewegungs- und Gestensteuerung getreten. Die Fähigkeit zur Positionsbestimmung via GPS oder WLAN ermöglicht es erstmals präzise ortsbezogene Dienste bereitzustellen. Benutzer müssen nicht mehr in einem lästigen Formular angeben wo sie sich befinden; intelligente Dienste können dies völlig transparent im Hintergrund erledigen, und dem Benutzer Ergebnisse zur Verfügung stellen, ohne dass ein manuelles Eingreifen des Benutzers notwendig wäre. Die Konfiguration der Programme tritt in den Hintergrund; der Benutzer erhält direkt die Informationen, die gerade für ihn relevant sind. Beispiele dafür sind Informationen über das lokale Wetter, Sehenswürdigkeiten, Gastronomie oder einfach die Bestimmung der aktuellen Position auf Straßenkarten.

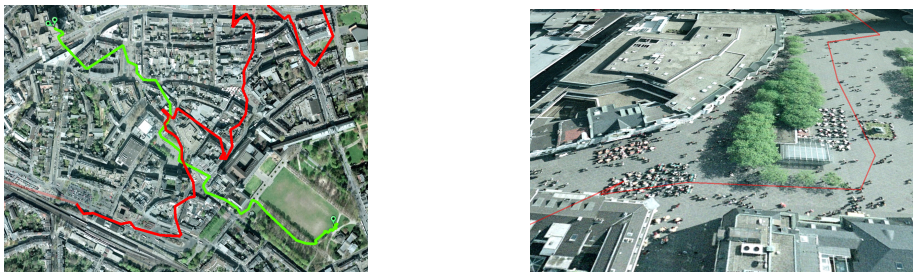
Eine Beobachtung des Positionsverlaufs über längere Zeit ermöglicht Auswertungen über die Interessen und den sozialen Kontext eines Benutzers. Ein interessantes Einsatzgebiet für diese Technik bieten soziale Netzwerke. Genau wie Smartphones haben soziale Netzwerke in kürzester Zeit eine beeindruckende Verbreitung gefunden. Dabei haben sich beide Entwicklungen wohl auch gegenseitig unterstützt: Knapp die Hälfte der Facebook Nutzer, über 400 Millionen Menschen, greifen über mobile Geräte auf das Netzwerk zu [Fac12]. Die App des Social Networks gehört unter iOS und Android zu den fünf meistverwendeten Anwendungen. Etwa 70% aller Android- und 80% aller iOS Nutzer verwenden ihr Smartphone um sich mit Facebook zu verbinden. Im Monat verbringen die Nutzer dabei durchschnittlich 441 Minuten mit dem mobilen Facebook [com12]. Genau wie mobile Geräte, haben

soziale Netzwerke offenbar längst einen Weg in unser alltägliches Leben gefunden. Letztendlich leben solche Netzwerke gerade von einer engen Einbindung in das reale Leben ihrer Nutzer. Die gegebenen Fähigkeiten von Smartphones können diese Einbindung noch nahtloser möglich machen. Die Grundlage dafür ist gegeben: Bereits jetzt verzeichnen soziale Netzwerke deutlich mehr Zugriffe von Apps auf mobilen Geräten, als über den klassischen Zugang mit einem Webbrowser [com12].

Der vielleicht bedeutendste Faktor für einen Nutzer ist die Anzahl realer Freunde, die bereits in einem Netzwerk angemeldet sind. Im Laufe der Zeit finden sich in der realen Welt zudem auch neue Bekanntschaften, wie z.B. neue Arbeitskollegen oder Nachbarn. Letztlich läuft es darauf hinaus, dass man in einem Netzwerk versucht die andere Person zu finden, was sich häufig als frustrierend herausstellt. Zwar verwenden die Netzwerke entsprechende Algorithmen um potentielle Freunde vorzuschlagen, diese scheinen jedoch größtenteils auf bereits bekannten Verbindungen zu anderen Personen zu basieren (die genauen Vorgehensweisen sind nicht bekannt). Neue Bekannte, zu denen man keine indirekte Verbindung über andere Freunde hat (z.B. nach einem Umzug in eine andere Stadt) können nicht vorgeschlagen werden. Eine Suche nach dem genauen Namen liefert bei den großen Netzwerken unüberschaubar lange Ergebnislisten. Auch Pseudonyme machen die Suche nach Bekannten nicht einfacher.

In dieser Arbeit soll eine alternative Art der Bestimmung potentieller Freunde in einem sozialen Netzwerk bearbeitet werden. Grundlage dafür ist die Möglichkeit der Positionsbestimmung, über die derzeit nahezu jedes der erhältlichen Smartphones verfügt. Soziale Netzwerke wie Facebook, Google Latitude oder Foursquare benutzen diese Daten bereits um Benutzern ortsbezogene Dienste anbieten zu können. Darüber hinausgehend kann man auf diesem Weg auch Bekannte in einem Netzwerk aufzeigen. Die Grundannahme ist simpel: Bekannte, die man als Freunde in einem Netzwerk hinzufügen möchte, sind Personen, mit denen man auch in der realen Welt Zeit verbringt. Oder anders: Über einen gewissen Zeitraum stimmen die Positionen dieser zwei Personen überein.

Die Speicherung eines Positionsverlaufs einer Person hat einige Vorteile: Ein Abgleich von Verläufen liefert nur tatsächliche Bekanntschaften, oder zumindest Personen, mit denen man anderweitig Zeit verbringt. Zudem kann eine Segmentierung nach dem Typ der Bekanntschaft berechnet werden, indem z.B. Ort und Zeit der Übereinstimmungen beachtet werden. Reine Arbeitskollegen trifft man z.B. nicht spät abends. Zudem ist dieses Verfahren unabhängig von Stützhilfen wie bekannten Verbindungen zwischen Personen. Letztendlich sind keinerlei Zusatzangaben vom Benutzer notwendig. Das Verfahren arbeitet völlig transparent im Hintergrund. Ein motivierendes Beispiel soll die Idee dieses Verfahrens verdeutlichen.



(a) Übereinstimmung **(b)** Genauigkeit

Abbildung 1.1: Darstellung der zwei Positionsverläufe mittels Google Earth. (a) zeigt die Übereinstimmung beider Verläufe, (b) die recht hohe Präzision der Positionsaufnahme. Der ursprüngliche Weg während der Aufnahme ist gut erkennbar.

1.2 Motivierendes Beispiel

Für dieses Beispiel wurden mit einem handelsüblichen Smartphone zwei Positionsverläufe in Bonn aufgezeichnet. Abbildung 1.1 zeigt eine Visualisierung beider Verläufe. Direkt erkennbar ist die Überkreuzung beider Verläufe. Betrachtet man zudem die Zeitpunkte dieser Überkreuzung findet man eine Übereinstimmung der Positionen für etwa 20 Minuten - ein Hinweis auf eine mögliche Bekanntschaft. Um weitere Informationen über diese mögliche Bekanntschaft zu bekommen, können wir neben der Zeitdauer auch den Ort des Treffens näher untersuchen.

Die Google Places API [Goo12a] bietet dazu beispielsweise die Möglichkeit zu gegebenen Positionsdaten naheliegende interessante Punkte wie Geschäfte, Gastronomie, Sehenswürdigkeiten und sonstige bei Google eingetragene Punkte zu suchen. In diesem Fall war es möglich den Punkt des Treffens als ein Café zu identifizieren.

Selbst wenn solche Daten nicht zur Verfügung stehen, können wir Orte anhand ihrer Frequentierung als Orte mit hoher, oder niedriger Aussagekraft für eine Übereinstimmung zweier Nutzerpositionen bewerten. An Orten, an denen sich viele Nutzer aufhalten, ist die Wahrscheinlichkeit höher, dass das vermutete Treffen zwischen zwei Benutzern eventuell gar keines ist. In unserem Café sind zwar beide Nutzer gleichzeitig anwesend gewesen, müssen aber nicht zwangsläufig auch am selben Tisch gesessen haben. Aufgrund solcher Faktoren muss dieses Treffen mit einer geringen Relevanz bewertet werden. Hier sind weitere Informationen über z.B. frühere Treffen mit zu betrachten. Hätten sich unsere Nutzer an einem Ort mit geringer Frequentierung (und damit höherer Aussagekraft) getroffen, wie beispielsweise dem Wohnort eines Nutzers, ist die Schlussfolgerung, dass hier eine Bekanntschaft vorliegt, als sehr viel sicherer einzuordnen.

In beiden Fällen könnte ein soziales Netzwerk nun die beiden Nutzer, mit entsprechend angepasster Relevanz, als Freunde vorschlagen. Eventuell wussten sie ja nicht einmal, dass der jeweils andere im selben Netzwerk aktiv ist. Das Netzwerk orientiert sich hier am realen Leben der Nutzer, und bietet eine nahtlose Integration an. Es sind nicht die Nutzer, die ihre Einstellungen im Netzwerk an ihr Leben anpassen müssen; stattdessen passt sich das Netzwerk den Benutzern an.

1.3 Datenschutzprobleme

Neben den positiven Aspekten dieses Verfahrens fällt jedoch auch ein großes Problem ins Auge: Die Positionsverläufe in Abbildung 1.1 zeigen metergenau die Bewegungen der Nutzer. Um obiges Beispiel realisieren zu können, müssen diese Daten an den Betreiber des Social Networks übertragen, und dort ausgewertet werden. Aus diesen Daten können jedoch noch viel mehr Informationen gewonnen werden.

Betrachten wir den rot eingezeichneten Positionsverlauf: Während der Aufzeichnung wurde protokolliert, an welchen Orten sich der Nutzer über längere Zeit aufgehalten hat. Anhand des Positionsverlaufs können diese Orte rekonstruiert werden, indem nach Clustern gesucht wird, also Orten an denen die aufgezeichneten Punkte über längere Zeit innerhalb eines gewissen Radius liegen. Mithilfe der Google Places API können diesen Punkten semantische Informationen zugeordnet werden. Tabelle 1.1 zeigt, wie genau mit diesem einfachen Verfahren die Bewegung des Nutzers nachvollzogen werden konnte, und auch semantische Informationen über den Nutzer gesammelt werden konnten.

Zeit	Ort	Kategorie
09:30:25 - 09:35:12	Argelander-Institut für Astronomie	university, establishment
09:52:10 - 10:12:56	Starbucks	cafe, food, establishment
10:15:11 - 10:20:53	Bouvier	store, establishment
10:33:27 - 10:38:20	Fachschaft Informatik Universität Bonn	university, establishment

Tabelle 1.1: Extrahierte Cluster aus dem Positionsverlauf. Die Einträge wurden unverändert von Google Places übernommen.

Das Beispiel verdeutlicht, welches Potential ein solches Verfahren beinhaltet, jedoch auch mit welchen Nachteilen es verbunden ist.

In dieser Form wird ein solches Verfahren nur schwer zu verwirklichen sein. Die Speicherung metergenauer Positionsdaten wird von vielen Benutzern vermutlich als unangenehm empfunden werden. Obiges Beispiel zeigt nur eine Möglichkeit aus diesen Daten sehr präzise Nutzerprofile anlegen zu können. Datenschützer versuchen bereits seit Jahren die Internetnutzer zu den Themen Privatsphäre und Datenschutz zu sensibilisieren. Tatsächlich scheint die erhöhte Medienaufmerksam-

keit Wirkung zu zeigen, sodass immer mehr Nutzer auf ihre Privatsphäre insbesondere in sozialen Netzwerken achten [DJR12].

Eine Weitergabe derart präziser Positionsdaten kann also keine zufriedenstellende Lösung sein. Entsprechend muss eine Möglichkeit gegeben werden, um die Präzision der Positionsbestimmung individuell festzulegen.

1.4 Problemstellung

Aufgabe dieser Bachelorarbeit soll es sein, ein geeignetes Verfahren zur Auswertung und Speicherung von Positionsverläufen zu erarbeiten. Ziel dieses Verfahrens ist die Identifikation und Einordnung möglicher Bekanntschaften in der realen Welt.

Für die Speicherung muss eine Darstellung der Positionsdaten und -verläufe gefunden werden, die zum einen möglichst platzsparend ist, aber auch die Möglichkeit erlaubt die Präzision der Position einfach reduzieren zu können. Das heißt, die zu übertragenden Positionsdaten sollen bestenfalls gar nicht, oder nur ungenau auf die tatsächliche exakte Position des Nutzers schließen lassen. Die traditionellen Längen- und Breitengrade erlauben diese Reduzierung der Präzision nur ungenügend.

Zudem muss sich die Darstellung für die notwendigen Berechnungen eignen, allen voran der Distanzbestimmung zwischen zwei Positionen. Diese Berechnungen sollten mit möglichst wenig Aufwand durchgeführt werden können. Gesucht ist eine Form der Positionsdarstellung, die folgende Anforderungen erfüllt:

- Matchings (Übereinstimmungen zwischen zwei Nutzern) und Ortsbewertungen sind berechenbar, d.h. es müssen Entfernungen zwischen einzelnen Positionen berechenbar sein. Die berechneten Entfernungen müssen relativ mit Entfernungen in der echten Welt übereinstimmen (Distanzerhaltung).
- Positionen der Darstellung sollen möglichst nicht, oder zumindest nur ungenau, in exakte Positionen der echten Welt zurückrechenbar sein.
- Berechnungen (Matching und Ortsbewertungen) sollen nicht allzu gravierend von der Berechnung mit exakten Positionen (Längen-/Breitengrad) abweichen. Inwieweit dies möglich ist, ist insbesondere Untersuchungsgegenstand dieser Arbeit.
- Dynamische Anpassung der Genauigkeit der Positionen ist gewünscht. Benutzer einer entsprechenden Anwendung sollen selbst über die Genauigkeit ihrer Angaben bestimmen können.

Die Auswertung umfasst sowohl die Erarbeitung der Darstellung der Positionsdaten als auch deren Evaluation auf den gegebenen Anforderungen. Dabei soll ein besonderer Fokus darauf gelegt werden, dass der Benutzer in der Lage ist die Präzision seiner Position herabzustufen.

Eine wichtige Untersuchung soll sein herauszufinden bis zu welcher Genauigkeit aus der gewählten Positionsdarstellung noch annehmbare Resultate für Matchings und Ortsbewertungen berechnet werden können.

Diese Fragestellung ist durchaus nicht nur aus Datenschutzgründen interessant. Entwickler von Smartphone Anwendungen müssen oft hinnehmen, dass die Benutzer die präzise, aber stromhungrige GPS-Ortung abschalten, um ihren Akku zu schonen. Um dennoch Funktionalität bieten zu können, bleibt nichts anderes übrig als auf weniger genaue Methoden, wie Ortung per WLAN, umzuschalten.

1.5 Aufbau dieser Arbeit

Kapitel 2 führt zuerst einige grundlegende Definitionen ein, die im weiteren Verlauf verwendet werden.

Kapitel 3 widmet sich verwandten Arbeiten in diesem Bereich, sowohl im akademischen Bereich, als auch an Anwendungen, die derzeit in Verwendung sind.

Kapitel 4 geht auf verschiedene Möglichkeiten der Darstellung ungenauer Positionen ein, und gibt eine abschließende Bewertung.

Kapitel 5 stellt eine dieser Möglichkeiten, Quad Keys, im Detail vor, und zeigt die Vorteile in diesem Anwendungskontext auf.

Kapitel 6 untersucht diese Darstellungsform im Hinblick auf die eingangs definierten Anforderungen bei unterschiedlichen Präzisionen.

Kapitel 7 bietet einen zusammenfassenden Abschluss, und stellt mögliche zukünftige Arbeiten in Aussicht.

Kapitel 2

Definitionen und methodische Grundlagen

2.1 Positionen und Sequenzen

Mit Positionsdaten werden Tupel der Form (Positionsangabe, Zeitpunkt) bezeichnet. Positionsdaten bestimmen damit Orte auf der Weltkugel zu einem bestimmten Zeitpunkt. Die Art der Positionsangabe wird hier nicht weiter festgelegt.

