

TECHNISCHE UNIVERSITÄT DRESDEN

FAKULTÄT INFORMATIK

INSTITUT FÜR SOFTWARE- UND MULTIMEDIATECHNIK

PROFESSUR FÜR MULTIMEDIA-TECHNOLOGIE

PROF. RAIMUND DACHSELT

Bachelor-Arbeit

zur Erlangung des akademischen Grades
Bachelor of Science

Nutzung mobiler Endgeräte zur Analyse von dynamischen Graphen im Raum vor einer Displaywall

Elizaveta Ragozina

(Geboren am 10. September 1993 in Novouralsk)

Hochschullehrer: Prof. Raimund Dachsel

Betreuer: Ulrike Kister, Konstantin Klamka

Dresden, 31. Januar 2017

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tag dem Prüfungsausschuss der Fakultät Informatik eingereichte Bachelor-Arbeit zum Thema:

Nutzung mobiler Endgeräte zur Analyse von dynamischen Graphen im Raum vor einer Displaywall

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Dresden, den 31. Januar 2017

Elizaveta Ragozina

Kurzfassung

Eine Graph-Visualisierung von Daten findet Einsatz in verschiedenen Domänen und erleichtert die Gewinnung von Erkenntnissen über Relationen. Viele Datensätze haben eine dynamische Natur und verändern sich über die Zeit. Dynamische Graphen kodieren die temporalen Änderungen der Objekte und ihrer Relationen. Der Vorteil der hochauflösenden Displaywand besteht darin, dass durch einen vergrößerten Darstellungsraum ein guter Überblick über die strukturellen Zusammenhänge entsteht. Der Einsatz der heutzutage sehr verbreiteten mobilen Endgeräte bietet eine lokale nutzerspezifische Darstellung für die Exploration und Manipulation der Graphen einschließlich ihrer dynamischen Komponente.

Während sich viele Arbeiten auf die Analyse statischer Strukturen fokussieren, wird das Problem der Exploration und Visualisierung dynamischer Graphen vergleichsweise wenig adressiert. In der vorliegenden Arbeit werden die grundlegenden Aufgaben für die dynamischen Graphen sowie der Stand der Forschung über den interaktiven Raum der kombinierten Anwendung Displaywand und Mobilgerät untersucht. Auf dieser Basis werden die Interaktionskonzepte für die Exploration dynamischer Graphen und die Visualisierung der Änderungen erarbeitet. Der interaktive Raum bietet dabei die räumlichen Freiheitsgrade für die Gestaltung der Interaktionen. Die entstandene Konzeption dient abschließend als Grundlage für die prototypische Umsetzung in einem existierenden Projekt zur Exploration von Graphen mithilfe mobiler Endgeräte.

Abstract

Graph visualization for data is used in various domains. It facilitates the extraction of knowledges about the relationships. Many datasets have a dynamic nature and change over time. Dynamic graphs encode the temporal changes of objects and their relations. The advantage of the high-resolution displaywall is providing a good overview of the structural relations caused by an enlarged representation space. The use of nowadays widespread mobile devices provides a local user-specific view for the exploration of the graphs including their dynamic component.

While many studies focus on the analysis of static data structures, the issue of exploration and visualization of dynamic graphs is addressed by only a few works. This thesis investigates the basic tasks for dynamic graphs and state of research into the interaction space of combined applications of displaywall and mobile device. On this basis the interaction concepts for the exploration of dynamic graphs and the visualization of the changes are developed. The interaction space provides the spatial degrees of freedom for the design of interactions. The resulting set of concepts is the basis for the prototypical implementation in the existing project for the exploration of graphs using the mobile devices.

Inhaltsverzeichnis

1	Einleitung	3
1.1	Motivation	3
1.2	Zielsetzung	5
1.3	Aufbau der Arbeit	5
2	Grundlagen und verwandte Arbeiten	7
2.1	Graphen	7
2.1.1	Graphentheorie	7
2.1.2	Graph-Visualisierung	8
2.1.3	Graph-Datenbanken	9
2.1.4	Nutzeraufgaben an Graphen	9
2.2	Grundlagen der Interaktion	11
2.2.1	Interaktion	11
2.2.2	Allgemeine Interaktionstechniken	12
2.2.3	Tangible Magic Lenses	13
2.3	Große Displays und Mobilgeräte für die Graphanalyse	14
2.3.1	Große Displays und Mobilgeräte	14
2.3.2	Multitouch- und räumliche Interaktionen	15
2.3.3	Analyse von Graphen	19
2.3.4	Visualisierung temporaler Daten	20
2.3.5	Zusammenfassung	21
3	Konzeption	23
3.1	Abbildung der zeitlichen Dimension im Interaktionssystem	23
3.1.1	Bestandteile des Interaktionssystems	23
3.1.2	Zeitliche Dimension	24
3.1.3	Der interaktive Raum	27
3.1.4	Anwendungsszenarien	29
3.2	Visualisierung der Datenänderungen	29
3.2.1	Visuelle Kodierung in einem Graph-Zustand	30
3.2.2	Bezugsintervall der Änderungen	33
3.2.3	Temporale Verfolgung der aktiven Knotenmenge	33
3.2.4	Visualisierung der Änderungen in einem Zeit-Intervall	35
3.3	Einfrierung eines Teilgraphen-Zustandes	36
3.3.1	Freiheitsgrade der Einfrierung	37
3.3.2	Einordnung im interaktiven Raum	37
3.4	Multiple Ansichten	38

4	Umsetzung	41
4.1	System-Überblick	41
4.1.1	Verwendete Hardware	41
4.1.2	Tracking	41
4.1.3	MobileLens	42
4.2	Implementierung	43
4.2.1	Datenstruktur	44
4.2.2	Dynamischer Graph im interaktiven Raum	46
4.2.3	Visualisierung der Änderungen zwischen den Intervallen	47
4.2.4	Timeline-Ansicht	50
5	Zusammenfassung, Bewertung und Ausblick	51
	Literaturverzeichnis	55

1 Einleitung

Folgendes Kapitel beschreibt die Motivation für neue Interaktionstechniken im Umgang mit den sich zeitabhängig ändernden Daten, die als dynamische Graphen visualisiert werden können. Darauf basierend werden die Ziele dieser Arbeit formuliert und der strukturelle Aufbau beschrieben.

1.1 Motivation

Graphen sind eine in der Informationsvisualisierung verbreitete Datenvisualisierung für die Abbildung der Relationen zwischen Objekten in einer Datenmenge. Die Anwendungen sind in der Informatik, vielen anderen Wissenschaften und im täglichen Leben zu finden. Komplexe Datenmengen sind in textueller oder tabellarischer Form schwer überschaubar. Eine graphische Struktur gibt einen Überblick, erleichtert die Gewinnung von Erkenntnissen über die Daten und hilft, die Zusammenhänge zu erkennen. In der Mathematik sind Graphen ein gut erforschtes Gebiet mit vielen existierenden Algorithmen, die die Effizienz der Datenverarbeitung in Anwendungsfällen steigern.

Beispielsweise werden die miteinander durch das Internet verbundenen Computernetzwerke durch die Graph-Struktur dargestellt. Viele der typischen Aufgaben in Rechnernetzen wie Routing lassen sich durch die Graphen simulieren. Unser Zeitalter ist von verschiedenen sozialen Netzwerken wie Facebook, Twitter oder Instagram geprägt, die einen Teil des Online-Lebens von vielen Menschen repräsentieren. LinkedIn soll beispielsweise bei der Arbeitnehmer- oder der Arbeitgebersuche behilflich sein. Die Graph-Darstellung erweist sich als praktisch für die soziologische Analyse der Beziehungen zwischen Personen oder Personengruppen. In der Chemie und Biologie werden verschiedene Netzwerke traditionell durch die Graph-Modelle mit verschiedenen Eigenschaften dargestellt: Gen-Netzwerke durch gerichtete zyklische Graphen, Protein-Interaktionsnetzwerke durch ungerichtete Graphen, molekulare Strukturen durch ungerichtete zyklische Graphen, metabolische Pfade durch gerichtete zyklische Graphen [EB06] [BSRG06]. Diese Darstellungsart hilft bei den Aufbauforschungen und bei der Beobachtung der in diesen Netzwerken ablaufenden Prozesse. Zu anderen Anwendungsbereichen gehören Taxonomien von Arten, Gegenständen, Konzepten, verschiedene Hierarchien, beispielsweise Dateihierarchien, Metro-Pläne, semantische Netze usw. [HMM00].

Viele Graphen bleiben nicht statisch, sondern verändern sich dynamisch über die Zeit. So kann ein dynamischer Graph eine chemische Reaktion darstellen, wobei die einzelnen Graph-Zustände Schritte dieser Reaktion sind. In sozialen Netzwerken kommen Relationen hinzu, während andere verschwinden. Die Evaluation der dynamischen Graphen ist eine Abbildung der sich ständig ändernden Umwelt und ist für die Erkennung von Trends oder die Aufstellung von Prognosen sehr wertvoll.