Der Positionsverlauf eines Nutzers ist eine geordnete Menge von Positionsdaten, wobei die einzelnen Daten anhand ihres Zeitpunkts aufsteigend geordnet sind. Dies sind die Ausgangsdaten, die z.B. durch den GPS-Sensor eines Smartphones aufgenommen werden.

Als *Matching* werden zwei Positionsdaten bezeichnet, die innerhalb eines gegebenem Orts- und Zeitradius zueinander liegen.

Als *Sequenz* wird eine geordnete Menge von Matchings bezeichnet, die anhand ihrer Zeitpunkte aufsteigend geordnet sind. Für alle Matchings einer Sequenz gilt, dass sie sich innerhalb eines gegebenen Zeitradius zum jeweils vorherigen Matching befinden. Eine Sequenz enthält immer nur Matchings von genau zwei Nutzern. Sequenzen repräsentieren also einen gemeinsamen Bewegungsverlauf zweier Nutzer. Die Länge einer Sequenz wird durch die Differenz des ersten und letzten Zeitpunkts der Sequenz definiert.

2.2 Ortsbewertungen

Die hier verwendeten Metriken zur Bewertung von Orten, sind aus [CTH+10] entnommen.

Sei L ein Ort, und $O_{u,L}$ die Menge aller Positionsdaten von Benutzer u am Ort L. Weiter sei O_L die Menge der Positionsdaten aller Benutzer am Ort L, und U_L die Menge aller Benutzer am Ort L

Als *Frequency* Freq$(L) =\mid O_L \mid$ wird die totale Anzahl Besuche an einem Ort L bezeichnet. UserCount$(L) =\mid U_L \mid$ bezeichnet die Anzahl Benutzer, die einen Ort L besucht haben.

Die Wahrscheinlichkeit, dass ein zufälliges Positionsdatum p am Ort L zu Benutzer u gehört ist $P_L(u) = \frac{|O_{u,L}|}{|O_L|}$. Die *Location Entropy* sei nun definiert als

$$\text{Entropy}(L) = -\sum_{u \in U_L} P_L(u) \log P_L(u)$$

Ein Ort wird durch hohe Entropie ausgezeichnet, wenn sich dort viele verschiedene Nutzer zu gleichem Anteil aufhalten. Niedrige Entropie ergibt sich wenn der Ort hauptsächlich nur von wenigen Nutzern aufgesucht wurde.

Cranshaw et al. definieren noch weitere Bewertungskriterien, die jedoch zumeist auf anderen hier vorgestellten Berechnungen aufbauen. Um im späteren Teil der Arbeit eine Evaluation durchzuführen, beschränken wir uns auf diese grundlegenden Metriken.

Kapitel 3

Verwandte Arbeiten

Mehrere kommerzielle Betreiber wie Facebook, Google oder Foursquare ermöglichen Nutzern die Weitergabe ihrer aktuellen Position an andere Nutzer des Netzwerks. Ziel ist es das Netzwerk noch weiter in das reale Leben der Nutzer zu integrieren. Inwieweit diese Daten zur Berechnung von potentiellen Freunden verwendet werden ist nicht bekannt.

Im akademischen Bereich ist eine Arbeitsgruppe von Microsoft Research Asia sehr weit fortgeschritten in der Auswertung von Positionsverläufen. Zheng, et al. [ZZM+11] legen hier den Fokus jedoch nicht auf den Abgleich der Verläufe um zeitliche Übereinstimmungen zu finden, sondern darauf Interessen anhand bestimmter Orte, die der Nutzer besucht hat, zu finden. Anhand berechneter Ortsinteressen sollen dann Freunde und Orte vorgeschlagen werden, die den Interessen des Nutzers nahe kommen könnten.

Dafür werden in den Positionsverläufen der Nutzer interessante Orte identifiziert, an denen sich der Benutzer länger aufgehalten hat (sog. „Stay Points“). Auf Basis der Stay Points aller Nutzer wird ein hierarchisches Modell erstellt, in dem die Stay Points Clustern auf unterschiedlichen Genauigkeitsstufen zugeordnet werden. Dieses Modell erlaubt es den Nutzern ihre Positionsverläufe auf einer einheitlichen Basis mit den einheitlichen Clustern als Sequenz darzustellen. Ein Ranking-Algorithmus sucht nun nach Übereinstimmungen in den Sequenzen der Nutzer. Dabei wird die Länge der Übereinstimmung, die Genauigkeitsstufen auf denen Übereinstimmungen auftreten, und die Frequentierung des Ortes der Übereinstimmung mit berücksichtigt.

Mit diesem Ansatz werden die Positionsverläufe sowohl nach gemeinsamen Interessen der Nutzer durchsucht, als auch nach tatsächlichen zeitlichen Übereinstimmungen. Das System schlägt einem Nutzer entsprechend Personen vor, die gleiche Orte besucht haben, und vielleicht auch zeitliche Übereinstimmungen gezeigt haben. Zusätzlich werden aus den Sequenzen von Nutzern mit ähnlichen Interessen weitere Stay

Points identifiziert, sodass zusätzlich zu Personen auch interessante Orte vorgeschlagen werden. Somit werden auch gleichzeitig semantische Verbindungen zwischen Orten erarbeitet.

Li und Chen [LC09] haben mehrere Metriken für die Bestimmung von Freunden untersucht, darunter auch Ähnlichkeit von Positionsdaten. Die zugrunde liegenden Daten bestehen jedoch aus einzelnen Datenpunkten, die der Benutzer durch „Check-ins“ manuell generiert, und nicht aus ganzen Positionsverläufen. Die Autoren haben dafür ein 3-Schichten Modell für Freundschaften in sozialen Netzwerken entwickelt, das die Verbindungen zwischen den Benutzern („Social Graph“), gemeinsame Interessen (über Profilinformationen) und Positionsdaten verwendet. Es konnte gezeigt werden, dass die Ortsinformationen einen signifikanten Anteil an der Leistung des Algorithmus beisteuerten.

Crandall et al. [CBC+10] verfolgen das gleiche Ziel auf Basis von Fotos mit Geo-Tag Informationen. Sie zeigen dabei, dass sich bereits aus wenigen Übereinstimmungen bezüglich Zeit und Ort soziale Strukturen mit hoher Wahrscheinlichkeit erkennen lassen. Als Datengrundlage wurden Bilder des populären sozialen Netzwerks Flickr benutzt, das in erster Linie einen Fokus auf die Verbreitung von Fotos legt. Die Autoren geben zu bedenken, dass sich die Nutzung einer solchen Plattform von anderen sozialen Netzwerken unterscheidet. Dennoch konnten sie zeigen, dass auch mit solchen Daten Beziehungen zwischen Personen ermittelt werden konnten.

Cho, Myers und Leskovec [CML11] untersuchen den Einfluss von Freundschaften auf das Bewegungsmuster von Personen, ebenfalls anhand von „Check-in“ Punkten. Zudem beobachten sie typische periodische Bewegungen zwischen Arbeitsplatz und Wohnort, und stellen ein Modell vor, um solche Bewegungsabläufe zu beschreiben und vorherzusagen.

Sie können zeigen, dass das Bewegungsverhalten von Menschen zu einem großen Teil durch periodisches Verhalten erklärbar ist, jedoch nur zu einem geringen Teil durch die soziale Struktur. Zudem beobachten sie, dass eine längere Übereinstimmung von Positionen ein starkes Zeichen einer sozialen Verbindung ist, ein sehr großer Teil der beobachteten Nutzer solche Übereinstimmungen mit ihren Freunden jedoch gar nicht zeigen. Das aus den erarbeiteten Erkenntnissen entwickelte Modell erlaubt, anhand solcher sozialen und periodischen Parameter gesteuert, die Erstellung relativ präziser Vorhersagen für Bewegungsverhalten von Personen.

Cranshaw et al. [CTH+10] leiten aus Positionsverläufen eine Reihe von Merkmalen ab, und untersuchen inwieweit diese Merkmale über verschiedene Ansätze des maschinellen Lernens für die Bestimmung von Freundschaftsbeziehungen verwendet werden können. Als Merkmale

werden hier nicht nur Übereinstimmung zwischen Positionen, sondern insbesondere auch der soziale Kontext eines Ortes beachtet.

Dabei wird insbesondere die Location Entropy (siehe Kapitel 2) als geeignetes Mittel zur Bewertung herausgestellt. Weitere Metriken, wie die Location Specificity, beziehen insbesondere das Verhalten der individuellen Benutzer in die Bewertung mit ein. Die Ergebnisse, die mit diesen Metriken erstellt wurden, konnten im Vergleich mit einfacheren, allgemeinen Metriken deutlich besser abschneiden. Dennoch konnten auch Cranshaw et al. feststellen, dass überzeugende Bewertungen anhand der Positionsdaten nicht unbedingt eine Freundschaft in einem sozialen Netzwerk bedeuten muss und umgekehrt. Als weiteres interessantes Ergebnis konnte gezeigt werden, dass aus den Orten, die ein Nutzer besucht auch auf die Anzahl seiner Freunde in einem sozialen Netzwerk geschlossen werden kann. Damit können diese Daten auch annäherungsweise einen Hinweis auf das soziale Verhalten der Nutzer liefern.

Backstrom, Sun und Marlow [BSM10] untersuchen den inversen Weg, und versuchen anhand von Benutzerdaten eines sozialen Netzwerks die Position von anderen Benutzern zu erkennen. Zuvor stellen die Autoren dar, inwieweit der Zusammenhang zwischen Freundschaften und Entfernung auch in sozialen Netzwerken noch gültig ist.

Die Autoren zeigen, dass allein durch die Verbindung zu Freunden in einem sozialen Netzwerk Positionsdaten zumindest approximativ bestimmt werden können. Dabei wird in erster Linie auf die bekannten Positionsdaten von Freunden zurückgegriffen. Eine genauere Gewichtung dieser Daten kann durch Einbeziehung weiterer Faktoren, wie der Stärke der Beziehung in dem Netzwerk, erfolgen. Dies kann z.B. durch die Beobachtung der Kommunikation, oder der Anzahl Profilaufrufe von Freunden gemessen werden.

Benkert et al. [BDGW07] stellen einen Sweep-Algorithmus vor, um aus Positionsverläufen populäre Orte zu identifizieren. Die Problemstellung hat weitreichende Anwendung in der Verhaltensanalyse von Populationen, aber auch im Kontext sozialer Netzwerke, um die Relevanz von Orten zu bewerten.

Amir et al. [AEM+07] verwenden Positionsdaten im Anwendungskontext sozialer Netzwerke. Sie stellen ein Verfahren vor, das Benutzer benachrichtig sobald ein Freund sich in der Umgebung befindet. Vorgestellt wird ein klassisches zentral gesteuertes System, primär jedoch ein dezentrales System, das die Berechnung nach dem Peer-to-Peer Prinzip auf die Nutzergeräte aufteilt.

Dabei legen die Autoren insbesondere Wert auf die Effizienz der Algorithmen. Sie können zeigen, dass der von ihnen entwickelte Algorithmus auf dezentraler Basis eine effizientere Lösung darstellt, als ein zentral arbeitender Algorithmus, der jederzeit die Positionen aller Benut-

zer verwaltet. Dabei gehen die Autoren insbesondere auf das Problem ein, dass in großen Netzwerken die Anzahl eingehender Positionsdaten die Grenzen der Berechenbarkeit sprengen könnten.

Amir et al. verwendeten in ihrem Ansatz mit zentraler Berechnung Quadtrees, deren Prinzip auch in dieser Arbeit eine wichtige Rolle spielt. Grundlegende Arbeiten dazu wurden von Finkel und Bentley [FB74] geleistet, effiziente Algorithmen zur Suche nach nächsten Nachbarn, die insbesondere relevant sind, wurden etwa von Bhattacharya [Bha01] und Frisken und Perry [FP02] entwickelt.

Quadtrees als Grundlage für ortsbasierte Anwendungen zur Auswertung von Benutzerpunkten werden in einem breiten Feld verwendet. Narasimhan et al. [NMBM] zeigen den Einsatz für Navigationsverfahren (allerdings mit der Erweiterung auf drei Dimensionen), Steward und van der Ree [SR10] verwenden sie als Basis für die Erstellung von Voronoi-Diagrammen zur Populationsanalyse von Wildtieren.

Zusammenfassend zeigt sich eine eindeutige Schlussfolgerung: All jene Arbeiten, die sich mit der Analyse von Nutzerdaten beschäftigen, verwenden Informationen, die eigentlich nur einen unvollständigen Ausschnitt aus den Aktivitäten der Benutzer darstellen. Trotz dieser Einschränkung, zeigen alle, dass es dennoch möglich ist zu einem gewissen Grad soziale Strukturen und Verhaltensformen zu identifizieren.

Kapitel 4

Methoden zur Darstellung von unscharfen Positionen

4.1 Allgemeine Überlegungen

Positionen auf der Weltkugel werden in aller Regel mit Längen- und Breitengraden angegeben. Exakte Entfernungsbestimmungen sind daher aufwendig, da auf die Eigenschaften der Sphäre Rücksicht genommen werden muss. Die Haversine Formel [Sin84] stellt eine oft genutzte Lösung für dieses Problem dar, ist jedoch relativ komplex zu berechnen. Approximationen über eine (angenommene) Projektion der Weltkugel auf eine 2D-Ebene sind nur bedingt exakt. Der einfache Euklidische Abstand zwischen zwei Punkten ist nur bei kleinen Distanzen hinreichend korrekt; Projektionen auf eine Ebene bringen zwangsläufig Verzerrungen mit sich.

Nach der in Abschnitt 1.4 definierten Anforderungen, ist eine Darstellung von Koordinaten gesucht, die nicht, oder nur schwer auf Positionen in der realen Welt schließen lassen. Zudem sollen Entfernungsbestimmungen (effizient) möglich sein, und idealerweise nicht allzu stark von den tatsächlichen Werten abweichen.

Um in einem vertrauten Euklidischen Raum arbeiten zu können, betrachten wir Positionen auf der Erdkugel nicht in Längen- und Breitengraden, sondern als Punkte im $\mathbb{R}^3$, wobei die Positionen auf Punkte einer Sphäre im 3D-Raum abgebildet werden. Sei R der Radius, und $m \in \mathbb{R}^3$ der Mittelpunkt der Sphäre, dann ist die Menge $S = (v \in \mathbb{R}^3 \mid \|v - m\|_2 = R)$ die Menge über alle Punkte der Sphäre. Zwischen den Elementen von S sei eine Distanzfunktion definiert, die die Entfernung über die Oberfläche der Sphäre bestimmt. Dieses Modell entspricht der Darstellung der Erde im $\mathbb{R}^3$ (Oberflächenerhebungen werden vernachlässigt).

Gesucht ist eine Abbildung $f : S \to V$, die die Elemente aus S auf eine Menge V abbildet, deren Elemente keine Rückschlüsse auf die ursprüngliche, tatsächliche Position zulassen. Eine solche Abbildung muss jedoch, bedingt durch die Aufgabenstellung, die Bedingung erfüllen, dass die Distanzen zwischen allen Punkten erhalten bleiben, d.h. die Distanz aller Punkte zueinander muss im Quell- und Zielraum im gleichen Verhältnis liegen. Durch diese Forderung wird die Zielmenge V bereits massiv eingeschränkt.

Ein Punkt im $\mathbb{R}^2$, der zu drei anderen Punkten ein festes Distanzverhältnis einhalten muss, hat gar keine andere Wahl als im Zielraum die gleiche relative Position einzunehmen wie im Ursprungsraum (Bed: Die vier Punkte sind paarweise verschieden). Letztendlich findet nur eine Verschiebung, Rotation und/oder Skalierung aller Punkte der Quellmenge statt. Entsprechend ist es möglich drei bekannte Punkte abzubilden, und mit diesem Bezugssystem mittels einfacher Geometrie alle anderen abgebildeten Punkte zurückzurechnen. Dasselbe Problem tritt auch bei den Punkten auf der Sphäre im $\mathbb{R}^3$ auf. Die Forderung der Distanzerhaltung schließt damit die erhoffte Eigenschaft, dass abgebildete Positionen nicht zurückrechenbar sind, aus. Damit bleibt nur die Lösung, die Rückführung auf die tatsächlichen Positionen möglichst ungenau zu forcieren.