Die Analyse von Daten, speziell auch Graph-Daten, erfolgt im Allgemeinen durch zwei Vorgänge: Exploration und Konfirmation [Beh97]. Exploration ist die Suche nach charakteristischen Eigenschaften des Netzwerks, um hierüber Hypothesen aufstellen zu können. Bei der Konfirmation sucht man gezielt nach Eigenschaften des Netzwerkes, um Hypothesen zu verifizieren oder zu widerlegen. Gra-

phische Methoden der Datenvisualisierung ermöglichen es dem Nutzer, die Regelmäßigkeiten und Abweichungen schnell zu finden, sie sind somit gut für explorative Zwecke geeignet. Algorithmische Methoden erlauben die Konfirmation durch gezielte Abfragen. Der Fokus dieser Arbeit liegt auf den explorativen Aufgaben für Graphen.

Bei einer Graph-Visualisierung ergibt sich eine Reihe von Problemen, die die Exploration beeinträchtigen können [HMM00]. Das Schlüsselproblem hierbei ist die Graphgröße. Wie visualisiert man große Graphen, damit alle Elemente sichtbar sind und die Viewability und die Usability erhalten bleiben? Zu den anderen Herausforderungen zählt Graphlayout, welches effizient und sinnvoll für bestimmte Daten sein soll. Eine Vielzahl von Graphlayouts beschäftigt sich mit bestimmten Wunschkriterien wie Planarität bzw. Minimierung der Überschneidung von Kanten und verschiedenen ästhetischen Vorgaben. Für die bequeme Echtzeitinteraktion mit dem Graphen ist es wichtig, die Zeitkomplexität des Zeichnungsalgorithmus zu beachten. Das Layout soll außerdem berechenbar sein, so dass verschiedene Abläufe des Algorithmus nicht extrem verschiedene visuelle Darstellungen liefern und damit die kognitive Wahrnehmung des Nutzers stören.

Für die Visualisierung von größeren Graphen können größere Oberflächen verwendet werden. Interaktive Oberflächen moderner Displaywände bieten ein großes Spektrum der Interaktionsmöglichkeiten für die Exploration und Manipulation der Daten. Zusätzliche Ausstattung des Raums vor der Displaywand mit den Sensoren für die Wahrnehmung von Bewegungen erlaubt räumliche Interaktionen mit getrackten Objekten, beispielsweise Menschenkörpern oder zusätzlichen Geräten. Die große Oberfläche des Displays erhöht den Wert der maximalen Darstellbarkeitsgröße im Vergleich zu gängigen PC-Displays. Somit kann das Problem der Anzeige großer Graphen vermindert werden. Die Möglichkeit der Nutzer-Kollaboration ist ein weiterer Vorteil der Displaywand. Durch einen ausreichend großen physischen Raum vor der Display-Oberfläche können mehrere Nutzer an einer Datenmenge arbeiten.

Bei gegebenen Vorteilen existieren auch einige Probleme bei der Arbeit mit den Daten an einer Displaywand. Die maximale Größe der angezeigten Daten wird zwar erhöht, jedoch bleiben die Probleme mit Usability wegen der fundamentalen perceptualen und physischen Eingrenzungen von Menschen bestehen, wozu beispielsweise der relativ kleine Fokusbereich des menschlichen Auges zählt [War12]. Rädle et al. [RJMR14] haben durch eine Studie herausgefunden, dass größere mit Informationen gefüllte Display-Bereiche die Lern- und Navigationsgeschwindigkeit im Vergleich zu typischen Mobilgerät-Größen erhöhen. Allerdings verringert sich dieser Vorteil mit wachsender Größe des Display-Bereichs.

Der Einsatz von mobilen Endgeräten für die Visualisierungen ermöglicht eine Nutzeridentifizierung und somit die Gestaltung eines nutzerspezifischen Interface. Smartphones und Tablets sind heutzutage weit verbreitet und stellen somit eine vertraute Interaktion für den Nutzer bereit. Gleichzeitig ist die Display-Oberfläche der mobilen Geräte zu gering für die Darstellung großer Datenmengen.

Während für die Exploration der statischen Graphdaten bereits viele Ansätze entwickelt wurden, erfordert die Darstellung einer dynamischen Graphstruktur die Konzepte für die zusätzliche Abbildung und Interaktionen mit der Zeit-Variable. Diese Arbeit versucht diese Herausforderung durch den Einsatz der kombinierten Anwendung von Displaywand und Mobilgerät zu bewältigen, wobei die Vorteile beider für die Analyse der Graphdaten maximiert und die Nachteile minimiert werden sollen. Die Displaywand kann die Rolle des Kontexts annehmen und dient bei der Mehrnutzerarbeit als öffentlicher Bereich. Die Navigation in Graphdaten erfolgt dadurch eher physisch als virtuell

[BN05]. Virtuelle Daten sind abstrakt, eine Erweiterung um physikalische Greifbarkeit fördert den Einsatz des räumlichen Gedächtnis. Das Mobilgerät kann in der Kombination mit der Displaywand das nutzerspezifische Interface mit der personalisierten Sicht auf die dynamischen Daten bereitstellen. Die aus der Kombination resultierenden Freiheitsgrade bei der Gestaltung der Touch- und räumlichen Interaktionen stellen ein Potential für die Verwendung bei der Exploration der Zeit-Variable dar.

1.2 Zielsetzung

Allgemeines Ziel vorliegender Arbeit ist die Entwicklung von Konzepten für die Exploration dynamischer Graphen. Die Konzepte sollen die Freiheitsgrade einer kombinierten Verwendung von Displaywand mit Mobilgeräten integrieren. Das mobile Endgerät soll dabei die Rolle einer Tangible Magic Lens spielen, es ist ein Werkzeug mit persönlicher, manipulierbarer Fokussicht. Für die Exploration sollen sinnvolle Abbildungen gefunden werden, die die räumlichen Bewegungen der Linse in der realen Welt auf bestimmte Interaktionen im Informationsraum des Graphen übertragen. Es sind Interaktionstechniken zu erarbeiten, die die räumlichen Freiheitsgrade sowohl im absoluten Bezug zu der Displaywand als auch im körperrelativen Bezug zu dem Nutzer für explorative Zwecke nutzen.

Der Schwerpunkt dieser Arbeit liegt auf der Exploration der zeitlichen Komponente des Graphen. Es soll dem Nutzer ermöglicht werden, in der zeitlichen Dimension zu navigieren und einzelne Zeitzustände des Graphen zu selektieren. Zudem soll Vergleichen als die Basismethode für die Gewinnung der Erkenntnisse über die Eigenschaften der zeitlichen Entwicklung des Graphen ermöglicht werden. Dazu sollen geeignete Visualisierungsformen für die Änderungen der visuellen Variablen entwickelt werden, die typische Knoten- und Kanteneigenschaften repräsentieren.

Im praktischen Teil dieser Arbeit sollen ausgewählte Teile der Konzeption prototypisch umgesetzt werden. Damit soll die Basis der Konzeption für die Exploration dynamischer Graphen veranschaulicht werden.

1.3 Aufbau der Arbeit

Diese Bachelorarbeit gliedert sich in die Kapitel: Grundlagen, Konzepte, Umsetzung sowie Bewertung, Zusammenfassung und Ausblick.

- **Grundlagen:** In Kapitel 2 werden alle zum Verständnis der Arbeit benötigten theoretischen Grundlagen behandelt. Außerdem werden einige grundlegende Konzepte der Informationsvisualisierung genannt und bestehende verwandte Arbeiten in relevanten Bereichen große Displays und Mobilgeräte, Multitouch- und räumliche Interaktionen, Analyse von Graphen und Visualisierung temporaler Daten aufgeführt.
- **Konzeption:** Kapitel 3 präsentiert eine ausführliche Beschreibung der grundlegenden Konzepte zur temporalen Exploration der dynamischen Graphen und Gestaltung der Interaktionen zwischen dem Nutzer, dem Mobilgerät und der Displaywand.
- **Umsetzung:** In Kapitel 4 wird die bereits existierende Anwendung *Mobile Lenses* beschrieben, und einige ausgewählte Implementierungsdetails der im praktischen Teil der Bachelorarbeit umgesetzten Teile der Konzepte werden genauer behandelt.

- **Zusammenfassung, Bewertung und Ausblick:** Kapitel 5 fasst die Ergebnisse dieser Arbeit zusammen und diskutiert die erarbeiteten Konzepte und die vorgenommene Implementierung. Zudem wird ein Ausblick auf die mögliche Weiterentwicklung gegeben.

2 Grundlagen und verwandte Arbeiten

Dieses Kapitel führt in die für das Verständnis dieser Arbeit nötigen Grundlagen der Forschungsbereiche Graphen und User Interface ein. Es wird außerdem der Stand der Forschung in den Bereichen der großen interaktiven Displays und der kombinierten Interaktion mit Mobilgeräten präsentiert. Schließlich werden einige existierende Ansätze der kombinierten Interaktion für die Exploration und Manipulation von Graphen und speziell der dynamischen Graphen beleuchtet.

2.1 Graphen

In diesem Abschnitt werden die wichtigen Definitionen in den Bereichen Graphentheorie, Graph-Visualisierung, Datenhaltung und Nutzeranfragen an Graphen gegeben.

2.1.1 Graphentheorie

Graphen sind neben relationalen Datenbanken eine Möglichkeit der Darstellung von relationalen Daten. Graphstrukturen haben sich als naturgetreu in unserer Welt verschiedener Netzwerke herausgestellt. Graphentheorie ist ein Teilgebiet der diskreten Mathematik, das sich mit der Untersuchung von Graphen befasst.

Im Folgenden werden die gekürzten Definitionen von Graphen nach [Die00] und von dynamischen Graphen nach [Sem16] adaptiert. Ein *Graph* G besteht aus zwei disjunkten Mengen: der Menge der Knoten V und der Menge der Kanten E und wird formal als $G = (V, E)$ mit $E \subseteq V^2$ notiert.

Graphen haben vielfältige charakteristische Eigenschaften wie Richtung, Zyklen, Zusammenhang, Planarität usw. Ein Graph kann zusammenhängend oder nicht zusammenhängend sein. Zusammenhang bedeutet, dass im Graphen zwischen je zwei Knoten ein Weg existiert. Ein Graph heißt gerichtet, wenn jede seiner Kanten einen Anfangs- und einen Endknoten hat. Ein zyklischer Graph enthält Kreise. Planare Graphen können ohne Kantenüberschneidungen gezeichnet werden. Bezüglich der Anzahl an Verbindungen kann ein Graph dünn- oder dicht besetzt sein. Bei dicht besetzten Graphen liegt die Anzahl der Kanten nah an die maximale. Die Vielfalt der Eigenschaften macht die Graphen zu einer für Anwendungen aus vielen verschiedenen Bereichen passenden Datenstruktur.

Der Begriff *Dynamischer Graph* erweitert die Repräsentation statischer Daten um temporale Informationen. Die Änderungen der dynamischen Daten können das Hinzufügen oder das Löschen von Knoten/Kanten sowie die Änderungen ihrer Eigenschaften sein. Der dynamische Graph G besteht aus einem geordneten Set von Graphen bzw. Snapshots G_0, \dots, G_{n-1} und wird formal als $G = \{G_0, \dots, G_{n-1}\}$ notiert. Zum Zeitpunkt t_i mit $0 \leq i \leq n-1$ repräsentiert der Graph G_i zugrundeliegende Daten.

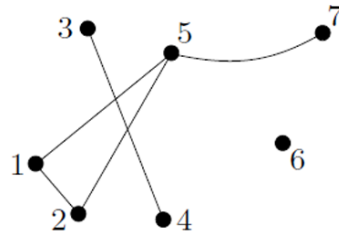


Abbildung 2.1: Der Graph auf $V = \{1, \dots, 7\}$ mit der Kantenmenge $E = \{\{1, 2\}, \{1, 5\}, \{2, 5\}, \{3, 4\}, \{5, 7\}\}$ [Die00].

2.1.2 Graph-Visualisierung

Die Graph-Daten liegen zunächst in textueller Form vor, beispielsweise in einer Datenbank. Die Erzeugung einer geeigneten Visualisierung aus der Datenbank ist die Aufgabe des Visualisierungswerkzeuges. Es existiert eine große Anzahl von verschiedenen Graph-Visualisierungstechniken. Bei der verbreiteten *Node-Link* Visualisierung [PD10b] zeichnet man die Knoten als Objekte (z.B. Kreise, Glyphen) und die Kanten als Linien zwischen diesen Punkten. Ein Beispiel des Node-Link-Diagramms ist in Abbildung 2.1 zu sehen.

In Studien zur Formerkennung ist bereits bewiesen, dass die Bilderkennung wesentlich schneller als die Worterkennung ist [Bra99]. Somit spielt die Kodierung der Eigenschaften von Daten, in dieser Arbeit speziell von Graph-Daten, durch bestimmte visuelle Variablen eine wichtige Rolle für die Wahrnehmung der Visualisierung. Nur eine begrenzte Anzahl visueller Eigenschaften wird präattentiv wahrgenommen, das bedeutet innerhalb von maximal 200-250 ms. Zu den Kriterien für die Auswahl visueller Variablen gehören Effektivität und Ausdrucksfähigkeit. Die Variable muss schnell interpretierbar sein und Unterschiede für die Kodierung mehrerer Zustände aufweisen. Außerdem sollen möglichst keine falschen Botschaften übertragen werden [CMS99]. Zu den gängigen visuellen Variablen zählen Position, Länge, Größe, Helligkeit, Farbe, Textur, Ausrichtung, Form. Einige visuelle Variablen besitzen eine intrinsische Bedeutung: Vergrößerung, Verlängerung symbolisieren Vermehrung, Wachstum. Objekte mit der Position „links“ oder „oben“ werden durch die Leserichtung als die zuerst Erscheinenden wahrgenommen [Hea08]. Die falsche Verwendung solcher Variablen kann für den Nutzer irreführend sein.

Die zu kodierenden Eigenschaften kann man in qualitative und quantitative Datentypen unterteilen. Qualitative oder kategoriale Eigenschaften sind Eigenschaften, die in Worten oder Zahlen beschrieben werden können und diskrete Werte annehmen. Die Menge aller Werte bzw. Kategorien ist abzählbar. Bei nominalen quantitativen Eigenschaften handelt es sich um ungeordnete Mengen, beispielsweise Essenskategorien. Bei ordinalen qualitativen Daten ist die Menge dagegen geordnet. Quantitative Eigenschaften können gezählt werden (beispielsweise 1337, 42) [Hea08]. Mackinlay [Mac86] hat eine Rating für die relative Effektivität der visuellen Variablen für die Datentypen vorgeschlagen. Für die quantitativen Datentypen eignen sich demnach am besten Position, Länge, Winkel, Steigung. Nicht geeignet sind Textur oder Form. Effizient für ordinale Datentypen sind Position, Dichte, Farbsättigung und Farbton, ineffizient sind hingegen sind Form, Volumen und Fläche. Für nominale Datentypen eignen sich Position, Farbton, Textur und Verbindung.

Ware [War04] beschreibt in seinem Buch eine visuelle Grammatik von Diagrammelementen, die speziell für die Visualisierung von Node-Link-Diagrammen verwendet werden können (Abbildung














Grafische Merkmale	Bedeutung und Verwendung	
Geschlossene Konturen		Entität, Objekt, Knoten
Form geschlossener Regionen		Art/Typ von Entitäten
Farbe geschlossener Regionen		Art/Typ von Entitäten
Größe geschlossener Regionen		Wert von Entitäten größer = mehr
Teilende Linien in geschlossenen Regionen		Erzeugung von Teilbereichen in Entitäten, z.B. TreeMaps
Hinzugefügte Formen		Hinzugefügte Entitäten, Teil der Relation
Eingeschlossene Formen		Enthaltene Entitäten
Räumlich angeordnete Formen		Serie, Reihenfolge
Verbindende Linien, Kanten		Relationen, Beziehung zwischen Entitäten
Beschaffenheit der Verbindung		Art/Typ der Relation
Linienstärke der Verbindung		Stärke der Relation
Tab-Verbindungen		Eindeutige Verbindung zwischen Entitäten
Nachbarschaft		Gruppen von Entitäten

Abbildung 2.2: Eine visuelle Grammatik von Diagrammelementen zur Kodierung verschiedener Aspekte in NodeLink-Diagrammen [PD10b].

2.2).

2.1.3 Graph-Datenbanken

Der Bereich der Graph-Datenbanken ist vom Problem sehr große Graph-Strukturen geprägt. Es geht um die schnelle Analyse von Big Data durch graphanalytische Systeme und effizientes Behandeln der Updates. Daraus resultieren die Anforderungen an die Gestaltung der Datenstruktur. Die Ansätze für die Behandlung der dynamischen Graphen sind in die Bereiche verteilte Systeme, verteilter Speicher und out-of-memory unterteilt, sie sind in vielen Systemen wie GraphChi, X-Stream oder TurboGraph umgesetzt. Im Unterschied zum explorativen Ansatz beschränkt sich die Interaktion des Nutzers mit den Daten auf das abstrakte Stellen der Anfragen.

2.1.4 Nutzeraufgaben an Graphen

Bei der Implementierung von analytischen Tools für Graph-Daten setzt der Entwickler bestimmte Nutzeranfragen an den Graphen um. Amar et al. [AES05] schlagen 10 grundlegende Low-Level Visualisierungsaufgaben für Analyse der Daten vor:

- Werte-Anfragen
- Filtern der Daten nach bestimmten Kriterien
- Berechnen eines abgeleiteten Wertes
- Finden der Daten mit extremen Werten
- Sortieren nach bestimmten Ordinal-Metriken

- Bestimmen der Bereiche
- Charakterisieren der Verteilung für bestimmte quantitative Eigenschaften
- Finden der Anomalien in den Daten
- Finden der Cluster
- Bestimmen der Beziehungen zwischen zwei Eigenschaften

Lee et al. [LPP⁺06] vervollständigen diese Liste um den Scan, bei dem schneller Überblick über die Daten ohne genaue Werte gemacht wird und Mengen-Aufgaben. Außerdem erweitern sie die Liste um eine Graphdaten-spezifische Aufgabe: Finden der benachbarten Knoten. In ihren Studien erarbeiten sie eine Taxonomie der häufig auftretenden Aufgaben für Graph-Visualisierungen. Die Taxonomie enthält topologiebasierte, attributbasierte sowie Surf- und Überblickaufgaben, die sich in Low-Level Aufgaben zerlegen lassen.