4.2 Unscharfe Positionen

Um eine ungenaue Positionsdarstellung zu realisieren, bieten sich zwei Möglichkeiten an:

- Modifizierung der Darstellung der Ursprungspunkte
- Modifizierung der Abbildung vom $\mathbb{R}^3$ in einen Zielraum (Nichtinjektive Abbildungen)

4.2.1 Modifizierung der Darstellung der Ursprungspunkte

In der bisherigen Darstellung realisieren wir die Positionen als Vektoren im $\mathbb{R}^3$, wobei die Vektoren wie üblich relativ zum Koordinatenursprung definiert sind. In dieser Variante werden die Positionen relativ zu sog. Ankerpunkten definiert. Die Positionen dieser Ankerpunkte im Raum sind fest definiert und bekannt. Die Positionen der Nutzer hingegen werden nur durch ihre Entfernung zu den Ankerpunkten definiert, und nicht durch bekannte Koordinaten im $\mathbb{R}^3$.

Solange die Entfernung zu mindestens vier Ankerpunkten bekannt ist, kann die Position eines Punktes exakt auf einen Punkt im $\mathbb{R}^3$ festgelegt werden. Eine unscharfe Positionsdarstellung kann nun erfolgen, indem diese Entfernungsangaben durch absichtliche Abweichungen modifiziert werden. Die Stärke der eingefügten Störung regelt dabei die Unschärfe der Positionsdarstellung. Die Idee ist aus dem Alltag bekannt: Das GPS-System zur Bestimmung von Positionen auf der Erde arbeitet nach diesem Prinzip. Sofern die Entfernung zu genug bekannten An-

kerpunkten (den Satelliten) bekannt ist, kann daraus die Position auf der Erde ermittelt werden. Sind die Entfernungsmessungen durch z.B. Reflexionen der Signale an einem Berg, leicht verfälscht, kann die Position nur ungenau bestimmt werden.

Eine formale Definition dieses Konzeptes mittels Graphentheorie wird in [GBC+06] gegeben. Goldenberg et al. kommen hier jedoch zu dem Schluss, dass es grundsätzlich möglich ist solche ungenauen Positionen auf recht exakte Punkte zurückzuführen, sofern nur genügend Ankerpunkte vorhanden sind.

Grundsätzlich bietet dieses Verfahren eine einfache und elegante Möglichkeit Positionen ungenau zu codieren. Leider kollidiert dieses Verfahren, neben der Möglichkeit Positionen zurückzurechnen, auch mit der eingangs definierten Forderung, dass eine Entfernungsbestimmung zwischen zwei Punkten möglich sein muss. Da lediglich die Entfernung zu den Ankerpunkten bekannt ist, müssen die exakten Positionen der Punkte im $\mathbb{R}^3$ berechnet werden, damit diese Berechnung möglich ist. Damit ist eine solche Darstellung für den Anwendungskontext entsprechend nicht geeignet.

4.2.2 Nicht-injektive Abbildungen

Wir können allgemein zwei Typen von Funktionen unterscheiden: Zum einen injektive Abbildungen, also Abbildungen, die sicherstellen, dass ein Punkt in der Zielmenge genau ein Urbild in der Quellmenge hat. Es wird also eine eindeutige Zuordnung zwischen zwei Punkten hergestellt. Das Problem einer solchen Funktion ist offensichtlich: Sie bietet keine große Hürde, da alle möglichen Zielpunkte bekannt sind (die Sphäre, siehe 4.1), und die Funktion somit recht einfach invertierbar ist. Selbst wenn die Abbildungsvorschrift nicht bekannt wäre, bieten injektive Abbildungen keinen Schutz vor in Abschnitt 4.1 genannter Methode der Zurückrechenbarkeit mittels Geometrie. Punkte können nach Definition der Injektivität eindeutig auf ihren Ursprungspunkt zurückgerechnet werden.

Die einzige Möglichkeit bieten also nicht-injektive Abbildungen, z.B. Hashfunktionen, die explizit darauf ausgelegt sind, dass es zu einem Bild mehrere mögliche Urbilder gibt. Hashfunktionen bilden Punkte der Quellmenge in eine Menge von diskreten „Buckets“ ab. Sie bieten also eine Abbildungen einer Quellmenge in eine kleinere Zielmenge an, und realisieren somit den erwünschten Genauigkeitsverlust. Je kleiner die Zahl der Buckets ist, desto ungenauer sind die abgebildeten Positionen.

Klassische Hashfunktionen haben den Vorteil, dass sie nur schwer, oder gar nicht zurückrechenbar sind, jedoch verstreuen diese Funktionen die abgebildeten Punkte auch mehr oder weniger gleichmäßig auf die möglichen Buckets, ohne auf die relative Position der Punkte zu achten. Damit ist die Position eines Nutzers zwar nicht mehr eindeutig zurück-

rechenbar, jedoch tritt nun das Problem auf, dass wir durch die scheinbar willkürliche Einordnung in die Buckets plötzlich Übereinstimmungen mit Benutzern aus aller Welt erreichen. Die Distanzerhaltung ist nicht mehr gegeben.

4.2.3 Nicht-injektive Abbildungen mit Locality Sensitivity

Eine Möglichkeit bieten Locality Sensitive Hash Funktionen, bei denen nahe beieinander liegende Punkte auch auf nahe beieinander liegende Punkte abgebildet werden. Das Konzept wurde von Indyk und Motwani in [IM98] eingeführt. LSH Funktionen wurden dort auf folgende Weise definiert:

Eine Familie $\mathcal{H} = \{h : S \to U\}$ heißt (r_1, r_2, p_1, p_2) -sensitiv auf einem Distanzmaß D, wenn für alle $v, q \in S$ gilt:

- Falls $v \in B(q, r_1)$, dann $\Pr_{\mathcal{H}}[h(q) = h(v)] \geq p_1$
- Falls $v \notin B(q, r_2)$, dann $\Pr_{\mathcal{H}}[h(q) = h(v)] \leq p_2$

$B(v, r) = \{q \in X \mid d(v, q) \leq r\}$ bezeichne die Menge aller Punkte im Radius r um v, wobei d(v,q) eine geeignete Distanzfunktion darstellt. Eine geeignete LSH-Familie erfüllt $p_1 > p_2$ und $r_1 < r_2$. Eine LSH-Funktion nach dieser Definition bildet somit nahe beieinander liegende Punkte mit hoher Wahrscheinlichkeit auf den gleichen Hashwert ab.

Das primäre Aufgabengebiet solcher Funktionen sind effiziente Lösungen für das Nearest-Neighbor Problem, speziell in hochdimensionalen Räumen. In diesem Szenario bildet man alle Punkte des Datenbestandes mit dem LSH Algorithmus auf diskrete Buckets ab. Werden nun die nächsten Nachbarn zu einem Punkt p gesucht, reicht es die Punkte des Buckets zu überprüfen, in das p eingeordnet wurde. Die Ortssensitivität des Algorithmus sollte sicherstellen, dass die dort befindlichen Punkte die gewünschte Nähe zu p aufweisen. Dieses Verfahren wurde beispielsweise für effiziente Speicherzugriffsalgorithmen, Image Retrieval und Data Mining Anwendungen vorgeschlagen, in denen häufig in hochdimensionalen Räumen gearbeitet werden muss[SC08].

Mittlerweile wurden verschiedene konkrete Funktionen vorgeschlagen, die diese Anforderungen erfüllen. Die Fähigkeit dieser Familie von Funktionen trifft auch den Kern der Anforderung für diesen Anwendungskontext. LSH Funktionen würden die Distanzerhaltung zumindest grob garantieren, allerdings tritt hier das gleiche Problem auf, dass diese Positionen approximativ über die Geometrie und bekannte Punkte zurückrechenbar sind. Wie bei klassischen Hashfunktionen kann hier über die Größe, bzw. Anzahl der Buckets im Zielraum die resultierende Genauigkeit der Position vorgegeben werden.

4.3 Zusammenfassung und Bewertung

Die erhoffte komplette Anonymisierung von Positionen scheint aufgrund der gegebenen Anforderungen, die durch die Anwendung gegeben sind, nicht möglich zu sein. Allerdings bieten sich eine Reihe von Möglichkeiten, die Positionen zumindest so darzustellen, dass sie nur ungenau auf ihre exakte Position schließen lassen. Von allen Möglichkeiten bietet Locality Sensitive Hashing die beste Möglichkeit. Ein relativ einfacher Algorithmus, der die LSH-Definition auf einer Fläche fester Größe erfüllt, wird im nächsten Kapitel im Detail vorgestellt.

Kapitel 5

Quad Keys

5.1 Definition und Funktionsweise

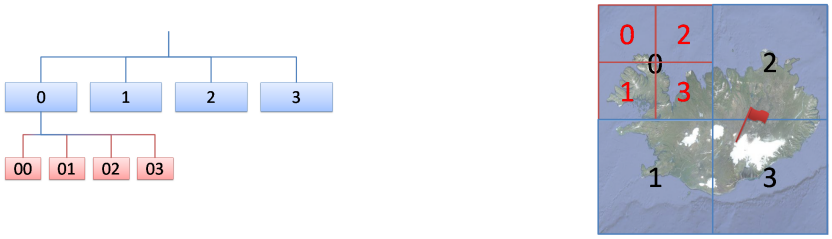
5.1.1 Funktionsweise

Das Prinzip der Quad Keys [Mic12a] baut auf der Idee von Quadtrees [FB74] auf, einer Datenstruktur, die häufig in der Computergrafik Verwendung findet. Ein Quadtree ist eine spezielle Baumstruktur, bei der jeder innere Knoten genau vier Kinder hat. Das Quad Keys Verfahren teilt eine quadratische Fläche rekursiv in vier Teilquadrate auf, wobei jedem Teilquadrat ein Index zugewiesen wird. Ein Quad Key besteht dann aus einer Sequenz von Indices der Teilquadrate. Abbildung 5.1 zeigt eine solche Aufteilung, und den dazugehörigen Quadtree. Auf diese Weise können Quad Keys leicht unterschiedliche Präzisionen darstellen, und bestehende Quad Keys können ohne nennenswerten Aufwand in ihrer Präzision modifiziert werden. Je höher die Rekursionstiefe, desto genauer wird die Darstellung. Reduzierungen der Genauigkeit eines bestehenden Quad Keys werden einfach durch Verkürzung der Indexsequenz realisiert.

Der Quad Keys Algorithmus bildet damit Punkte auf einer 2D-Fläche auf eine Menge von diskreten Quad Keys ab. Dabei wird berücksichtigt, dass nahe beieinander liegende Punkte mit hoher Wahrscheinlichkeit auf den identischen Key abgebildet werden. Das Verfahren genügt damit den Anforderungen des Locality Sensitive Hashing. Es sei jedoch betont, dass das Verfahren durch die Voraussetzungen nur auf einer fest definierten, begrenzten Fläche zu arbeiten, ein sehr eingeschränkter Vertreter dieser Klasse von Funktionen ist.

5.1.2 Unterschied Quad Keys zu Quadtrees

Das Quad Keys Verfahren adaptiert einen Großteil von dem Prinzip der Quadtrees [FB74]. Ein wichtiger Unterschied verbleibt jedoch: Ein



(a) Quadtree **(b)** Rekursive Aufteilung der Karte

Abbildung 5.1: Darstellung des Prinzips der rekursiven Aufteilung und Indizierung. Die gekennzeichnete Fläche (Flagge) in (b) würde bspw. durch den Key 30 beschrieben werden.

Quadtree ist eine Datenstruktur, wohingegen Quad Keys die Position eines Punktes beschreiben.

Der Aufbau eines Quadtrees funktioniert grob nach folgendem Prinzip: Anfangend mit einem Wurzelknoten werden Punkte so lange diesem Knoten zugeordnet, bis eine gewisse Obergrenze erreicht wird. Dann erst wird der Knoten in vier Unterknoten gespalten. Diese Vorgehensweise wird entsprechend auch für tiefere Knoten so fortgeführt. Ein Punkt, der als Quad Key dargestellt wird, wird mit einer bestimmten Rekursionstiefe berechnet, die vorgegeben wird. Ein Punkt, der in einen Quadtree eingeordnet wird, wird, abhängig vom Zustand des Baumes, auf einer höheren oder tiefen Ebene des Baumes eingeordnet. Im Laufe der Zeit kann sich diese Position zudem ändern.

5.1.3 Konventionen im weiteren Teil der Arbeit zu Quad Keys

Nach den Anforderungen in Abschnitt 1.4 gehen wir davon aus, dass ein System mit Quad Keys unterschiedlicher Länge arbeiten muss. Ein Quad Key wird daher im folgenden als 2-Tupel aus *Key* und *Keylength* definiert. Der Key entspricht der Sequenz der Indices, wobei per Konvention der höchstwertige Index am Anfang steht. Die Keylength gibt die Anzahl Indices an, die der Key enthält. Dies ist in erster Linie für die Implementierung des Verfahrens notwendig.

Für die Darstellung des Keys sind verschiedene Varianten möglich. Die Variante, die in Abbildung 5.1 verwendet wurde, ist für einen menschlichen Leser am vorteilhaftesten. Die Teilquadrate werden hier mit Zahlen 0-3 codiert. Für die Implementierung bietet sich eine Binärcodierung dieser Variante an, wobei die Zahlen 0-3 mit ihrer entsprechenden Binärdarstellung codiert werden. Der resultierende Binärstring kann, als Zahl interpretiert, als solche direkt in einem entsprechenden Datenbankfeld gespeichert werden, über das jedes gängige DBMS verfügt.

Der gekennzeichnete Teilbereich in Abbildung 5.1 (b) würde dann durch den Quad Key 30 (0-3 Codierung), 1100 (Binärcodierung), bzw. 12 (Zahlintepretation) dargestellt werden.

5.2 Abbildung von Ortspositionen auf Quad Keys

5.2.1 Abbildung von Lat/Long Koordinaten auf eine quadratische 2D-Karte

Das Quad Keys Verfahren arbeitet auf einer quadratischen 2D-Fläche, daher müssen die Positionen der Erdkugel auf eine entsprechende Darstellung überführt werden. Eine Überführung einer Kugeloberfläche auf eine zweidimensionale Fläche bringt dabei in jedem Fall Verzerrungen mit sich. Verfahren, die Effekte dieser Verzerrungen möglichst gering zu halten, werden seit Jahrhunderten im Bereich der Kartographie und Geodäsie entwickelt. Eine verbreitete Methode ist die Mercator-Projektion, die u. A. von Google Maps verwendet wird [Goo12b]. Diese Projektion bringt zwar Verzerrungen mit sich, erhält jedoch Winkel, was z.B. für Straßenkarten unabdingbar ist.

Wir definieren die Größe der resultierenden Karte in Pixeln, wobei die kleinstmögliche Einheit 1 Pixel ist. Da der Quad Keys Algorithmus diese Karte rekursiv in Länge und Breite halbieren wird, sollte die Karte eine Länge, bzw. Breite von 2^x haben, damit in der tiefsten Rekursionsebene genau ein einzelnes Pixel angesprochen wird. Somit kann ein Quad Key die Karte in ihrer bestmöglichen Genauigkeit darstellen, d.h. jedes Pixel kann eindeutig durch einen Key dargestellt werden. Die Variable x wird im Folgenden als *Detailgrad* bezeichnet.