- Die *topologiebasierten Aufgaben* beschäftigen sich mit der Adjazenz, d.h. der direkten Verbindung zwischen den Knoten, der Zugänglichkeit, d.h. der direkten oder indirekten Verbindung, allgemeinen Verbindungen und Konnektivität. Typische Beispiele der Aufgaben sind Nachbarschaften und ihre Eigenschaften, Erreichbarkeitsprobleme, Pfandsuchen und Cluster-Erkennungen.
- Bei den *attributbasierten Aufgaben* geht es um Eigenschaften der Knoten und der Kanten. Die Aufgaben können durch Filter-, Bereichs-, Berechnungs- und Verteilungsaufgaben beschrieben werden.
- Die *Surfaufgaben* beinhalten Aufgaben zu der Pfad-Verfolgung und dem Wiederbesuchen den Knoten.
- Die *Überblick-Aufgaben* sind zusammengesetzte Forschungsaufgaben für das schnelle Erhalten der Schätzwerte, sie beschäftigen sich mit Mustersuchen und Ausreißern. Schnelle Schätzungen können für den Überblick interessanter sein als genaue Werte.

Es gibt zudem *High-Level Aufgaben*, die von oben genannten Aufgaben nicht abgedeckt sind. Dazu zählen Lee et al. Aufgaben, die sich mit der Analyse der Vergleiche, der Fehlererkennung in den Daten und den temporalen Aufgaben beschäftigen. Auf die temporalen Eigenschaften eines dynamischen Graphen gehen Ahn et al. [APS14] in ihrer Studie genauer ein. Sie kategorisieren die Eigenschaften in Eigenschaften von individuellen und zusammengesetzten Ereignissen, auf die sich die Low-Level Anfragen beziehen können.

- *Individuelle Ereignisse* treten zu getrennten Zeitpunkten auf. Zu ihren Eigenschaften zählen Einzel-Ereignisse wie das Hinzufügen oder das Löschen, Ersetzungereignisse, bei denen das Hinzufügen und das Löschen zum gleichen Zeitpunkt passieren, sowie die Geburt und der Tod, die Lebenszyklusgrenzen setzen.
- *Zusammengesetzte Ereignisse* bestehen aus einer geordneten Menge von individuellen Zeitpunkten mit diskreten oder kontinuierlichen Werten. Zu der Art der Veränderung zählen Gregory et al. [Gre12] Wachstum oder Verringerung, Konvergenz oder Divergenz und Stabilität der Eigenschaft. Ahn vervollständigt dies um graphspezifische Eigenschaften Musterwiederholung sowie Höchst- oder Tiefstwerte über die Zeit. Zu den Eigenschaftsanfragen der Veränderungsrate zählt Ermitteln der Geschwindigkeit und Beschleunigung der Eigenschaft. Ferner betrachtet Ahn Aufgaben, die sich aus Low-Level Aufgaben zusammensetzen lassen. Folger-

de zusammengesetzte Aufgaben beinhalten die Berechnung der temporalen Unterschiede sowie das Finden der Beziehungen zwischen diesen Änderungen. Bei den vergleichenden zusammengesetzten Aufgaben geht es um Vergleiche der temporalen Änderungen der Eigenschaften.

Während bei Ahn et al. die Aufgaben an einen dynamischen Graphen nach Ereignisart sowie Art und Rate der Veränderung kategorisiert sind, unterteilt Semertzidis historische Anfragen in drei Kategorien: historische Graphanfragen, historische Zeitanfragen und historische top-k Anfragen [Sem16]. Die Graphanfragen sind Anfragen mit typischen Taxonomieaufgaben an die Graphzustände. Bei den Zeitanfragen geht es um den Zeitpunkt oder das Zeitintervall, in dem ein Ereignis passiert ist. Die top-k Anfragen beziehen sich auf das längste Zeitintervall, in dem ein Ereignis passiert ist.

2.2 Grundlagen der Interaktion

Für das Erforschen der großen Datenmengen in kombinierter Anwendung Displaywand + Mobilgeräte wird die Gestaltung der Interaktionen sowie geeigneter User Interfaces benötigt. Dieser Abschnitt befasst sich mit allgemeinen Interaktionsaufgaben und -techniken im Bereich User Interface.

2.2.1 Interaktion

In der Informatik versteht man unter *Interaktion* im weit gefassten Sinne allgemein die Kommunikation zwischen Anwender und System [DFAB04]. Yi et al. [YKSJ07] sehen die Rolle der Interaktion für die Informationsvisualisierung in der Möglichkeit, die Repräsentation direkt und indirekt zu manipulieren und zu interpretieren. Nach Lehmann [Leh15] besteht die Interaktion eines Nutzers mit virtuellen Umgebungen aus drei wesentlichen Komponenten: Eingabemodalität, Interaktionsaufgabe und Mapping. Die Eingabemodalität der Nutzerinteraktion beschreibt, welches Eingabegerät der Nutzer verwendet, um mit dem System oder der Anwendung zu interagieren und welche zusätzlichen Sensordaten (z.B. Positionstracking) für die Nutzereingabe verwendet werden. Typischerweise werden für die Eingaben verschiedene Eingabegeräte wie Laser, Maus, aber auch menschliche Eingaben durch Gesten oder Sprache verwendet. Die Interaktionsaufgabe beschreibt, welche Aufgabe der Nutzer durchführt. Das Mapping beschreibt, wie die Interaktionsaufgabe abgebildet wird.

Schneiderman [Shn96] klassifiziert die Interaktionsaufgaben in Overview, Zoom, Filter, Details-on-demand, Relate, History und Extract.

- *Overview*: Erlangen eines Gesamtüberblicks über den Informationsraum.
- *Zoom*: Heranzoomen von interessanten Informationen.
- *Filter*: Herausfiltern von irrelevanten Daten.
- *Details-on-demand*: Selektion einzelner oder gruppierter Objekte und Anzeige der Details zu diesen.
- *Relate*: Erkennen der Beziehungen zwischen einzelnen Objekten, Wertevergleiche.
- *History*: Speicherung vorheriger Aktionen, um Änderungen rückgängig machen zu können.
- *Extract*: Extrahieren von Teilmengen und deren Parameter.

In [PD10a] formulieren Preim et al. die wichtigen Begriffen für User Interfaces. Das Gebiet der Mensch-Computer-Interaktion beschäftigt sich mit der Gestaltung von interaktiven Benutzerschnitt-

stellen, die durch Usability und User Experience charakterisiert sind. *Usability* beschreibt, wie gut der Nutzer die Funktionen des Systems benutzen kann und basiert auf einigen Kriterien wie Nützlichkeit bestimmter Interaktionstechniken, Fehlerrate, Lernaufwand, Zeiten der Erledigung einer Aufgabe, usw. *User Experience* betrachtet neben Usability auch subjektive Wahrnehmungen und grundlegende Bedürfnisse des Nutzers. *User Interface Engineering* wird als phasenorientiertes Vorgehen der Softwareentwicklung definiert. Zu den typischen Hauptphasen gehören Analyse, Design, Implementierung und Evaluierung des Systems. Dabei kann die Gesamtaufgabe in verschiedene Ebenen zerlegt werden. Neben dem Interaktionsdesign gehören dazu z.B. das Informationsdesign und das visuelle Design.

Diese Arbeit betrachtet mögliche Abbildungen der Interaktionsaufgaben auf die Graphaufgaben und fokussiert sich auf das Interaktionsdesign und das visuelle Design unter Berücksichtigung von Usability und User Experience.

2.2.2 Allgemeine Interaktionstechniken

Für die Kombination Displaywand + Mobilgerät werden Strategien benötigt, um unter Berücksichtigung der Vor- und Nachteile beider die Informationsräume navigierbar zu machen. Das Problem der Anzeige komplexer Datenmengen auf begrenzten Displays korreliert mit Usability und User Experience von User Interfaces. Man unterscheidet zwischen verschiedenen Ansätzen, die den Nutzer bei der Interaktion mit Daten unterstützen sollen. Cockburn [CKB09] unterteilt die Gestaltungstechniken nach getrennten und nicht getrennten Ansichten. Zu den Ansätzen mit nicht getrennten Ansichten gehören die *Fokus + Kontext* Technik, bei der der Fokusbereich innerhalb des Kontextes angezeigt wird. Ein Beispiel der Technik ist in Abbildung 2.3 a) dargestellt. Unterverzeichnisse, auf die fokussiert wird, werden durch FishEye vergrößert. Die räumliche Trennung der Fokus- und der Kontext-Ansicht wird beim *Overview + Detail* Ansatz benutzt, die zeitliche Trennung beim *Zooming and Panning* Ansatz.