Mit einem mittleren Erdradius von $R = 6371$ km, und dementsprechend einer Äquatorlänge von $L = 2\pi R \approx 40.030$ km, kann in Abhängigkeit der Größe der Karte 2^x die Entsprechung eines Pixels in Metern auf der Erde bestimmt werden.

$$s_p = \frac{2\pi R}{2^x} = \frac{2\pi\ 6371 \cdot 10^3}{2^x} m$$

Der Detailgrad x der Karte entspricht damit auch der tiefsten Rekursionsebene, und legt damit die Länge des Keys auf Keylength = x, für einen Quad Key maximaler Genauigkeit, fest. Für einen Key in Binärcodierung werden dafür $2x$ Bits benötigt. Mit einem 32-Bit Integer lässt sich somit eine Genauigkeit von 610.81 m, erreichen. Mit 64 Bit erhöht sich die Genauigkeit auf $9.32 \cdot 10^{-3}$ m, was bereits weit jenseits des Auflösungsvermögen des GPS-Positionierungssystems liegt. Tabelle 5.1 zeigt einige markante Werte. Eine Genauigkeit von 26 sollte eine gute Wahl sein, um eine Position exakt wiederzugeben.

Nach der Abbildungen einer Position mittels einer geeigneten Projektion, liegt die Position nun als (x, y) Tupel vor, wobei die Tupel auf einzelne Pixel der Karte verweisen.

Detailgrad	Genauigkeit in Metern
1	20.015 m
2	10.008 m
...	...
22	9.54 m
23	4.77 m
24	2.39 m
25	1.19 m
26	0.59 m
...	...

Tabelle 5.1: Entsprechung eines Pixels in Metern

5.2.2 Abbildung von Kartenkoordinaten auf Quad Keys

Entsprechend der Definition, teilt der Algorithmus nun die Karte rekursiv in Teilquadrate auf, und bestimmt so die Position $p = (x, y)$ als Sequenz der Indices. Der Algorithmus ist im folgenden dargestellt. Die Ausgabe erfolgt in diesem Fall als Binärcodierung.

Algorithmus 5.1: Implementierung der (x,y) –> Quad Keys Abbildung

```
1  public QKey XYToQuadKey(int pixelX, int pixelY){
2      long key = 0;
3      int size = mapSize / 2;
4
5      for(int i = 0; i < maxDetailDepth; i++){
6        key |= sign(pixelY - size);
7        key <<= 1;
8        if(sign(pixelY - size) == 1) pixelY -= size;
9
10       key |= sign(pixelX - size);
11       key <<= 1;
12       if(sign(pixelX - size) == 1) pixelX -= size;
13
14     size /= 2;
15     }
16     key >>= 1;  // Shift last position back
17     return new QKey(key, maxDetailDepth);
18 }
19
20 public static int sign(double number){
21     if(number >= 0) return 1;
22     else return 0;
23 }
```

5.3 Matching mit Quad Keys

Quad Keys bieten zwei Möglichkeiten Entfernungen zwischen zwei Keys zu bestimmen. Dabei muss jedoch bedacht werden, dass zwei Keys, die nicht die maximal mögliche Länge besitzen, nur Ober- und Untergrenzen ihrer Entfernung angeben können. Abbildung 5.2 (a) zeigt zwei solche Keys, die jeweils ein Teilquadrat der Karte beschrei-

(a) Euklidische Distanz **(b)** Hierarchischer Aufbau

Abbildung 5.2: Darstellung der Entfernung zwischen zwei Quad Keys. (a) zeigt die euklidische Distanz mit Unsicherheitsbereich. (b) zeigt das Prinzip des hierarchischen Aufbaus. Die Quad Keys 101 und 13 stimmen auf einem Index überein. Die maximale Distanz ist durch die Diagonale eines Teilquadrats auf Rekursionsebene 1 gegeben.

ben. Es ist nun möglich die Entfernung der Mittelpunkte der Quadrate zu bestimmen, und die jeweiligen Ausdehnungen der Quadrate als Unsicherheitsbereich mit anzugeben. Als Matchings gelten nun zwei Keys, die innerhalb eines bestimmten Abstandes liegen.

Diese Berechnung liefert die bestmögliche Entfernungsbestimmung zwischen zwei Quad Keys, ist jedoch vergleichsweise aufwendig. Die hierarchische Beschaffenheit der Quad Keys erlaubt eine deutlich einfachere Möglichkeit Matchings zu bestimmen. Grundsätzlich gilt, dass zwei Punkte, bei deren Quad Keys die führenden k Indices übereinstimmen, auf den ersten k Rekursionsebenen in den selben Teilquadraten liegen (Abb. 5.2 (b)). Die maximale Distanz d, die zwischen zwei Punkten in einem Quadrat liegen kann, ist $d = \sqrt{2a^2}$, wobei a die Seitenlänge des Quadrats ist. Die zwei Punkte liegen in diesem Fall auf zwei gegenüberliegenden Ecken; die Distanz ist die Diagonale. Die maximale Distanz $\bar{d}$ zwischen zwei Punkten mit k übereinstimmenden führenden Indices ist dann:

$$\bar{d} = \sqrt{2s_p^2} = \sqrt{2\left(\frac{2\pi R}{2^k}\right)^2}$$

Die Werte für verschiedene k können vorab berechnet werden:

Anzahl Indices k	Max. Distanz $\bar{d}$ in Metern
1	28.305 m
2	14.153 m
...	...
22	13.49 m
23	6.75 m
24	3.38 m
25	1.68 m

Tabelle 5.2: Maximale Distanz zweier Punkte in einem Quad Key

(a) Approxmation des Umkreises **(b)** Mögliche Abweichung

Abbildung 5.3: Darstellung des Nachbarschaft-Matchings zwischen zwei Quad Keys. (a) zeigt die Approximation eines radialen Umfelds mittels benachbarten Bereichen. (b) zeigt die mögliche Abweichung bei nicht-mittiger Lage des Ursprungspunktes.

Um zu bestimmen, ob die Entfernung zweier Quad Keys unter einer bestimmten Obergrenze d liegt, genügt es einfach zu überprüfen ob die zwei Keys auf der zu d gehörigen Rekursionsebene benachbart sind. Abbildung 5.3 (a) zeigt, wie auf diese Weise eine radiale Umgebung auf dem Gitter approximiert wird. Im Folgenden wird Matching mit diesem Verfahren als *Nachbarschafts-Matching* bezeichnet.

Da die Position des ursprünglichen Punktes in dem Quad Key Bereich nicht mehr genau bekannt ist, tritt auch hier ein Unsicherheitsbereich auf. Sei $\bar{d}$ die Länge der Diagonale des Bereichs. Liegt der Punkt genau mittig in dem Bereich, so bietet die Nachbarschaft eine annähernd gleichmäßige Approximation eines radialen Umfeldes mit Radius $\frac{3}{2}a = \frac{3}{4}d$. Liegt der Punkt jedoch bspw. in der oberen rechten Ecke, so ist die Approximation eines radialen Umfeldes nicht mehr gegeben. Abbildungen 5.3 (b) stellt dies dar. Es gibt hier in den Abständen zu den Rändern des Umfeldes erkennbare Asymmetrien, wobei die größte Entfernung zum Punkt nun $2d$, und die kleinste $a = \frac{1}{2}d$ ist. Ebenso wie beim zuerst genannten Verfahren kann es also auch hier passieren, dass Punkte als benachbart gewertet werden, die es eigentlich nicht sind, bzw. nicht gewertet, obwohl sie es eigentlich wären.

Algorithmus 5.2 zeigt eine Implementierung für dieses Verfahren, die überprüft ob zwei Quad Keys benachbart sind.

Algorithmus 5.2: Implementierung des Nachbarschafts-Matchings

```
1    boolean isWithinNeighborship(QKey qkey1, QKey qkey2, int
         detailDepth){
2      qkey1.reduceToDetailDepth(detailDepth);
3      qkey2.reduceToDetailDepth(detailDepth);
4
5    setMapDetailDepth(detailDepth);
6      QPoint middlePoint = QuadKeyToMapXY(qkey1);
7      QPoint neighborPoint = QuadKeyToMapXY(qkey2);
8
9      if(Math.abs(middlePoint.getX() - neighborPoint.getX())
           <= 1){
```

```
10          if(Math.abs(middlePoint.getY() - neighborPoint.getY
                ()) <= 1){
11            return true;
12          }
13        }
14        return false;
15      }
```

Im Vergleich zum erstgenannten Verfahren, das für die Entfernungsberechnung die klassische Euklidische Entfernungsberechnung verwendet, sind hier keine aufwendigen Wurzelfunktionen notwendig. Auf der anderen Seite erlaubt das zweite Verfahren nur die Verwendung von festen Obergrenzen, und macht keine Aussagen über den Unsicherheitsbereich einer Berechnung. Für zwei Quad Keys, die mit maximaler Genauigkeit jeweils ein einzelnes Pixel ansprechen, kann das erste Verfahren die genaue Distanz bestimmen, während das zweite Verfahren nur die Obergrenze angibt, obwohl eine genaue Bestimmung möglich wäre. Im gegebenen Anwendungskontext ist diese Obergrenze jedoch ausreichend. Die in Tabelle 5.2 dargestellten möglichen Obergrenzen von $k = 22, 23$ oder 24 sollten ausreichen, um ein Matching zwischen zwei Positionen hinreichend genau erkennen zu können.

5.4 Datenstrukturen für Positionen mit Quad Keys

Der Aufbau der Quad Keys impliziert, dass ein Punkt mit einem Key der Länge l, auch immer in dem Bereich eines Keys der Länge $l - i$ für $i < l$ enthalten ist. Die führenden Indices bleiben dabei erhalten. Dies folgt durch die weitere Aufteilung der Teilquadrate. Ein Key wird durch Ergänzung weiterer Indices feiner definiert, wobei die bisherigen Indices unverändert bleiben. Abbildung 5.1 (a) stellt das Prinzip als Baumstruktur dar. Dies entspricht dem Aufbau des klassischen Quadtrees.

Eine solche Baumstruktur kann zum Speichern von Positionsdaten verwendet werden. Die Daten werden dazu in Mengen gespeichert, wobei alle Positionsdaten mit identischen Quad Keys in einer Menge gespeichert werden, die dem jeweiligen Knoten im Baum zugeordnet wird. Werden zu einer Position p nun Positionen gesucht, deren Keys auf den führenden k Bits übereinstimmen, so müssen nur die entsprechenden Mengen in dem Teilbaum, der zu p führt überprüft werden. Um schnelle Nachbarschaftssuchen zu ermöglichen, können die Nachbarn einmalig bei Erstellung der Struktur vorberechnet werden, und die Zeiger auf die Nachbarn in den Knoten abgelegt werden.

Je nach Vorgehensweise der Anwendung bieten sich auch Algorithmen an, die typischerweise in Quadtrees Verwendung finden. Für die Suche nach Nachbarn in einem bestimmten Umkreis bietet sich ein einfacher Algorithmus [Ind01] an, der in Abbildung 5.3 dargestellt ist. Da Quadtrees gerade für solche Aufgaben vielfältige Anwendungen fin-

den, sind zudem verschiedene effizientere Verfahren entwickelt worden (siehe Kapitel 3).

Algorithmus 5.3: Vorgehensweise für Suche nach Nachbarn zu einem Punkt p in einem Umkreis mit Radius r

```
1    Lege Wurzel des Baumes auf den Stack
2       while(Stack nicht leer){
3         Nehme den nächsten Knoten k vom Stack
4       forall(Kindknoten c von k){
5         if(c ist Blatt) Füge Punkte von c den Nachbarn hinzu
6         if(c schneidet den Kreis um p) Lege c auf den Stack
7       }
8     }
```

5.5 Zusammenfassung

Zusammenfassend erfüllen Quad Keys die notwendigen Anforderungen, die eingangs definiert wurden. Sie ermöglichen die Darstellung ungenauer Positionen, wobei die Genauigkeit durch den hierarchischen Aufbau sehr einfach reduziert werden kann. Dieser Aufbau ermöglicht zudem einen effiziente Berechnungen und Manipulationen der Quad Keys über Bitoperationen. Letztendlich auch nicht zu übersehen ist die relativ einfache, deterministische Funktionsweise, die eine überschaubare, und gut zu testende Implementierung in Aussicht stellt.

Damit wären die Grundvoraussetzungen gegeben. Es bleibt zu überprüfen in welchem Maß eine solch ungenaue Positionsdarstellung in dem gegebenen Kontext überhaupt verwendbare Resultate liefern kann. Das nächste Kapitel widmet sich dieser Evaluation von Quad Keys für die entsprechenden Anforderungen.

Kapitel 6

Evaluation

6.1 Ziel der Untersuchungen

In diesem Kapitel soll die Anwendbarkeit von Quad Keys, insbesondere mit verringerter Genauigkeit, für den gegebenen Anwendungskontext untersucht werden. Nach Abschnitt 1.4 soll dabei die Berechnung von Matchings, zusammenhängenden Sequenzen und Ortsbewertungen evaluiert werden. Letztendlich sollen dabei zwei Fragen geklärt werden:

- Inwieweit sind Quad Keys (mit reduzierter Genauigkeit) für den gegebenen Anwendungskontext verwendbar?
- Wie stark kann die Genauigkeit reduziert werden, um noch mit akzeptablen Fehlerraten arbeiten zu können?

6.2 Evaluationsumgebung

Als Grundlage für die folgende Evaluation wird ein Datensatz [Mic12b] verwendet, der von Microsoft Research für die Untersuchung von Positionsverläufen im Rahmen der Arbeiten von Zheng et al. ([ZZXM09], [ZLC+08], [ZXM]) erstellt wurde. Die Sammlung umfasst einen Datenbestand von 17.621 Positionsverläufen mit einer Gesamtlänge von 1,2 Millionen Kilometern und einer Dauer von über 48.000 Stunden. Die Daten wurden im Laufe von drei Jahren von 178 Testnutzern erfasst. Der größte Anteil der Daten wurde dabei im Großraum Beijing, China aufgenommen.

Bei den Auswertungen wurde Wert darauf gelegt möglichst gleiche Arbeitsbedingungen zu schaffen. Alle Auswertungen wurden auf dem gleichen Rechner unter Verwendung der gleichen Laufzeitumgebung und Datenbank durchgeführt. Beim Vergleich verschiedener Verfahren wurde soweit wie möglich auf Gleichheit des verwendeten Codes geachtet. So ist die Implementierung der Algorithmen zur Berechnung von Matchings und Sequenzen für das Längen/Breitengrad- und das

Quad Keys Verfahren identisch. Nur die Art der Distanzbestimmung ist notwendigerweise unterschiedlich.

6.3 Evaluation: Vergleich Längen-/Breitengrade zu Quad Keys

In Abschnitt 5.3 wurde das Verfahren des Nachbarschafts-Matching für Quad Keys vorgestellt. Während in der exakten Darstellung mit Längen- und Breitengraden Entfernungen präzise berechnen können, bringt das Nachbarschafts-Matching einen Unsicherheitsbereich mit sich. Genauer gibt es eine maximale Entfernung von $2\bar{d}$, und eine minimale Entfernung von $\frac{1}{2}\bar{d}$, in der nach Matchings gesucht wird. Da das Verfahren letztendlich möglichst genau einen Umkreis approximieren soll, um Matchings zu finden, ist zu überprüfen inwieweit diese Eigenschaft erfüllt wird.

6.3.1 Konventionen

Im folgenden werden häufig Ergebnisse verschiedener Verfahren oder Parametrisierung anhand der Anzahl gefundener Matchings, bzw. Sequenzen verglichen. Wichtig ist im direkten Vergleich die Anzahl neuer, bzw. alter Sequenzen. Eine Sequenz gilt im Ergebnis e_1 als alt, wenn sie in einem Referenzergebnis e_2 ebenfalls vorhanden ist. Dies gilt auch, wenn die Sequenz in e_2 kürzer als die in e_1 ist, die Sequenz in e_2 diejenige in e_1 also vollständig enthält. Eine Sequenz in e_1 gilt folglich als alt wenn sie eine Sequenz in e_2 erweitert oder mit ihr identisch ist. Eine Sequenz gilt entsprechend in e_1 als neu, wenn sie in e_1 enthalten ist, jedoch nicht in e_2.