Durch Studien wurde bewiesen, dass der Verlust des Kontextes zur Erhöhung des kognitiven Aufwands bei Anordnung der Details führt. Um dieses Problem zu lösen, wird bei dem Fokus + Kontext Ansatz die Anzeige lokaler Details mit globalem Kontext kombiniert. In der Klassifikation der Fokus + Kontext Methoden unterscheiden Kosara et al. zwischen räumlichen, dimensional und Cue

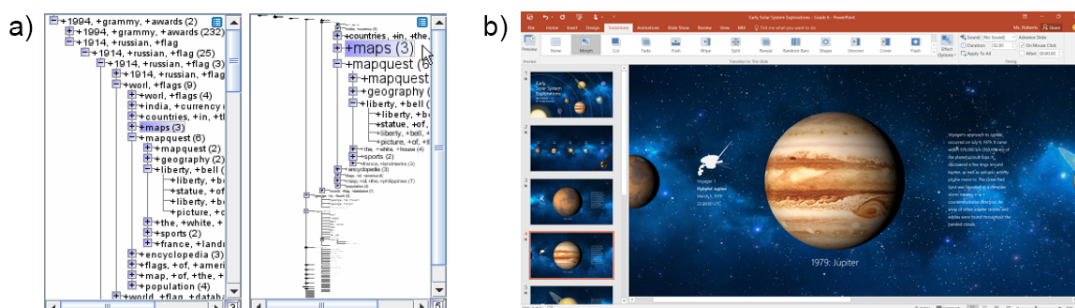


Abbildung 2.3: a) Fokus + Kontext Technik Fisheye für einen hierarchischen Baum. [TAVS06]. b) Trennung des Kontextes (links) und des Fokus bei der Overview + Detail Technik in PowerPoint.

Methoden. Bei den räumlichen Methoden wird in einer Ansicht der Fokus-Bereich durch eine Verzerrungsfunktion feiner und der Kontext gröber dargestellt. Die Verzerrung kann in bestimmten Fällen zu kognitiven Problemen bei der Wissenserkennung führen, z.B. wenn die Position in der Datenstruktur eine bestimmte Bedeutung hat. Durch die dimensionalen Methoden wird das Konzept der magischen Linse verwendet, mit der der Fokus über die Daten bewegt werden kann.

Overview + Detail Ansätze lösen das Problem durch eine Trennung der Kontext- und Fokusansichten. In einer Vielzahl von Anwendungen wie Google Maps, Adobe Reader, Powerpoint (Abbildung 2.3 b)), Sublime Text usw. werden beide Ansichten getrennt auf einem Display dargestellt. Andere Konzepte, unter anderem Tangible User Interfaces [Ish07], führen die digitale mit der physikalischen Welt zusammen und erlauben die räumliche Trennung der Ansichten. Die Interaktion mit der digitalen Information wird durch physische Manipulation von greifbaren Objekten unterstützt.

2.2.3 Tangible Magic Lenses

Die greifbaren magischen Linsen bieten ein Beispiel von Tangible User Interfaces. Tominski et al. [TGK⁺14] definieren die Linse als eine interaktiv parametrisierbare räumliche Selektion, auf die eine Visualisierung angewandt wird. Sie unterscheiden zwischen lokalen Linsen und Linsen, die das Display global beeinflussen, und konzentrieren sich auf die erste Art. Zu den grundlegenden Linseninteraktionen gehören Erstellen, Löschen und Manipulieren sowie räumliche Selektion und Linsenfunktion. Die vielfältigen Modalitäten der Selektion sind in Abbildung 2.4 a) dargestellt. Die Linsenfunktion kann parametrierbar sein und über komplexe Datenoperationen verfügen. Vorteilhaft bei den Linseninteraktionen ist das kontinuierliche visuelle Feedback an der Linsenoberfläche.

Die Linse im Allgemeinen ist ein Beispiel für die Fokus + Kontext Technik. Ein Beispiel der in den Kontext integrierten Linse ist auf der Abbildung 2.4 b) zu sehen. Die Besonderheit der greifbaren magischen Linsen liegt in der räumlichen z-Achsentrennung, die ein Eigenschaft der Technik Overview + Detail darstellt. Aufgrund dieser Eigenschaft kategorisieren Cockburn et al. [CKB09] die greifbare Linsen als Overview + Detail Technik. Dennoch ist die Linse ein beweglicher See-Though Bereich, welcher im Unterschied zu den in der 2D-Ebene geteilten Ansichten durch die räumliche Lage während der Bewegung stets an den Kontext geknüpft ist, wie in Abbildung 2.4 c) zu sehen. Dies kann als eine Integration in den Kontext interpretiert werden und ist Merkmal der Fokus + Kontext Technik. Looser et al. [LGB07] kategorisieren die greifbare Linse deswegen als Fokus + Kontext Technik. Die Linse kann die Rolle der detaillierten Ansicht spielen, während die Visualisierung auf dem Display, z.B. auf dem Tabletop oder einer Displaywall, als Kontext dient. Bei greifbaren Linsen, die über die

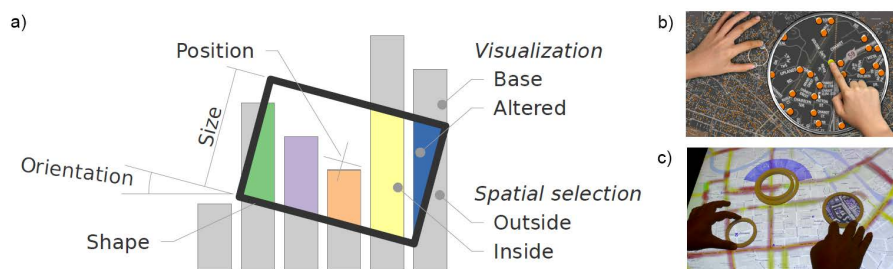


Abbildung 2.4: a) Linsenmodalitäten [TGK⁺14]. b) Beidhändige Touchinteraktion in FingerGlass [KAP11]. c) Exploration einer Karte durch TangibleRings [EWCP13].

entsprechenden Sensoren verfügen, können räumliche Manipulationen mit der Linse auf bestimmte Aufgaben abgebildet werden, z.B. Translationen auf Auswahl des Fokusbereichs.

2.3 Große Displays und Mobilgeräte für die Graphanalyse

Dieser Abschnitt bietet einen Überblick über die zurzeit existierenden Forschungen und Projekte in den für diese Arbeit relevanten Bereichen. Dazu gehören die kombinierten Techniken zum Einsatz von großen interaktiven Displays und mobilen Endgeräten, Touch-basierte und räumliche Techniken der Interaktionen, graphanalytische Konzepte sowie Techniken, in denen die temporalen Aspekte umgesetzt werden.

2.3.1 Große Displays und Mobilgeräte

Große Displays bieten ausreichend Platz für die Analyse großer Daten. Aus technologischer Sicht gehören zu den großen Displays Displays, bei denen die Anzeigefläche erhöht ist. Nach der Definition von Andrews et al. [AEYN11] sind damit die Displays, deren Displaygröße dem Einfluss des menschlichen Körpers entspricht, gemeint. Bei den hochauflösenden Displays handelt es sich um die Erhöhung der Informationsmenge. Ardito et al. [ABCD15] nehmen eine Kategorisierung der Displays unter anderem nach räumlicher Platzierung, Visualisierungstechnologie und Interaktionsmodalität vor. Räumlich platziert können die Displays als vertikale Tabletops, horizontale Wände, diagonale, Boden-Displays usw. sein. Zu den Visualisierungstechnologien gehören verschiedene Projektionen und Monitor. Zu den Hauptinteraktionsmodalitäten gehören Touch, Körper, externe Geräte. Die *Touchinteraktionen* sind ein gut untersuchtes Gebiet: Moderne Displays erlauben die Erkennung mehrerer Finger- und Händeeingaben sowie unterschiedlicher Gesten. Die *körperbasierten Ansätze* basieren auf Reaktionen des Systems auf Präsenz, Bewegungen, Augen des Nutzers. Ein Beispiel für körperbasierte Anwendungen ist distanzabhängige Interaktion von Lehmann [Leh15]. Zu den *externen Geräten*, die oft in Kombination mit den großen Displays eingesetzt werden, zählen Mäuse, Eingabestifte, Tastaturen, Mobilgeräte. Moderne Technologien wie Bluetooth und Wi-Fi bestreben eine möglichst schnelle und störungsfreie Kommunikation zwischen dem Gerät und dem Display.

Große Displays eignen sich auf natürliche Weise für die physische Navigation. Die Nutzer bewegen sich mit dem ganzen Körper, dem Kopf, den Augen entlang des Displays, positionieren sich näher, um mehr Details zu erkennen, oder weiter weg von dem Display, um einen Überblick zu erhalten. In Studien wurde untersucht, ob eine solche natürliche Navigation effizienter als die für die kleinen Displays typische virtuelle. Die Ergebnisse der Studien von Czerwinski et al. [CSR⁺03] zeigen einen positiven Einfluss auf Nutzerperformance. Zu ähnlichen Ergebnissen kommen Ball et al. [BNB07].