In Tabellen und Abbildungen werden zwecks Lesbarkeit die folgenden Abkürzungen verwendet:

- **NM-24**: Nachbarschafts-Matching auf Detailgrad 24 (andere Detailgrade analog)
- **Lat/Long**: Matching über Haversine-Entfernungsberechnung mit Positionen in Längen-/Breitengraden, Radius wird explizit mit angeben

6.3.2 Anzahl Matchings/Sequenzen

Im ersten Schritt vergleichen wir die Ergebnisse des Nachbarschafts-Matchings mit den Matching mit exakten Radien, um festzustellen welche Radien tatsächlich dem Nachbarschafts-Matching am ehesten entsprechen. Betrachtet werden die Detailgrade 23 und 24, die in Abschnitt 5.3 als sinnvolle Größen für die Suche nach Übereinstimmungen bestimmt wurden. Tabellen 6.1, A.1 (Anhang) geben jeweils die Anzahl gefundener Matchings für NM-24, bzw. NM-23, sowie die entsprechenden Lat/Long Ergebnisse an. Die Ergebnisse für NM-23 sind vergleichbar mit NM-24, und werden daher nicht hier dargestellt.

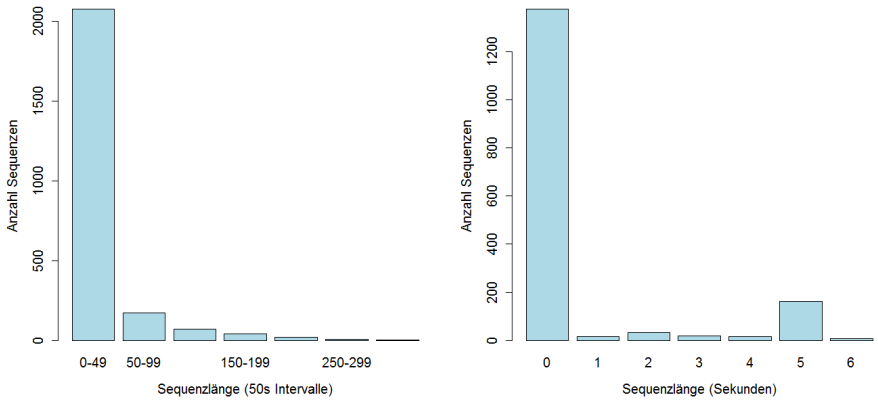
Variante	Anzahl Matchings	Anzahl Sequenzen
Nachbarschafts-Verfahren	4.086.638	29259
Lat/Long mit $r = \frac{1}{2}\bar{d}_{24}$	3.170.965	27685
Lat/Long mit $r = \frac{3}{4}\bar{d}_{24}$	3.776.450	29193
Lat/Long mit $r = 1\bar{d}_{24}$	4.344.625	29156
Lat/Long mit $r = \frac{3}{2}\bar{d}_{24}$	5.418.791	28395
Lat/Long mit $r = 2\bar{d}_{24}$	6.399.912	27166

Tabelle 6.1: Anzahl Matchings von NM-24 im Vergleich zu verschiedenen Lat/Long Radien

Am ehesten entsprechen die Radien $r = \frac{3}{4}\bar{d}$ und $r = 1\bar{d}$ den Ergebnissen des Nachbarschafts-Matchings. Im Folgenden werden diese drei Ergebnisse genauer betrachtet.

6.3.3 Untersuchung neuer Sequenzen

Wir betrachten die Ergebnisse zu NM-24. Im Vergleich zum Matching mit $r = \frac{3}{4}\bar{d}_{24}$ wurden von NM-24 2418 neue Sequenzen mit einer Gesamtlänge von 980 Minuten gefunden. Abbildung 6.1 gibt die Verteilung dieser neuen Sequenzen anhand ihrer Länge an. Es zeigt sich, dass der überwiegende Teil (85.5%) im Bereich von 0-55 Sekunden liegt. Betrachtet man diesen Bereich genauer, zeigt sich, dass 56,7% aller Sequenzen eine Länge von 0 Sekunden aufweisen. Dabei handelt es sich um Sequenzen, die aus genau einem einzigen Punkt bestehen. Solche Sequenzen sind wenig aussagekräftig, und würden in einer realen Anwendung vermutlich verworfen werden. Die restlichen Sequenzen erreichen zusammen eine Länge, die gerade einmal 0.65% der Gesamtlänge des Lat/Long Ergebnisses entspricht. Die Abweichung ist also nur minimal.



(a) Längenverteilung (0-350s) **(b)** Längenverteilung (0-6s)

Abbildung 6.1: Längenverteilung der neuen Sequenzen von Lat/Long $r = \frac{3}{4}\bar{d}_{24}$ relativ zu NM-24. (Vollständige Diagramme unter A.1)

Die Auswertung mit $r = 1\bar{d}$ findet nun 2924 neue Sequenzen im Vergleich zu NM-24 mit einer Gesamtlänge von 3608 Minuten. Auch hier zeigt sich, dass ein Großteil auf den 0-Sekunden Bereich fällt. Hier fallen 62,1% in den Bereich von 0-55 Sekunden, und 31,5 in den 0 Sekunden Bereich. Die Gesamtlänge neuer Sequenzen erreicht 2,3% der Länge von NM-24. Allgemein zeigt diese Auswertung größere Abweichungen als die zuerst betrachtete mit $r = \frac{3}{4}\bar{d}_{24}$. Ähnliche Ergebnisse zeigen sich für die Auswertung mit Detailgrad 23.

Wie erwartet kann das Nachbarschafts-Matching einen Umkreis wie beim Lat/Long Verfahren nur ungenau approximieren. Eine relativ gute Annäherung lässt sich dennoch für $r = \frac{3}{4}\bar{d}$ beobachten. Die Abweichungen dieser Ergebnisse sind dergestalt, dass von einer annehmbaren Approximation gesprochen werden kann. Damit sind Quad Keys für den gegebenen Anwendungskontext grundsätzlich verwendbar.

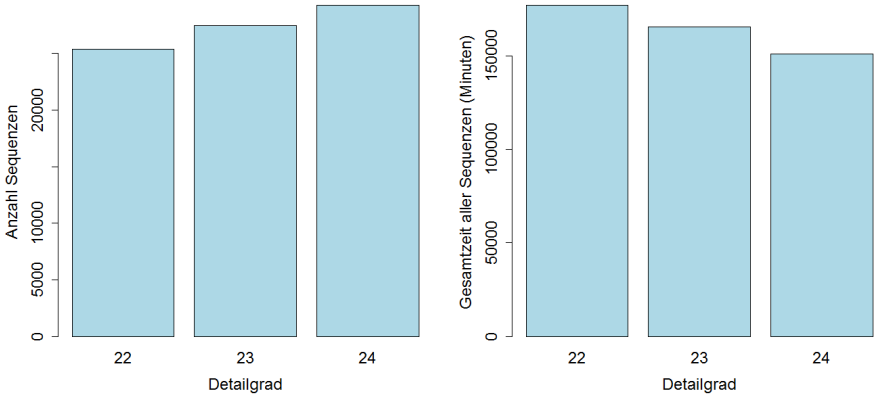
6.4 Evaluation: Finden von Sequenzen

Nach der vorherigen Untersuchung über die Anwendbarkeit von Quad Keys an sich, soll nun die Auswirkung unterschiedlicher Positionsgenauigkeiten auf die berechneten Ergebnisse untersucht werden. Eine der Voraussetzungen an das Verfahren war, dass der Nutzer die Genauigkeit seiner Position reduzieren kann. Ziel ist es zu untersuchen, inwiefern sich diese reduzierten Genauigkeiten auf die gefundenen Sequenzen auswirken.

Wir wählen dazu NM-24 als Referenz, und untersuchen wie sich die Ergebnisse der weniger genauen NM-23 und NM-22 dazu verhalten. Es ist zu erwarten, dass niedrigere Detailgrade aufgrund des größeren Suchradius mehr Matchings finden. Die Frage ist, inwiefern diese Änderungen die tatsächlichen Ergebnisse verfälschen.

6.4.1 Auswirkung auf die Anzahl der Sequenzen

Abbildung 6.2 (a) zeigt die Veränderung der Anzahl gefundener Sequenzen, (b) die Gesamtlänge aller gefundener Sequenzen. Da mit kleinerem Detailgrad der Radius einbezogener Punkte steigt, zeigt sich entsprechend auch eine Steigung der Gesamtzeit aller Sequenzen. Diese Entwicklung stimmt mit der Gesamtanzahl gefundener Matchings überein (A.2). Interessant ist, dass die Anzahl Sequenzen mit abnehmender Genauigkeit sinkt. Offenbar bewirkt die höhere Anzahl Matchings, dass bestehende Sequenzen nun zusammengeführt werden (Erinnerung: Die Bedingung, damit Punkte einer Sequenz angehören, ist, dass sie einen maximalen zeitlichen Abstand zu einem Vorgängerpunkt einhalten. Diese Zeitgrenze wurde für alle Untersuchungen nicht geändert). Eine ähnliche Entwicklung konnte bereits bei der Betrachtung unterschiedlicher Radien in Abschnitt 6.3.2 beobachtet werden; dieser Effekt ist also nicht dem Quad Keys Verfahren geschuldet.

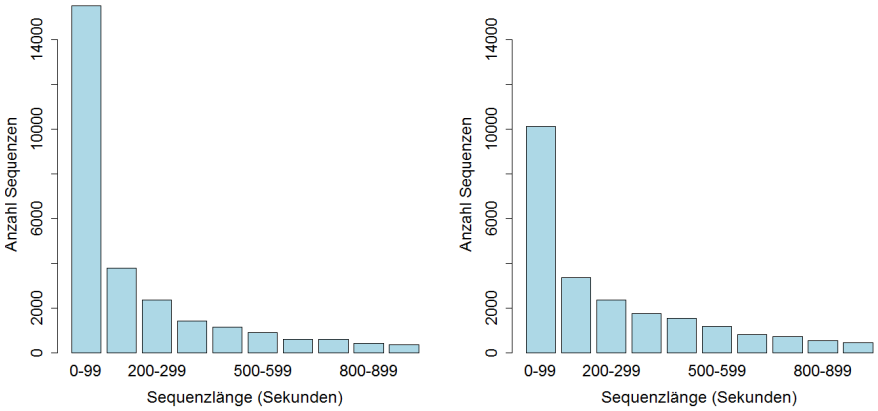


(a) Anzahl Sequenzen **(b)** Gesamtlänge aller Sequenzen

Abbildung 6.2: Entwicklung der Matchings bei unterschiedlichem Detailgrad.

6.4.2 Auswirkung auf die Länge der Sequenzen

Die Vermutung, dass Sequenzen zusammengeführt werden, bestätigt sich durch Analyse der Sequenzlängen. Abbildung 6.3 stellt die Verteilung der Sequenzen auf unterschiedliche Längen dar. Dargestellt werden die relevanten Ausschnitte von NM-24 und NM-22. NM-23 stellt eine Zwischenstufe zwischen beiden dar, und ist mit den vollständigen Daten unter Abschnitt A.2.2 zu finden.



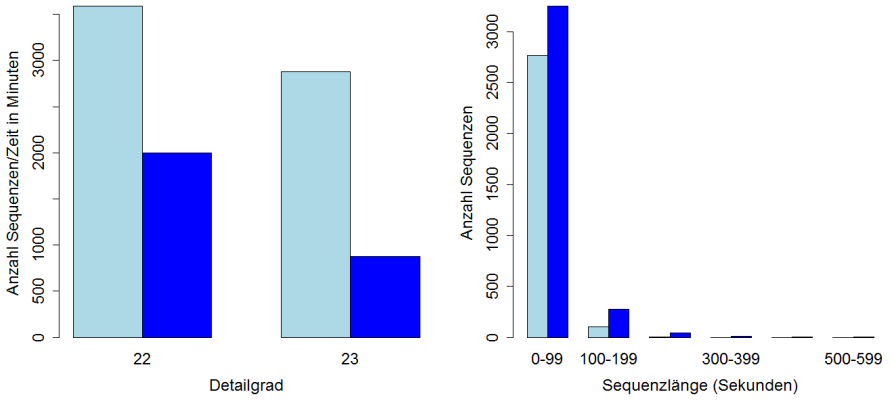
(a) Längenverteilung NM-24 **(b)** Längenverteilung NM-22

Abbildung 6.3: Verteilung der Sequenzlängen von NM-24 und NM-22

Es ist gut zu erkennen, dass viele kurze Sequenzen, die in NM-24 sehr zahlreich sind, in NM-22 in längere Sequenzen übergehen.

6.4.3 Auswirkung auf die Anzahl neuer Sequenzen

Um festzustellen, inwiefern niedrigere Detailgrade die Ergebnisse verfälschen, ist insbesondere interessant wie viele neue Sequenzen gefunden werden.



(a) Sequenzen & Gesamtzeit **(b)** Längenverteilung

Abbildung 6.4: (a) Veränderung der Anzahl neuer Sequenzen (hellblau) und ihrer Gesamtzeit (blau) von NM-23 und NM-22
(b) Verteilung der Länge neuer Sequenzen von NM-23 (hellblau) und NM-22 (blau)

Die Anzahl neu gefundener Sequenzen ist relativ gesehen eher niedrig. Verglichen mit NM-24 (29259 Sequenzen), werden 2877 (NM-23) und 3588 (NM-22) neue Sequenzen mit einer Länge von 874 (NM-23) und 2002 (NM-22) Minuten gefunden. In Relation zu der Gesamtlänge aller Sequenzen von 151215 Minuten ist diese neue Menge relativ gering. Ein Großteil der allgemein hinzugekommenen Zeitlänge (siehe Abschnitt 6.4.1) fließt offenbar in die Erweiterung bereits vorhandener Sequenzen, nicht in neue.

Abbildung 6.4 (b) zeigt darüber hinaus, dass ein Großteil dieser Zeit in kurze Sequenzen fließt. Ein genauerer Blick auf diesen Längenbereich zeigt, dass der überwiegende Teil neuer Sequenzen zudem in den 0-Sekunden Bereich fällt (Abbildung A.5, Anhang).

6.4.4 Anzahl neue Sequenzen mit neuen Benutzern

Bislang zeigten sich eher wenige gravierende Änderungen in den Ergebnissen. Es stellt sich die Frage ob überhaupt Sequenzen mit anderen Benutzern gefunden werden, oder nur bisherige erweitert werden.

In NM-24 hatten 100 Benutzer Übereinstimmungen mit 290 anderen Benutzern. In NM-23 belaufen sich die Übereinstimmungen bereits auf 110 Benutzern, die insgesamt Matchings mit 364 Benutzern hatten. Bei NM-22 sind es 118 Benutzer mit Matchings zu insgesamt 452 Benutzern.

Damit werden durch die ungenauere Positionsangabe (und den daraus resultierenden größeren Suchradius) tatsächlich falsche Matchings mit Benutzern induziert, die für eine Auswertung störend sein könn-

ten. Aus den vorherigen Untersuchungen wurde deutlich, dass diese Sequenzen mit neuen Benutzern jedoch vergleichsweise kurz sind. Inwieweit diese falschen Matchings tatsächlich eine gravierende Störung darstellen wird in Abschnitt 6.6 untersucht.

6.4.5 Filterung falscher Sequenzen

Für einen ersten Versuch, falsche Matchings bei ungenaueren Positionen auszusortieren, versuchen wir mit einem Filter Sequenzen kurzer Längen zu entfernen.