Moderne Mobilgeräte sind programmierbar und verfügen über eine wachsende Menge von relativ günstigen eingebauten Sensoren [LML⁺10]. Das Touch-sensitive Display erlaubt die Erkennung von Berührungen. Mit dem Beschleunigungssensor, werden die physischen Bewegungen des Geräts im Raum erkannt. Mit Gyroskop kann man bestimmen, ob die Position des Geräts vertikal oder horizontal positioniert ist. Die Sensoren erlauben Anwendungen in breiter Vielfalt von Domänen. Die Idee, Mobilgeräte als Erweiterung der Funktionalitäten der Displaywand für die Exploration der Daten zu verwenden, findet sich zahlreichen Forschungen. Häufig werden die Mobilgeräte in Anlehnung an die klassischen Eingabegeräte Maus und Tastatur als *Steuerung für das große Display* eingesetzt, die um

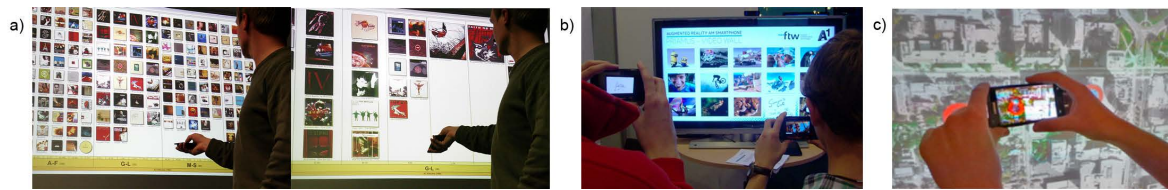


Abbildung 2.5: a) Steuerung des Musik-Browsers Mambo mit Gesten [DB09]. b) Selektion eines Videos [BLF12]. c) Das Mobilgerät als Linse bei der Kartennavigation [GPG⁺15].

weitere Funktionalitäten erweitert werden können. Vepsäläinen et al. [VDRN⁺15] beschreiben eine mobile Anwendung für die Steuerung des Spiels "Ducks and Pirates", die eine Art der Spielkonsole darstellt. Die Spielfiguren werden durch die Touch-Interaktionen mit dem speziell für das Spiel entwickelten User Interface gesteuert. Das Spiel an sich verläuft in einem Browser des separaten Displays. Ein weiterer Ansatz des Mobilgeräts als Steuerung für kombinierte gestenbasierte Interaktionen mit dem großen Display findet sich bei Dachzelt et al. [DB09]. Für Transfer und Steuerung der Daten durch das Mobilgerät wird ein Set von einfachen Werf- und Neigungsgesten entwickelt (Abbildung 2.5 a)). Hierbei können die Medienobjekte in einem Musik-Browser Mambo exploriert werden. Man kann gezielte Suchen und Filterungen durchführen. Die durch den Einsatz der Methaper gewährleistete Intuitivität der Gesten sorgt dafür, dass die Aufmerksamkeit des Nutzers vom großen Display nicht abgelenkt wird.

Verbreitet sind die sogenannten See-Through oder *Augmented Reality Ansätze*, bei denen die Informationen der Umwelt durch ein Gerät, beispielsweise eine Brille, um virtuelle Komponenten in Echtzeit erweitert und vervollständigt werden. Die greifbaren magischen Linsen überschneiden sich mit dem Augmented Reality Ansatz. Die Linse erweitert die Ansicht abhängig von der Linsenfunktion und kann über Steuerungselemente, Interface usw. verfügen. Die Mobilgeräte können die Rolle eines Tangible User Interfaces mit dem Display als räumliche magische Linse mit festgelegter Form spielen, die Interaktionen werden durch die Mobilgerätsensoren unterstützt. Da die Smartphones und Tablets im Allgemeinen nicht durchsichtig sind, benötigen sie Techniken für die Abbildung des Inhalts vor ihnen auf dem Display. Dazu zählen unter anderem die Kameranutzung und Tracking der Position des Geräts. Baldauf et al. [BLF12] beschreiben die Anwendung eines Augmented Reality Mobilgeräts für eine Videowand (Abbildung 2.5 b)). Das Gerät erlaubt die Selektion und das Abspielen von den an der Displaywand dargestellten Videos, erweitert um verschiedene Abspielmodi. Im öffentlichen Modus ist es möglich, das ausgewählte Video an der Displaywand abzuspielen, im privaten Modus erfolgt das Abspielen im Mobilgerät. Die Ergebnisse der Studie mit mehreren Nutzern haben gezeigt, dass der öffentliche Modus öfter zu Irritationen geführt hat, da unklar war, wer das Abspielen eines Videos aktiviert hat. Dies zeigt die Wichtigkeit privater Ansichten, da gegenseitige Störungen vermieden werden. Grubet et al. [GPG⁺15] beschreiben die Anwendung der Mobilgeräte als magische Linsen für die Karten-Navigation auf den öffentlichen vertikalen Displays (Abbildung 2.5 c)). Die Linse wird für das Highlighting von Interessenobjekten und für erweiterte Informationen wie z.B. ein Panorama verwendet.

2.3.2 Multitouch- und räumliche Interaktionen

Mit dem Aufkommen von verbraucherorientierten Multitouch Geräten haben sich die Multitouch Interaktionen als eine verbreitete Art der Interaktionen mit Mobile User Interfaces etabliert. Gleich-

zeitig erlauben die Sensoren der Mobilgeräte und der Umgebungen die Verwendung des Raums für die Gestaltung der Interaktionen. Viele Ansätze im Bereich der großen Displays sind gemischt und nutzen sowohl die Touch-Gesten als auch die räumlichen Interaktionen.

Yee et al. [KRD14a] unterscheiden zwischen direkten und indirekten (abstrakten) Touch-Gesten. Direkte Gesten repräsentieren direkte Manipulation von Objekten, z.B. Drag fürs Bewegen des Objekts. Indirekte Gesten sind spezieller und abstrakter, z.B. Bewegen von Gruppe von Objekten. Die Autoren fassen die fünf wichtigsten Kriterien für die Effektivität von Gesten-Interaktionen zusammen.

- Hoher Grad des Interaktionskontextes. Das System sollte dem Nutzer klar machen, dass die Gesten verwendet werden können. Diese sollen möglichst intuitiv und offensichtlich sein.
- Minimaler Aufwand. Effiziente Gesten sollten einfach sein und keine besondere Geschicklichkeit anfordern.
- Passende Metaphern. Gesten sollten eine logische Beziehung mit den resultierenden Funktionalitäten haben.
- Wiederholte Benutzung und minimaler Muskelstress. Aufgrund des Wiederholungsbedarfs vieler mit der Produktivität assoziierten Aktivitäten sollte die Wiederholung möglichst einfach sein.
- Erleichterung der Erkennung durch die Anwendung.

Eine Reihe von Gesten finden häufige Verwendung in Touch-baiserten interaktiven Systemen. Dazu gehören Finger Taps, die oft für die Selektion verwendet werden, Finger Drags für das Bewegen, Pinch, Spread Gesten für das Zoomen. Zu weiteren verbreiteten Gesten zählen Mehrfinger-Taps und -Drags sowie Rotationsdrags. Die Multitouch Interaktionen sind nicht standardisiert. Die Entwickler definieren ihre Bedeutung abhängig von der Domäne und den Aufgaben der Anwendungen.

Während die meisten verbreiteten Touch-Gesten auf den Metaphern für räumliche Manipulationen basieren, gibt es auch Ansätze, die die bildlichen Symbole ausnutzen. Dazu gehört der Sketch-Ansatz von Goncalves et al.[GF12]. Für die Exploration von Fotokollektionen wurden Gesten entwickelt, die stark vereinfachte Skizzen für Begriffe darstellen, die die Interaktionen repräsentieren. Beispielsweise wird die Geste in Form eines Vierecks für Einstellung des Hoch- oder Querformats eines Fotos verwendet. Der Einsatz von Skizzen-Gesten kann außerdem in den Fällen nützlich sein, in denen andere einfachen Gesten bereits verwendet sind, oder die zu aufrufende Funktion zu komplex ist.

Die Verwendung von Touch-Gesten kann sowohl an den großen Displays erfolgen als auch an den Mobilgeräten für die Manipulationen der Linsenfunktionalitäten und der Daten in der Linse. Allgemein erfolgen alle Touch-Gesten in zwei Dimensionen des Raums. Einige Forscher untersuchen auch die dritte Dimension auf die Interaktionsfreiheitsgrade. In seinen Arbeiten lösen sich Spindler et al. [SD09, STSD10] von den Beschränkungen einer interaktiven Oberfläche und erweitern den Interaktionsraum mit den Daten auf den physischen 3D Raum über horizontale Tabletops. Spindler et al. nehmen eine Klassifikation der Daten vor, die zu explorieren sind (Abbildung 2.6). Dazu zählen zoombare, volumetrische, temporale und geschichtete Daten. Der volumetrische Informationsraum kann als kontinuierlicher Set von Voxeln, eine lineare Abbildung der 3D Daten, beschrieben werden. Geschichtete, zoombare und temporale Informationsräume sind die Mengen von zweidimensionalen, entlang der z-Achse angeordneten Daten. Die Schichten bei den geschichteten Daten repräsentieren unikate Eigenschaften des Modells. Bei den zoombaren Daten sind die Schichten verschiedene

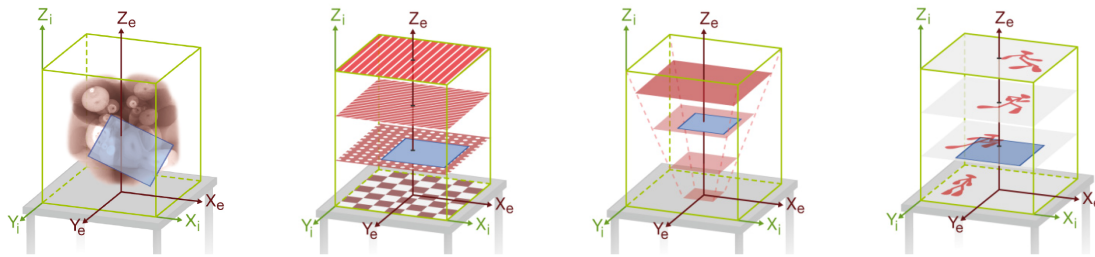


Abbildung 2.6: a) Volumetrischer Informationsraum. b) Geschichteter Informationsraum. c) Zoombarer Informationsraum. d) Temporaler Informationsraum. [SD09].