Wir beobachten die Anzahl Benutzer und Anzahl Nutzer mit Matchings zu diesen nach Filtern unterschiedlicher Längen:

Verfahren und Filter	Anzahl Benutzer	Matchings mit anderen Benutzern
NM-24 (-)	100	290
NM-24 (0s)	100	261
NM-24 (10s)	100	240
NM-24 (20s)	100	234
NM-24 (30s)	100	228
NM-23 (-)	110	364
NM-23 (0s)	110	308
NM-23 (10s)	110	263
NM-23 (20s)	110	251
NM-23 (30s)	110	248
NM-22 (-)	118	452
NM-22 (0s)	118	394
NM-22 (10s)	118	322
NM-22 (20s)	118	279
NM-22 (30s)	118	277

Tabelle 6.2: Benutzer-Matchings nach Filterung über die Sequenzlänge

Die Ergebnisse zeigen, dass solch einfache Filtermethoden bereits einige falsche Übereinstimmungen entfernen können. Offensichtlich gelingt dies nicht in allen Fällen, zumindest leichte Störungen können jedoch ausgefiltert werden. Legt man einen 30 Sekunden Filter an, zeigen sich für die drei Detailstufen bereits deutliche Unterschiede zu den ungefilterten Ergebnissen. Ein solcher Filter kommt für eine spätere Auswertung vermutlich ohnehin zum Einsatz, um wenig aussagekräftige Übereinstimmungen zu kurzer Länge zu entfernen.

6.4.6 Zwischenergebnis: Matchings und Sequenzen

In den bisherigen Schritten wurde erkennbar, dass Quad Keys prinzipiell für den gegebenen Anwendungskontext einsetzbar sind. Dennoch zeigen sich auch mit recht präzisen Detailgraden leichte Abweichungen von dem exakten Lat/Long, die sich jedoch noch in wenig problematischen Größenordnungen bewegen.

Mit abnehmendem Detailgrad zeigen sich nun durchaus größere Abweichungen, die zumindest zu Teilen ausgebessert werden können. Große Teile der Abweichungen fallen auf die Verlängerung bereits vorhandener Sequenzen, oder Erzeugung sehr kurzer Sequenzen. Beide Fälle sind nur in geringem Maß störend. Potentielle Störquellen sind jedoch eine kleine Zahl längerer Sequenzen, die Übereinstimmungen mit bislang nicht detektierten Benutzern induzieren. Für Nutzer, für die nur ein geringer Datenbestand vorliegt, können diese falschen Sequenzen durchaus zu falschen Freundschaftsvorschlägen führen. Bei Nutzern mit einem großen Datenbestand sollten solche Sequenzen hingegen nur zu geringen Störungen führen.

Dennoch ist zu beachten, dass gerade für das Finden von Sequenzen in Gebieten mit hoher Frequentierung, wie z.B. Innenstädten mit niedrigen Detailgraden, eine entsprechend hohe Fehlerrate zu erwarten ist.

6.5 Evaluation: Ortsbewertungen

Im folgenden betrachten wir die Beeinflussung der Ortsbewertungen als zweite wichtige Stütze neben den Sequenzen für einen Ansatz zur Freundschaftsbewertung. Für die folgenden Untersuchungen wird eine Karte mit Detailgrad $d = 24$ entsprechenden Größe als Grundlage gelegt. Auf dieser Karte werden die in Abschnitt 2.2 definierten Kriterien zur Ortsbewertung berechnet, wobei bei Quad Keys mit $d = 24$ jedes Pixel eine individuelle Bewertung erfährt. Beobachtet wird nun, inwieweit sich Veränderungen ergeben, wenn niedrigere Detailgrade verwendet werden.

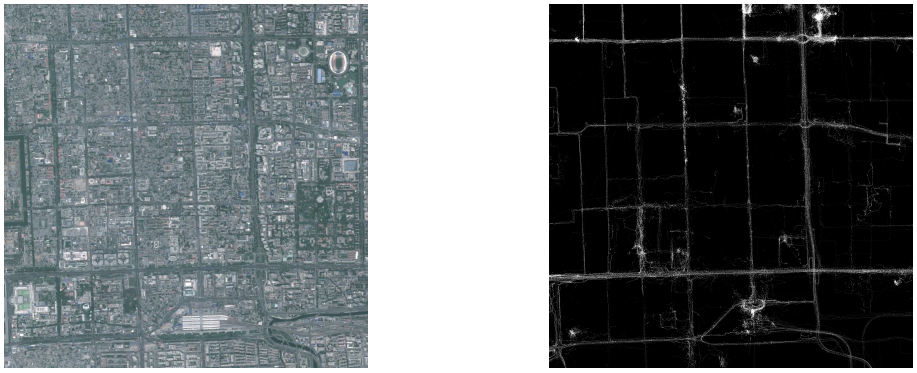
6.5.1 Fehlerbewertung und Visualisierung

Zur Bewertung der Ergebnisse werden jeweils die, für die Karte exakten, Ergebnisse mit Detailgrad 24 mit den erhaltenen Ergebnissen verglichen. Untersucht werden die Werte Freq(x, y), UserCount(x, y) und Entropy(x, y). Die Angabe $p(x, y)$ gibt dabei den Wert der untersuchten Eigenschaft an Stelle (x, y) der gegebenen Karte an. Die gesamte Karte hat eine Größe von $X \cdot Y$ Pixeln.

Zur Darstellung der Fehler verwendet wird die/der:

- *Durchschnittl. Differenz* $d_{avg} = \frac{1}{X \cdot Y} \sum_x \sum_y \mid p_{obs}(x, y) - p_{24}(x, y) \mid$
- *Standardabweichung* $d_{std} = \frac{1}{X \cdot Y} \sum_x \sum_y \left(p_{obs}(x, y) - p_{24}(x, y)\right)^2$
- *Maximalwert* über alle Pixel der Karte $p_{max} = max_{x,y}\, p(x, y)$
- *Durchschnittswert* über alle Pixel der Karte $p_{avg} = \frac{1}{X \cdot Y} \sum_x \sum_y p(x, y)$

Eine Bewertung des gesamten Datensatzes ist mit verfügbaren Mitteln nicht möglich. Daher wurde für die Evaluation ein Bereich von 2400x2400 Pixeln ausgewählt, was einer Fläche von $5.736^2 = 32.90\text{km}^2$ entspricht. Ausgewählt wurde ein bekannter Einkaufsbezirk in der Innenstadt.



(a) Satellitenaufnahme **(b)** Datenvisualisierung

Abbildung 6.5: Gegenüberstellung der Datenvisualiserung mit einer Satellitenaufnahme

Eine Visualisierung der resultierenden Ergebnisse wird erreicht, indem zu jedem Pixel der Karte die, auf die Grauwertskala skalierten, Werte dargestellt werden. Weiß steht für hohe Werte, Schwarz für niedrige. Zum Vergleich werden immer drei Visualisierungen nebeneinander dargestellt. Diese sind alle auf den gleichen Maximalwert skaliert, um einen visuellen Vergleich der Werte anhand der Grautöne zu ermöglichen. Abbildung 6.5 zeigt ein Beispiel, das die Frequency darstellt. Um das Prinzip besser erkennen zu lassen, wird in diesem Beispiel ausnahmsweise nur auf Schwarz/Weiß skaliert, wobei alle Pixel mit Frequency > 0 weiß werden. Gut zu erkennen ist die Nachbildung des Straßenverlaufs durch die Benutzerpositionen.

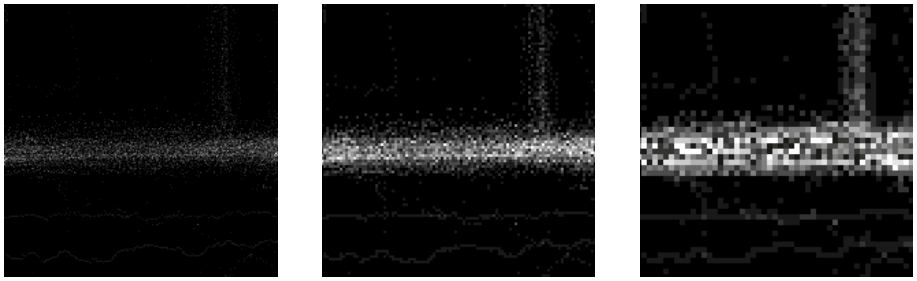
6.5.2 Auswirkung auf die Frequency

Wir betrachten zuerst die Auswirkungen auf die Bestimmung der Frequency:

Detailgrad	p_{max}	p_{avg}	d_{avg}	d_{std}
24	1744	0.822	-	-
23	3598	3.285	2.463	168.072
22	6030	13.137	12.316	3722.246

Tabelle 6.3: Auswirkung unterschiedlicher Detailgrade auf die Frequency. Detailgrad 24 wird als Referenz verwendet.

Wie zu erwarten, nehmen die Werte erkennbar zu, und erzeugen so einen zunehmend hohen Fehler im Vergleich zum Referenzergebnis. Die Visualisierung 6.6 zeigt, wie die Auflösung bei niedrigeren Detailgraden sichtbar schlechter wird. Während ein Key k_{24} mit Detailgrad 24 ein Pixel der Karte genau ansprechen kann, zeigt ein Key k_{23} mit Detailgrad 23 auf einen Bereich mit insgesamt 4 Pixeln, wobei jenes eine von k_{24} darunter ist. Entsprechend wird der Wert von 4 Pixeln gleichzeitig erhöht, wodurch die Werte allgemein höher liegen müssen als



(a) Detailgrad 24 **(b)** Detailgrad 23 **(c)** Detailgrad 22

Abbildung 6.6: Visualisierung der Veränderung der Ortsbewertungen bei unterschiedlichem Detailgrad. Hier dargestellt anhand des User-Counts eines kleinen Bereichs (200x200 Pixel $\sim$ 478x478 Meter)

bei Keys mit Detailgrad 24 (siehe auch Maximalwerte). Je niedriger die Stufe, desto drastischer wird dieser Effekt. Bei Detailgrad 22 werden schon 16 Pixel, bei Detailgrad 21 64 Pixel angesprochen.

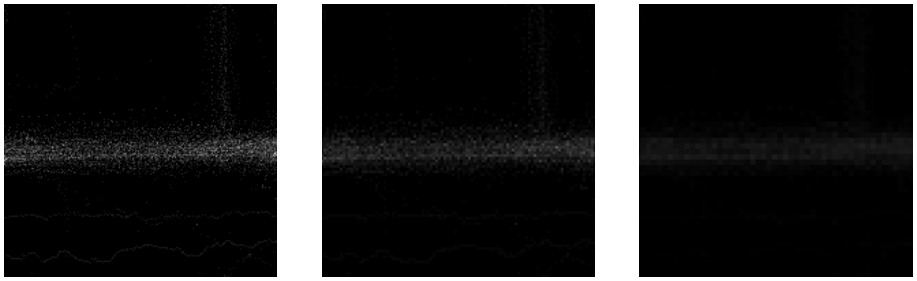
Verbesserung durch Skalierung der Werte

Um diesen Effekt entgegenzusteuern, kann eine Skalierung der Werte nach Detailgrad der Keys vorgenommen werden. Ein Key k_{23} mit Detailgrad 23 spricht 4 Pixel an, und würde den Frequency-Wert dieser Pixel um 1 erhöhen. Die Wahrscheinlichkeit, dass tatsächlich ein bestimmter dieser 4 Pixel gemeint war, beträgt $\frac{1}{4} = 25\%$. Um dies zu repräsentieren wird der Wert aller vier Pixel nicht um 1, sondern nur um $\frac{1}{4}$ erhöht. Bei Keys mit $d = 22$ entsprechend um den Wert $\frac{1}{16}$. Die Ergebnisse nach dieser Anpassung sind in Tabelle 6.4 dargestellt. Die erzielten Fehlerraten stellen dabei eine sichtbare Verbesserung dar; die Werte der Pixel liegen deutlich näher an denen der Referenzwerte.

Detailgrad	p_{max}	p_{avg}	d_{avg}	d_{std}
24	1744	0.822	-	-
23	899.5	0.697	0.253	2.947
22	376.875	0.664	0.302	4.726

Tabelle 6.4: Auswirkung unterschiedlicher Detailgrade auf die Frequency mit Skalierung. Detailgrad 24 wird als Referenz verwendet.

Es sei angemerkt, dass für eine weitere Anwendung die vorliegenden Ergebnisse normiert werden müssen (z.b. auf [0:1]), um sie geeignet verwenden zu können. Die Normierung erfolgt dann anhand des Maximalwerts, dem z.B. der Wert 1 zugewiesen wird. Problematisch bei den modifizierten Werten ist nun, dass der Maximalwert ebenfalls durch die Anpassung in gleichem Maß modifiziert wurde wie alle anderen Pixelwerte. Wird diese Karte nun anhand des Maximalwerts auf [0:1] skaliert, wird die Anpassung rückgängig gemacht. Die Werte wurden an die Referenzwerte angepasst, daher muss die Skalierung mit dem Referenz-Maximalwert erfolgen. Dieser ist per se nicht bekannt, und



(a) Detailgrad 24 **(b)** Detailgrad 23 **(c)** Detailgrad 22

Abbildung 6.7: Visualisierung der Veränderung der skalierten Ortsbewertungen bei unterschiedlichem Detailgrad. Hier dargestellt anhand des UserCounts eines kleinen Bereichs (200x200 Pixel $\sim$ 478x478 Meter)

muss so durch Heuristiken bestimmt werden. Dies ist insbesondere notwendig, wenn Keys mit unterschiedlicher Genauigkeit gleichzeitig verwendet werden, was durchaus einem realen Szenario entsprochen würde.

Eine Visualisierung, die mittels des Maximalwertes von $d = 24$ normiert wird, zeigt dies (Abbildung 6.7). Es ist erkennbar, dass diese einfache Skalierung nicht in der Lage ist die feine Auflösung höherer Detailgrade nachzubilden. Die großen Flächen werden lediglich an die Werte der Referenz angepasst. Ebenso lässt sich an den sichtlich niedrigeren Maximalwerten erkennen, dass die Skalierung Spitzenwerte stark glättet. Die hervortretenden hellen Flächen in (a) verschwinden weitestgehend.

Die naheliegende Vermutung, dass die positiven Resultate durch Anpassung der Punkte auf den weitgehend schwarzen Hintergrund erzeugt werden, lässt sich entgegnen indem nicht alle Pixel, sondern nur die Pixel untersucht werden, die im Referenzbild einen Wert > 0 hatten (also nicht schwarz dargestellt werden). Verwendet man nur diese Pixel, so stellt sich der positive Effekt dennoch dar.

6.5.3 Auswirkung auf den User Count

Die Ergebnisse der Auswertung des UserCounts sind sehr ähnlich der der Frequency, weshalb wir auf eine erneut tiefere Betrachtung verzichten. Die zuvor dargestellten Erkenntnisse lassen sich hier direkt übertragen.

Detailgrad	p_{max}	p_{avg}	d_{avg}	d_{std}
24	30	0.448	-	-
23	45	1.196	0.748	3.771
22	72	2.720	2.73	26.328

Tabelle 6.5: Auswirkung unterschiedlicher Detailgrade auf den UserCount. Detailgrad 24 wird als Referenz verwendet.

Auch hier lässt sich mit dem Ansatz der Skalierung anhand der Detailtiefe Verbesserungen erzielen:

Detailgrad	p_{max}	p_{avg}	d_{avg}	d_{std}
24	30	0.448	-	-
23	11.25	0.177	0.147	0.489
22	4.5	0.051	0.211	1.019

Tabelle 6.6: Auswirkung unterschiedlicher Detailgrade auf den User-Count mit Skalierung. Detailgrad 24 wird als Referenz verwendet.

6.5.4 Auswirkung auf die Location Entropy

Wie zuvor beobachten wir auch für die Location Entropy die Entwicklung der Ergebnisse:

Detailgrad	p_{max}	p_{avg}	d_{avg}	d_{std}
24	1.391	0.001	-	-
23	1.549	0.011	$2.979 \cdot 10^{-4}$	$2.979 \cdot 10^{-4}$
22	1.619	0.043	0.010	0.010

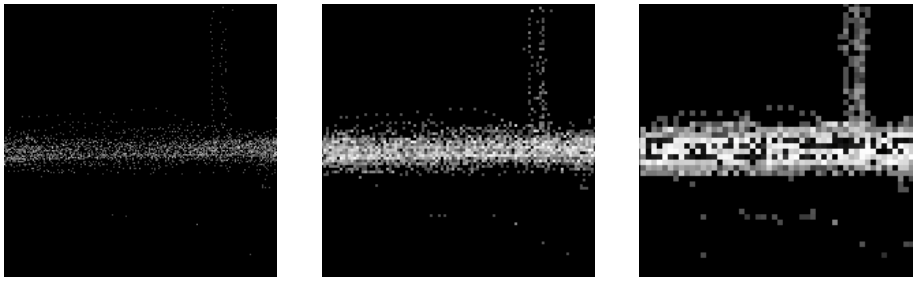
Tabelle 6.7: Auswirkung unterschiedlicher Detailgrade auf die Location Entropy. Detailgrad 24 wird als Referenz verwendet.