Skalierungsstufen einer Visualisierung. Man kann auch den Raum-Zeit-Würfel der temporalen Daten explorieren, der die räumlichen und zeitlichen Aspekte in einer vereinheitlichten 3D Darstellung integriert. Während räumliche Aspekte auf dem Tabletop dargestellt werden, kann die Zeitdimension auf der z-Achse abgebildet werden.

Das Konzept des lagebewussten greifbaren Displays für Interaktionen mit Informationsraum setzen Spindler et al. im PaperLens System um, welches einige räumliche Gesten sowie Stift- und Touch-Eingaben erkennen kann. Potentiell sind die Funktionalitäten des Systems auch auf Mobilgeräten umsetzbar.

Spindler et al. [STSD10] entwickelten des weiteren ein Vokabular der Freiheitsgrade für Interaktionen der Tangible Views mit 3D Raum (Abbildung 2.7). Sie beschreiben acht Basis-Nutzungsmuster: Translation, Rotation, Einfrierung, Gesten, direktes Pointing, Toolbox-Methapher sowie Multiple Views und visuelles Feedback. *Translation* ist ein Weg, im Informationsraum zu navigieren, sie kann horizontal, vertikal sein oder rotierend sein. *Einfrierung* wird benutzt, wenn keine Intention der Interaktion mit einem System besteht, z.B. dann, wenn der Nutzer eine bestimmte Ansicht näher untersuchen möchte. Die Menge von Gesten, *Flipping*, *Schütteln* und *Kippen* repräsentiert komplexere Arten der Interaktionen. *Visuelles Feedback* ist eine grundlegende Voraussetzung für die Interaktion mit einem visuellen System. Die Verwendung von diesen zusätzlichen Freiheitsgraden soll bewirken, dass der Nutzer direkter und natürlicher mit komplexen Informationsräumen interagieren kann. Sich überlappende oder nicht überlappende *Multiple Views* eignen sich für kollaborative Arbeit, wenn man beispielsweise die personelle Filter kombinieren möchte.

Im Folgenden werden einige Anwendungsszenarien von PaperLens beschrieben. Durch beliebiges Bewegen, Neigen und Halten der Linse im Informationsraum kann man die Schnitte der volumetrischen Daten, z.B. geologische, biologische oder medizinische, darstellen. Dies ist sinnvoll für die flexible Erkundung des Datensets. Bei den geclusterten Graphen wird die Bewegung entlang der z-Achse für die Ansichten in verschiedenen Abstraktionsebenen (semantisches Zoomen) verwendet. Bei den hochauflösten Bildern bietet es sich an das Tangible View als Vergrößerungsglas zu nutzen, um in beliebige Bereiche hineinzuzoomen (geometrisches Zoomen). Einen weiteren Anwendungsfall bieten geographische Karten, bei denen man die zeitliche Dimension durch Translation entlang der z-Achse explorieren kann. Um die Übersicht aller Zeitdaten zu erhalten, kann man das Display kippen, so dass der Raum-Zeit Würfel senkrecht geschnitten wird.

Exploration geschichteter Daten durch PaperLens ist ein Fall der *distanzabhängigen Nutzereingabe*. Von der Distanz des Mobilgeräts zum Display hängt die Informationsdarstellung und die Abbildung bestimmter Interaktionsarten ab. Ein weiteres Beispiel bietet der Ansatz von Lehmann et al. [Leh15]

zur Unterteilung des Raums vor der Displaywand in Zonen. Die Abstraktionsebene des Graphen an der Displaywand ändert sich dynamisch mit den Bewegungen des Nutzerkopfs. Vorteilhaft dabei ist die Natürlichkeit der Änderungen des Detailgrades. Im Unterschied zum Ansatz von Spindler mit Tangible Lens sind die Anzeigeänderungen global und werden an einen Einzelnutzer angepasst. In SpiderEyes [DHKQ14] wird die Distanz zwischen dem Nutzer und der Displaywand sowie zwischen mehreren Nutzern auf drei verschiedene Parameter abgebildet. Bei der Abbildung auf eine Visualisierungsart wird diese abhängig von der Distanz mit anderen Informationen vervollständigt (Abbildung 2.8 a). Ähnlich dazu erfolgt die Abbildung auf den Detailgrad: je weiter von der Displaywand, desto größer werden die Informationen dargestellt (Abbildung 2.8 b). Bei der Abbildung auf die Zoom-Stufe wird die Visualisierung vergrößert, wenn der Nutzer näher an der Displaywand ist. Auch bei diesem Ansatz erfolgt das Feedback des Systems auf die Interaktionen global.

Bei vielen Anwendungen werden die Touch- und die räumlichen Interaktionen kombiniert. Schmidt et al. [SSRG12] schlagen einen neuen Stil der kombinierten Interaktionen zwischen den interaktiven Displays und den Mobilgeräten, der Datentransfer, Personalisierung, Authentifizierung, mobile Menüs, privates Feedback und Eingabeoptionen unterstützt. Dabei wird der Raum vor dem Display für die direkten Manipulationen der Display-Inhalte durch Gesten mit dem Mobilgerät verwendet. In [SSG13] wird die Verwendung der Mobilgeräte als personalisierte Clipboards beschrieben. Sie werden für verschiedene Tasks wie Kopieren, Einfügen oder Manipulationen eingesetzt. Die Nutzeridentifikation durch die Clipboards erlaubt eine Nutzerkollaboration, die bei klassischen Clipboards an der öffentlichen Displayoberfläche ohne gegenseitige Störungen nicht möglich ist. Seifert et al. [SBR13] entwickelten PointerPhone, eine Anwendung, die Selektionen und Interaktionen an einem großen Display durch direktes Pointing mit dem Mobilgerät erlaubt. Die Anwendung unterstützt grundlegende Eingabeoptionen durch mehrere Alternativen: Software-Buttons auf dem mobilen Display, Touch-basierte Gesten, räumliche Interaktionen wie Rotation oder räumliche Bewegung des Geräts usw. Zusätzlich wird die Funktion des personellen Feedbacks unterstützt. Durch die Kombination von Pointing und Gesten werden weitere Funktionalitäten wie Datentransfer oder Zeichnen unterstützt. Die kollaborative Arbeit wird durch die Identifizierung jedes Geräts im System ermöglicht. Die Nutzer können personelle Ausgaben der sie interessierenden Informationen erhalten, die an der Displaywand die Gruppenaktivitäten stören könnten.

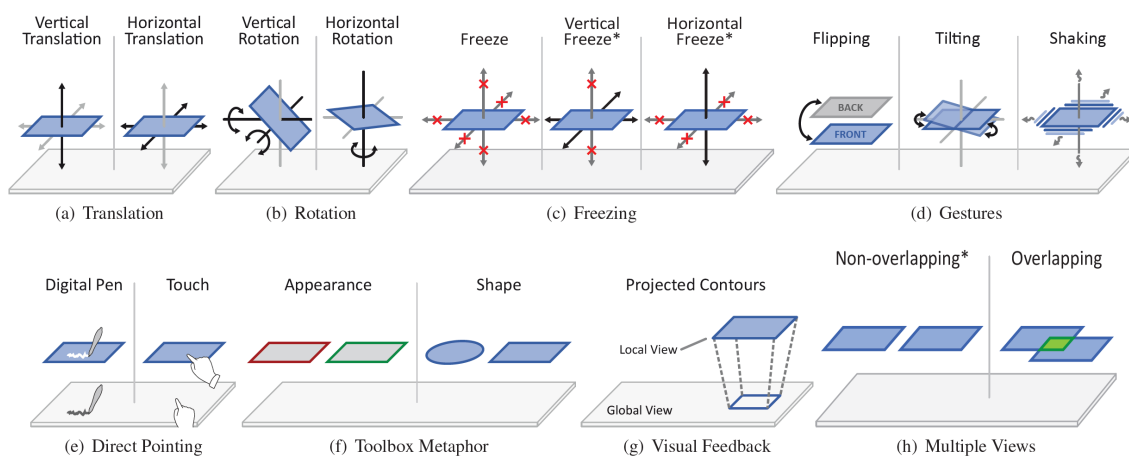


Abbildung 2.7: Überblick über das Interaktionsvokabular von Tangible Views. [STSD10].