Auf den ersten Blick zeigen diese Werte trotz niedrigen Detailgrades wenig Veränderungen. Um ein Gefühl für die Location Entropy zu bekommen geben wir einige Beispiele:

Beschreibung	Anzahl Benutzer	Besuche	Entropy
Stark frequentiert, viele Benutzer	100	jeweils 10	2
Stark frequentiert, wenig Benutzer	5	jeweils 20	0.70
Stark frequentiert, wenig Benutzer	3	jeweils 20	0.45
Mittelmäßig frequentiert, viele Besucher	50	jeweils 10	1.70
Mittelmäßig frequentiert, wenig Besucher	10	jeweils 10	1.00
Schwach frequentiert, viele Benutzer	100	jeweils 2	0.45
Schwach frequentiert, wenig Benutzer	3	jeweils 2	0.477

Tabelle 6.8: Beispiele verschieder Szenarien für die Location Entropy.

Da die Formel maßgeblich von zwei Variablen abhängt, ist es schwierig die Ergebnisse mit den Eingaben direkt in Verbindung zu bringen. Dennoch zeigen die Auswertungen in Tabelle 6.7 nur geringe Schwankungen, was am Maßstab obiger Beispiele durchaus positiv gewertet werden kann. Stellt man die Entwicklung in einer Visualisierung (Abbildung 6.8) dar, zeigt sich, dass sich zwar die Auflösung erwartungsgemäß verringert, die Werte in ihren Größenordnungen jedoch weitestgehend erhalten bleiben. Hohe Anstiege der Maximalwerte sind weniger zu erkennen, als in Darstellungen der vorherigen Untersuchungen.



(a) Detailgrad 24 **(b)** Detailgrad 23 **(c)** Detailgrad 22

Abbildung 6.8: Visualisierung der Veränderung der Location Entropy bei unterschiedlichem Detailgrad. Hier dargestellt anhand eines kleinen Bereichs (200x200 Pixel $\sim$ 478x478 Meter)

Eine Skalierung zur Verbesserung der Location Entropy wie bei den anderen Metriken ist direkt nicht möglich. Verwendet man die gleichen Skalierungsfaktoren wie zuvor, so kürzt sich der Faktor in der Formel sofort wieder raus.

6.5.5 Zwischenergebnis: Ortsbewertungen

Es zeigt sich damit zusammenfassend, dass die Auswirkungen der Positionsgenauigkeit einen viel stärkeren Einfluss auf die Ortsbewertungen haben, als auf die Identifikation von Sequenzen. Dies ist nicht verwunderlich, da die Ortsbewertung direkt an die Detailgrade der zugrunde liegenden Karte gebunden ist. Einfache Verbesserungsverfahren können zwar die allgemeinen Fehlerrate verbessern, drücken jedoch auch Spitzenwerte sehr stark auf ein allgemeines Niveau herunter. Während die Ergebnisse der Frequency und des UserCounts direkt interpretierbar sind, ist die Location Entropy nicht so direkt zugänglich. Hier scheinen sich die Werte nur geringfügig zu ändern. Im nächsten Abschnitt zeigt Tabelle 6.9, dass sich die Änderungen tatsächlich in einem vertretbaren Maß halten.

6.6 Evaluation: Einfaches Bewertungsverfahren

Nach Untersuchung der einzelnen Komponenten soll abschließend die Veränderung an einem einfachen Algorithmus zur Bestimmung von Freundschaften beobachtet werden, der jene Komponenten zusammenführt.

Verwendet werden dafür zwei Algorithmen, die für jeden Nutzer u die fünf relevantesten Nutzer $m_1, \ldots, m_5$ aus dessen Matchings identifiziert. Das einfache Verfahren bestimmt das Ranking dieser Benutzer lediglich anhand der Länge aller Sequenzen. Jedem Nutzer wird dazu ein Gesamtwert zugeordnet, der aus der Summe der Länge aller Sequenzen bestimmt wird. Der relevanteste Eintrag m_1 ist also der Nutzer, der die längste Zeit mit u Übereinstimmungen gezeigt hat. Das erweiterte Verfahren bezieht die Location Entropy mit ein. Jede Sequenz wird erst

mit der zugehörigen Entropy gewichtet, und dann mit allen übrigen Sequenzen zu einem Gesamtwert aufsummiert. Die Gewichte berechnen sich durch Bildung des Mittelwertes der Entropy des Start- und Endpunktes der Sequenz. Wie beim einfachen Verfahren entscheidet der resultierende Gesamtwert über die Reihenfolge der Einträge.

Tabelle 6.9 zeigt die Ergebnisse für Detailgrad 23 und 22, mit Detailgrad 24 als Referenz. In beiden Verfahren wird ein Filter eingesetzt, der Sequenzen mit weniger als 30s Länge ausfiltert.

Detailgrad	Neue Einträge	Entfallende Einträge	Vertauschungen	Einträge gesamt
23 (einfach)	15	3	12	200
22 (einfach)	40	6	18	222
23 (erweitert)	15	3	13	200
22 (erweitert)	40	6	27	222

Tabelle 6.9: Auswirkung niedriger Detailgrade auf ein einfaches Bewertungsverfahren. Als Referenz wird Detailgrad 24 verwendet, welches insgesamt 188 Einträge besitzt.

Es zeigt sich, dass Veränderungen tatsächlich nur minimal auftreten (siehe auch die durchschnittliche Anzahl übereinstimmender Nutzer pro Nutzer u in Abschnitt 6.4.4). Insbesondere die Anzahl neu hinzukommender Einträge (bei Nutzern, die zuvor weniger als 5 Einträge hatten) steigt jedoch auffallend an. Dies ist vermutlich auf die allgemeine Verlängerung bestehender Sequenzen zurückzuführen, die in Abschnitt 6.4.2 untersucht wurde. Durch diese Verlängerung kommen bei sinkendem Detailgrad zunehmend neue Sequenzen über die Hürde des 30s Filters, teilweise dann mit neuen Nutzern.

6.7 Zusammenfassung

Zusammenfassend erbrachte die Evaluation das Ergebnis, dass Quad Keys für den gegebenen Anwendungskontext prinzipiell geeignet sind. Veränderungen durch unschärfere Positionen schlagen sich in erster Linie durch Erzeugung kurzer Sequenzen, und Verlängerung bestehender nieder. Die Auswertung mit einem einfachen Bewertungsverfahren hat gezeigt, dass diese Einflüsse nur bedingt zu ernsthaften Unstimmigkeiten führen.

Stärkerer Einfluss wird jedoch auf die Ortsbewertungen ausgeübt, die von der Unschärfe mehr betroffen sind. Bei niedrigeren Positionsgenauigkeiten wachsen die Regionen mit unterschiedlichen Werten zwangsweise immer weiter zusammen.

Einfache Verbesserungsverfahren können die Veränderungen durch unscharfe Positionen wie neue, kurze Sequenzen teilweise ausbessern. Die hier vorgestellten Verbesserungsmaßnahmen wie Filterung (Sequenzen) und Skalierung (Ortsbewertungen) verstehen sich jedoch lediglich als Ansätze, die aufzeigen sollen, dass Verbesserungen prinzipiell möglich sind. Es wurde entsprechend auch aufgezeigt, dass die

Verfahren teilweise nicht optimale Ergebnisse liefern. Gerade die Ortsbewertungen konnten nur mit gewissen Einbußen verbessert werden.

6.7.1 Anmerkung zur Datengrundlage

Es sei angemerkt, dass der vorliegende Datensatz nur Daten von 178 Benutzern enthält. Diese sind zwar auf einen recht kleinen Ballungsraum konzentriert, in, für soziale Netzwerke realen Größenordnungen können die Ergebnisse jedoch variieren. Es kann angenommen werden, dass sich die resultierenden Störungen sich in einem Szenario mit sehr viel mehr Benutzern entsprechend vergrößern.

Kapitel 7

Zusammenfassung und zukünftige Arbeiten

7.1 Zusammenfassung der Ergebnisse

In dieser Arbeit wurde aufgezeigt, inwieweit Positionsverläufe im Kontext sozialer Netzwerke verwendet werden können. Die Möglichkeiten für eine Analyse solcher Positionsdaten sind weitreichend, jedoch in den meisten Fällen auch mit tiefen Eingriffen in die Privatsphäre der Nutzer verbunden.

Um das Problem des Eingriffs in die Privatsphäre abzuschwächen, wurden unscharfe Positionen vorgeschlagen, insbesondere eine Darstellung von Positionen mit Quad Keys. Diese erfüllen die gesetzten Anforderungen, dass Übereinstimmungen zwischen Positionsdaten identifizierbar sind, d.h. Entfernungen zwischen Positionen berechenbar sein müssen, Positionen in der gewählten unscharfen Darstellung jedoch nur ungenau auf die Ursprungspunkte zurückrechenbar sein sollen. Dies führt insbesondere auch zu der zentralen Forderung der Distanzerhaltung, d.h. dass die berechneten Distanzen relativ mit Entfernungen in der echten Welt übereinstimmen sollen. Die, mit den unscharfen Positionen durchgeführten Berechnungen, wie Übereinstimmungen der Positionen oder Ortsbewertungen, sollten zudem möglichst wenig von den Berechnungen mit exakten Positionen abweichen. Zuletzt wurde eine Möglichkeit gefordert, die Genauigkeit der Positionen dynamisch anpassen zu können, um den Benutzer die Festlegung der Genauigkeit seiner Angaben zu ermöglichen.

In den Untersuchungen zeigte sich, dass die unscharfen Positionen mit Quad Keys zwar bei der Suche nach Übereinstimmungen zwischen Benutzerpositionen eine erwartete Fehlerrate zeigten, diese jedoch in kleinen Größenordnungen liegen. Die Anzahl identifizierter Übereinstimmungen stieg mit abfallender Genauigkeit der Positionsangaben erkennbar an, auf die Bildung von zusammenhängenden Sequenzen

zeigten sich allerdings nur Veränderungen, die nicht allzu empfindlich die Arbeit eines Algorithmus zur Freundschaftsbewertung stören würden.

In erster Line machten sich Änderungen, im Vergleich zu höheren Genauigkeiten, hauptsächlich durch die Verlängerung bestehender Sequenzen, oder Erzeugung neuer, kurzer Sequenzen bemerkbar. Im Bezug auf Letztere wurde gezeigt, dass einfache Filtermethoden bereits deutliche Verbesserungen der Ergebnisse erzielen können. Ernsthafte Störquellen sind eine kleine Anzahl längerer Sequenzen, die Übereinstimmungen mit sonstig nicht einbezogenen Nutzern induzieren.

Änderungen der Genauigkeit haben auf die Bewertung von Orten allerdings einen viel stärkeren Einfluss als auf die Identifikation von Sequenzen. Einfache Verbesserungen können die Fehlerrate verbessern, drücken jedoch auch Spitzenwerte sehr auf ein allgemeines Niveau herab.

Abschließend kann gesagt werden, dass eine Nutzung solcher unscharfen Positionsdaten in diesem Kontext somit zumindest prinzipiell möglich ist.

7.2 Zukünftige Arbeiten

An mehreren Stellen könnten weitere Arbeiten angeschlossen werden. Zum einem wären Untersuchungen mit größeren Datensätzen mit noch viel dichteren Nutzerdaten interessant, wie sie in einem realen Szenario auftreten würden.

Zum anderen besteht die Aussicht, dass weitere Verbesserungen der unscharfen Ergebnisse möglich sind. Nicht-lineare Skalierung der Ortsbewertungen und ähnliche Techniken, die die ursprüngliche Form der Werte besser bewahren, wären insbesondere nützlich.

Anhang A

Anhang

A.1 Evaluation: Vergleich Längen-/Breitengrade zu Quad Keys

A.1.1 Anzahl Matchings/Sequenzen

Die hier dargestellten Daten beziehen sich auf Abschnitt 6.3.2.

Tabelle 6.1 zeigt die Tabelle für Detailgrad 24. Zusätzlich dazu liegen die Ergebnisse für Detailgrad 23 vor:

Variante	Anzahl Matchings	Anzahl Sequenzen
Nachbarschafts-Verfahren	5.915.883	27480
Lat/Long mit $r = \frac{1}{2}\bar{d}_{23}$	4.344.625	29156
Lat/Long mit $r = \frac{3}{4}\bar{d}_{23}$	5.418.791	28395
Lat/Long mit $r = 1\bar{d}_{23}$	6.399.912	27166
Lat/Long mit $r = \frac{3}{2}\bar{d}_{23}$	7.954.579	25989

Tabelle A.1: Anzahl Matchings von NM-23 im Vergleich zu verschiedenen Lat/Long Radien

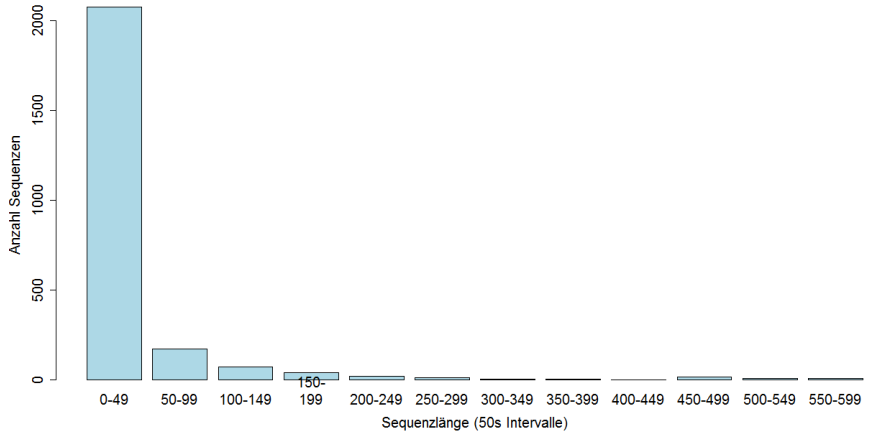
A.1.2 Untersuchung neuer Sequenzen

Die hier dargestellten Daten beziehen sich auf Abschnitt 6.3.3.

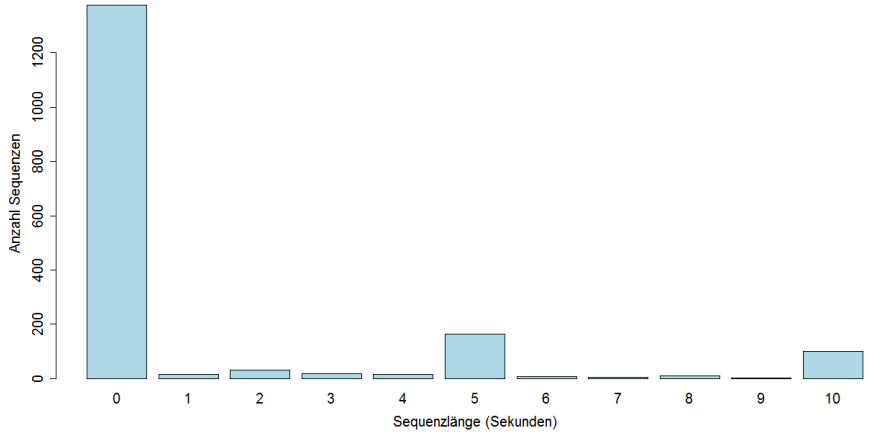
Abbildung 6.1 zeigt Diagramme, die auf den relevanten Teil gekürzt wurden. Hier liegen die vollständigen Versionen vor:

A.2 Evaluation: Finden von Sequenzen

A.2.1 Anzahl Matchings bei unterschiedlichem Detailgrad



(a) Längenverteilung (0-350s)



(b) Längenverteilung (0-6s)

Abbildung A.1: Längenverteilung der neuen Sequenzen von Lat/Long $r = \frac{3}{4}\bar{d}_{24}$ relativ zu NM-24.

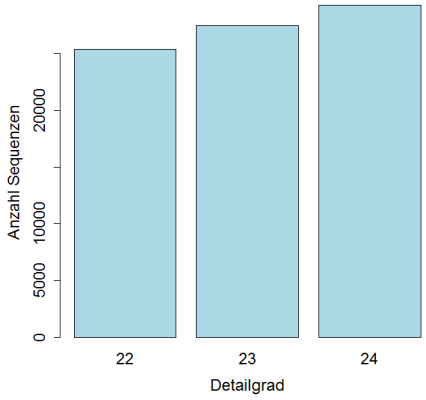
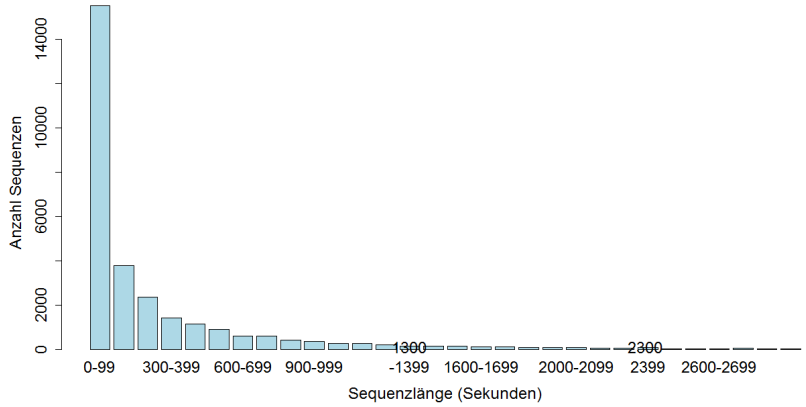


Abbildung A.2: Entwicklung der Anzahl Matchings bei unterschiedlichem Detailgrad

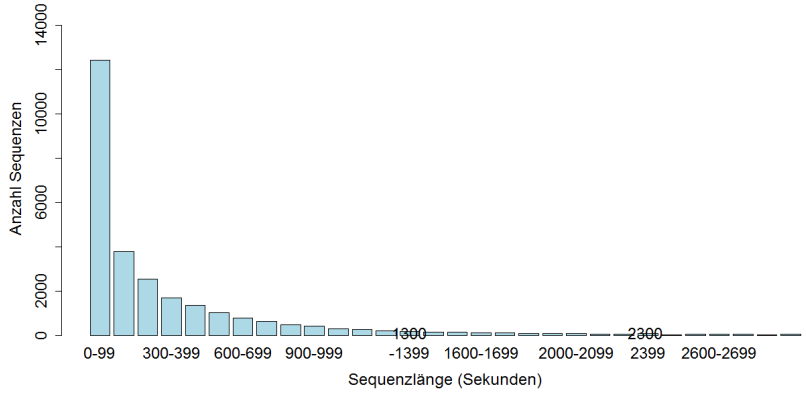
A.2.2 Auswirkung auf die Länge der Sequenzen

Die hier dargestellten Daten beziehen sich auf Abschnitt 6.4.2.

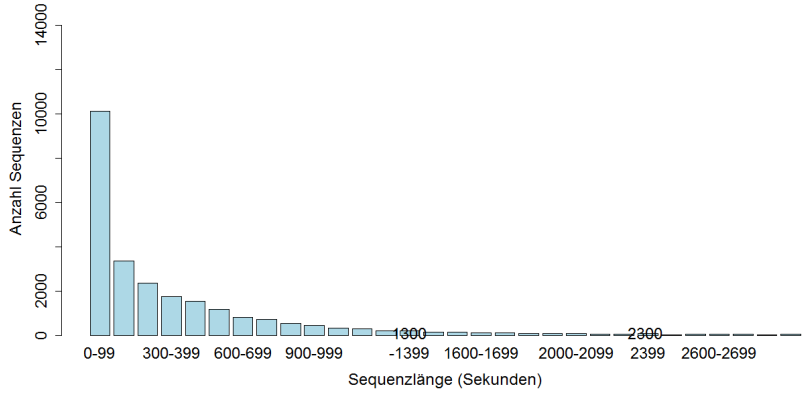
Vollständige Verteilung der Sequenzlängen verschiedener Detailgrade:



(a) Längenverteilung NM-24



(b) Längenverteilung NM-23



(c) Längenverteilung NM-22

Abbildung A.3: Verteilung der Sequenzlängen von NM-24, NM-23, und NM-22

Relevanter Ausschnitt von NM-23:

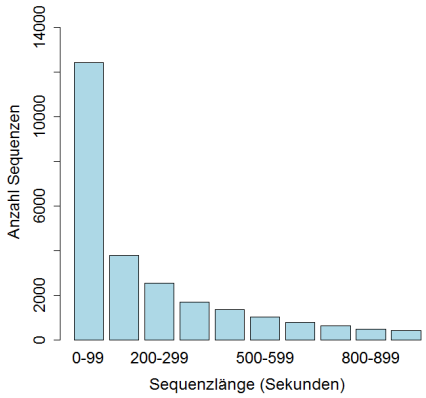
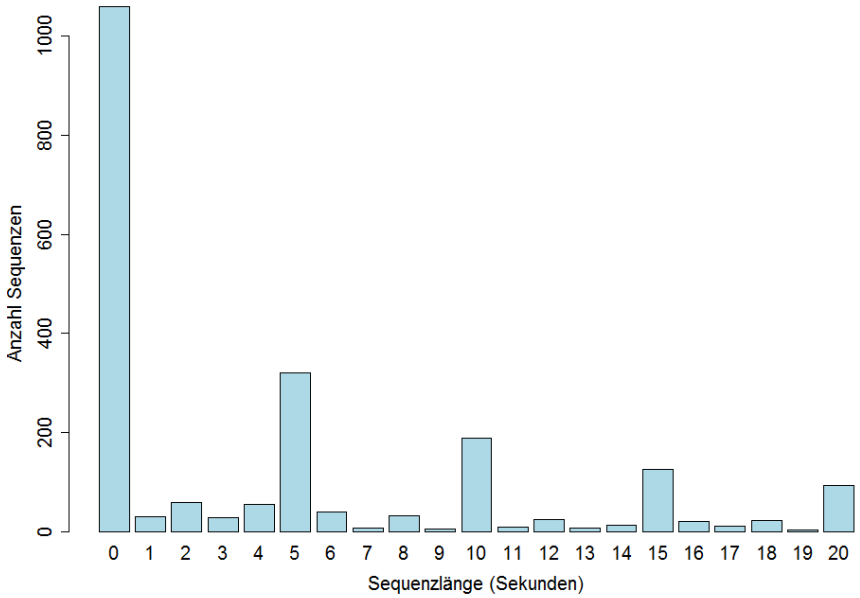


Abbildung A.4: Verteilung der Sequenzlängen von NM-23 (Ausschnitt)

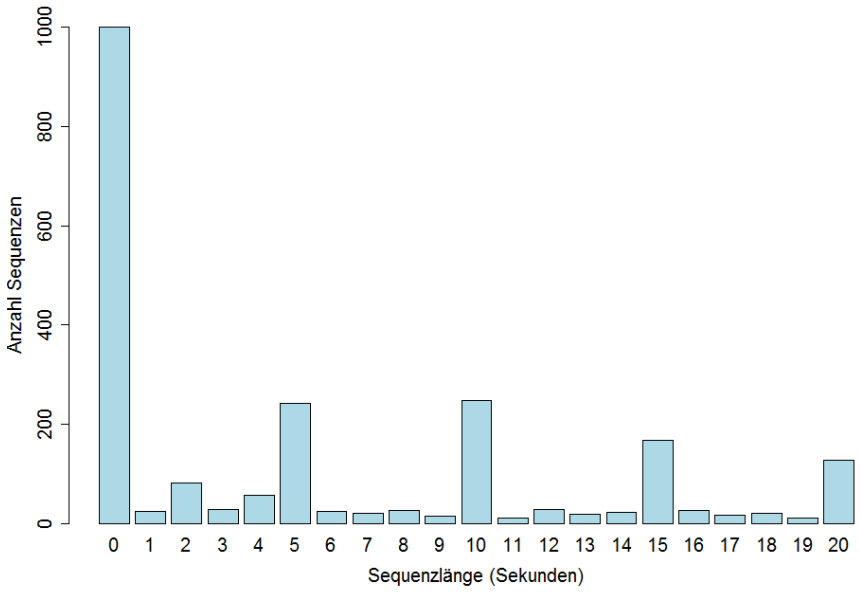
A.2.3 Auswirkung auf die Anzahl neuer Sequenzen

Die hier dargestellten Daten beziehen sich auf Abschnitt 6.4.3.

Abbildung 6.4 (b) zeigt die Längenverteilung von neuen Sequenzen. Zusätzlich dazu hier noch die feinere Aufteilung in 1s Intervalle.



(a) Längenverteilung NM-23



(b) Längenverteilung NM-24

Abbildung A.5: Längenverteilung der neuen Sequenzen von Lat/Long $r = \frac{3}{4}\bar{d}_{24}$ relativ zu NM-24.

Literaturverzeichnis

[AEM+07] A. Amir, A. Efrat, J. Myllymaki, L. Palaniappan, K. Wampler. Buddy tracking - efficient proximity detection among mobile friends. *Pervasive Mob. Comput.* 3(5):489–511, October 2007.

[BDGW07] M. Benkert, B. Djordjevic, J. Gudmundsson, T. Wolle. Finding Popular Places. In Tokuyama (ed.), *Algorithms and Computation*. Lecture Notes in Computer Science 4835, pp. 776–787. Springer Berlin / Heidelberg, 2007.

[Bha01] P. Bhattacharya. Efficient Neighbor Finding Algorithms in Quadtree and Octree. Master's thesis, Department of Computer Science & Engineering, Indian Institute of Technology, Kanpur, 2001.

[BSM10] L. Backstrom, E. Sun, C. Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th international conference on World wide web*. WWW '10, pp. 61–70. ACM, New York, NY, USA, 2010. doi:10.1145/1772690.1772698 http://doi.acm.org/10.1145/1772690.1772698

[CBC+10] D. J. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, J. Kleinberg. Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences* 107(52):22436–22441, 2010. doi:10.1073/pnas.1006155107 http://www.pnas.org/content/107/52/22436.abstract

[CML11] E. Cho, S. A. Myers, J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '11, pp. 1082–1090. ACM, New York, NY, USA, 2011. doi:10.1145/2020408.2020579 http://doi.acm.org/10.1145/2020408.2020579

[com12] comScore Inc. comScore Introduces Mobile Metrix 2.0, Revealing that Social Media Brands Experience Heavy Engagement on Smartphones. 05 2012. http://www.comscore.com / Press_Events / Press_Releases / 2012 / 5 / Introducing_Mobile_Metrix_2_Insight_into_Mobile_Behavior /

[CTH+10] J. Cranshaw, E. Toch, J. Hong, A. Kittur, N. Sadeh. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*. Ubicomp '10, pp. 119–128. ACM, New York, NY, USA, 2010. doi:10.1145/1864349.1864380 http://doi.acm.org/10.1145/1864349.1864380

[DJR12] R. Dey, Z. Jelveh, K. Ross. Facebook users have become much more private: A large-scale study. *Pervasive Computing and Communications Workshops, IEEE International Conference on* 0:346–352, 2012. doi:doi.ieeecomputersociety.org/10.1109/PerComW.2012.6197508

[Fac12] Facebook Inc. Mobile - Facebook Developers. 04 2012. http://developers.facebook.com/docs/guides/mobile

[FB74] R. A. Finkel, J. L. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta Informatica* 4:1–9, 1974. 10.1007/BF00288933. http://http://dx.doi.org/10.1007/BF00288933

[FP02] S. F. Frisken, R. N. Perry. Simple and efficient traversal methods for quadtrees and octrees. *Journal of Graphics Tools* 4:2002, 2002.

[GBC+06] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. D. O. Anderson, A. S. Morse, Y. R. Yang. Localization in sparse networks using sweeps. In *Proceedings of the 12th annual international conference on Mobile computing and networking*. MobiCom '06, pp. 110–121. ACM, New York, NY, USA, 2006. doi:10.1145/1161089.1161103 http://doi.acm.org/10.1145/1161089.1161103

[Goo12a] Google Inc. Google Places API. 08 2012. http://developers.google.com/places/

[Goo12b] Google Inc. Map Types - Google Maps JavaScript API v3. 09 2012. http://developers.google.com/maps/documentation/javascript/maptypes#MapCoordinates

[IM98] P. Indyk, R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. STOC '98, pp. 604–613. ACM, New York, NY, USA, 1998. doi:10.1145/276698.276876 http://http://doi.acm.org/10.1145/276698.276876

[Ind01] P. Indyk. Algorithms for Nearest Neighbor Search. Lecture at DIMACS Summer School Tutorial on New Frontiers in Data Mining, Rutgers University, Piscataway, NJ, 8 2001.

[LC09] N. Li, G. Chen. Multi-layered friendship modeling for location-based Mobile Social Networks. In *Mobile and Ubiquitous Systems: Networking Services, MobiQuitous, 2009. Mo-*

biQuitous '09. 6th Annual International. Pp. 1 –10. july 2009. doi:10.4108/ICST.MOBIQUITOUS2009.6828

[Mic12a] Microsoft Corp. Bing Maps Tile System. `http://msdn.microsoft.com/en-us/library/bb259689.aspx`, 05 2012. http:// msdn.microsoft.com / en-us / library / bb259689.aspx

[Mic12b] Microsoft Research. GeoLife GPS Trajectories. `http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/`, 04 2012.

[NMBM] S. Narasimhan, R. peter Mundani, H. joachim Bungartz, T. U. München. An Octree- and A Graph-Based Approach to Support Location Aware Navigation Services.

[SC08] M. Slaney, M. Casey. Locality-Sensitive Hashing for Finding Nearest Neighbors [Lecture Notes]. *Signal Processing Magazine, IEEE* 25(2):128 –131, march 2008. doi:10.1109/MSP.2007.914237

[Sin84] R. W. Sinnott. Virtues of the Haversine. *Sky and Telescope* 68(2):159+, 1984.

[SR10] C. W. Stewart, R. van der Ree. A Voronoi diagram based population model for social species of wildlife. *Ecological Modelling* 221(12):1554 – 1568, 2010. doi:10.1016/ j.ecolmodel.2010.03.019 http:// http: / / www.sciencedirect.com / science / article / pii / S030438001000164X

[ZLC+08] Y. Zheng, Q. Li, Y. Chen, X. Xie, W.-Y. Ma. Understanding mobility based on GPS data. In *Proceedings of the 10th international conference on Ubiquitous computing*. UbiComp '08, pp. 312–321. ACM, New York, NY, USA, 2008. doi:10.1145/1409635.1409677 http://http://doi.acm.org/10.1145/1409635.1409677

[ZXM] Y. Zheng, X. Xie, W. ying Ma. GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. Invited paper, in IEEE Data Engineering Bulletin. 33, 2, 2010, pp. 32-40.

[ZZM+11] Y. Zheng, L. Zhang, Z. Ma, X. Xie, W.-Y. Ma. Recommending friends and locations based on individual location history. *ACM Trans. Web* 5(1):5:1–5:44, February 2011. doi:10.1145/1921591.1921596 http://doi.acm.org/10.1145/1921591.1921596

[ZZXM09] Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th international conference on World wide web*. WWW '09, pp. 791–800. ACM, New York,

NY, USA, 2009. doi:10.1145/1526709.1526816 http://http://doi.acm.org/10.1145/1526709.1526816

Quellcode auf der CD vorhanden