Kurdyukova et al. [KRA12] untersuchen in einer Studie, wie sich der Einsatz von Touch vs. räumlicher Gesten für die Aufgabe Datentransfer an einem iPad zeigen. Zum Beispiel wurden für das Senden der Daten die Drag-Geste, die Werf-Geste und direkter Kontakt untersucht. Die Forscher kommen zum Ergebnis, dass im allgemeinen die Touch-Gesten von den Teilnehmern präferiert wurden. Eine interessante Beobachtung war, dass mit zunehmender Distanz zwischen dem iPad und öffentlichem Display, der Einsatz räumlicher statt Touch-Gesten anstieg. Die räumlichen Gesten werden als intuitiv stärker beschleunigend eingestuft.

2.3.3 Analyse von Graphen

Im Bereich der Graphen gibt es einige Anwendungen, die die interaktive Exploration und Manipulation durch Nutzer unterstützen sollen. Sie fokussieren sich auf verschiedene Aspekte der Graph-Analyse und haben in der Regel klare Anwendungsszenarien für Graphen bestimmter Art.

Beispielhaft für die Grundoperationen mit Graphenelementen ist ein von Frisch et al. [FHD09] entwickelter Multitouch-Gestenset. Dazu gehören das Erstellen und das Löschen von Knoten, ungerichteten und gerichteten Kanten, Selektionen, Bewegungen, Größenänderungen und Zoomen usw. Laut der Studie haben die meisten Nutzer Eine-Hand-Interaktionen als die passendsten eingestuft. Bei den Gesten dominieren die Eingaben mit einem oder zwei Fingern. Bei den Zooming-Aufgaben präferierten die Nutzer Zwei-Hände-Gesten, beispielsweise die Pinch-Geste. Auch für die Selektionen haben die Zwei-Hände-Gesten gut bei den Nutzer-Ratings abgeschnitten, da der Einsatz beider Hände die Auswahl beschleunigt. Andere Beispiele für die Verwendung von Multitouch-Gesten finden sich bei Kister et al. [KRD14b]. Sie beschäftigen sich mit der Kombination verschiedener Filter in einer Linse sowie mit speziell für die Graphen entwickelten Linsengesten, die die entsprechenden Filter aufrufen. So ruft eine spiralförmige Drag-Geste den Filter für die Anzeige der Knotennachbarn auf. Eine Fisheye-Vergrößerung wird durch eine Pinch-Geste aufgerufen, wobei die Distanz zwischen den Fingerberührungen als Parameter für Maß der Verzerrung dient. Schimdt et al. [SNDC10] haben einen Gestenset speziell für die Kantenmanipulationen entwickelt. Die Kanten können mit verschiedenen Optionen selektiert und gebündelt werden. Eine spezielle Linse soll außerdem im Auswahlbereich die Kanten, die nicht zu den Knoten in diesem Bereich gehören, daraus entfernen.

Während die Touch-Interaktionen für die manipulativen und explorativen Aufgaben an den einzelnen Graph-Elementen oder Gruppen in vielen Anwendungen eingesetzt werden, finden sich nur wenige Graph-Anwendungen, bei denen es zum Einsatz der räumlichen Interaktionen kommt. Ein Beispiel

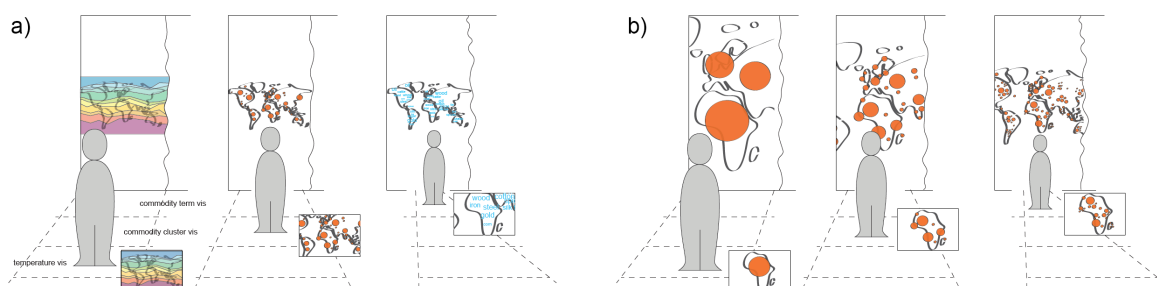


Abbildung 2.8: Interaktionen in der Anwendung SpiderEyes [DHKQ14]. a) Visualisierungsart ändert sich abhängig von der Distanz zur Displaywand b) Detailgrad der Visualisierung ändert sich abhängig von der Distanz zur Displaywand.

hier ist die bereits erwähnte Anwendung PaperLens. Die prototypische Umsetzung der räumlichen geschichteten Interaktion mit Graphen bezieht sich dabei auf die Auswahl durch Translationen und semantisches Zoomen durch die z-Achse-Translation bei geclusterten Graphen. Einige Anwendungen fokussieren sich auf die Analyse dreidimensionaler Netzwerke. In der von Betella et al. [BEM⁺14] entwickelten Anwendung kann der Nutzer durch die physischen Hand-Gesten die Knoten und Kanten erstellen, löschen und konfigurieren. Dies wird durch das Tracking der Nutzerbewegungen ermöglicht.

In vielen existierenden Anwendungen für die Graph-Analyse erfolgt die Interaktion jedoch eher mit dem Interface der Anwendung, bei der die gewünschten Parameter für die Visualisierung und Filterung einstellbar sind. Solche Anwendungen sind vor allem nötig, wenn kompliziertere Abfragen an die Daten erfolgen. Ein Beispielvertreter ist BiologicalNetworks [BSRG06], eine Anwendung für Visualisierung, detaillierte Analyse, Konstruktion, Abfragen an biologische Netzwerke wie Protein- oder DNA-Strukturen.

2.3.4 Visualisierung temporaler Daten

Aigner et al. [AMM⁺07] klassifizieren die existierenden Ansätze für die temporalen Daten nach Kriterien Zeit, Daten und Repräsentation. Die Zeit-Primitiven können Zeitpunkte oder Zeitintervalle sein, die Struktur der Zeit eine lineare, zyklische oder verzweigte Form haben. Die Daten können abstrakt oder räumlich sowie univariat oder multivariat sein. Die Zeit-Abhängigkeit der Repräsentation kann statisch oder dynamisch sein, insgesamt die Repräsentation zwei- oder dreidimensional.

Es wurden viele Ansätze zur Visualisierung und Interaktionen mit der Zeitdimension veröffentlicht, die meisten von ihnen beschäftigen sich nur mit spezifischen Problemen. Nach Aigner liegt es daran, dass die Einbeziehung von allen Zeitaspekten enorm schwierig ist. In der von Kumar et al. [KG06] entwickelten Anwendung können die dynamischen Graphen durch einen Slider in zeitlicher Dimension untersucht werden. Damit sich die Layouts von verschiedenen Graphzuständen nicht zu sehr verändern, wird der erste Graphzustand als Basis genommen. Später auftretende Zustände werden von diesem Zustand abgeleitet. Im von Morawa et al. [MHK⁺14] entwickelten Nornir wird die Timeline-Ansicht mit der Node-Link Repräsentation von Graphen kombiniert. Für die Abbildung der Zeit-Variable wird ein Schatten verwendet, das Zeitintervall wird in die Länge des Schattens einkodiert. Die Knoten werden an einer Timeline angeordnet.

Bei den linsenbasierten Ansätzen wird die Linse für die Steuerung der Zeit verwendet. Stoev et al. [SFR01] entwickeln einen Toolset für die Visualisierung und Interaktionen mit historischen dreidimensionalen Daten, beispielsweise Geo-Daten, bei denen die Zeit die vierte Dimension darstellt. TimeLens beruht auf dem Magic Lens Konzept und stellt ein Fenster dar, das über die Geo-Daten bewegt werden kann. Die Zeitdimension wird im Fenster durch einen Slider gesteuert. Hiermit zu vergleichen ist das Konzept von Temporal Magic Lens [RLE05] für Videosequenzen. An ein Video können räumliche und temporale Abfragen gestellt werden, wobei die Einstellungen von bestimmten Bewegungskurven und die Zeit-Intervalle-Einstellungen möglich sind. Zusätzlich wird die Objektverfolgung in Raum und Zeit unterstützt. In den oben erwähnten Konzepten von Spindler et al. erfolgt die Navigation in der Zeit durch räumliche geschichtete Interaktion. Beispielsweise wird das Konzept für die temporalen Untersuchungen der Geo-Daten umgesetzt.

2.3.5 Zusammenfassung

In diesem Abschnitt wurden einige Beispiele aus den für diese Arbeit relevanten Bereichen vorgestellt. Es ist festzustellen, dass der Einsatz von Mobilgeräten für die Interaktionen mit großen Displays ein relativ neues und intensiv zu untersuchendes Gebiet ist, auf dem mehrere interaktive Konzepte existieren. Öfter wird das Mobilgerät für die Steuerung oder Erweiterung der Display-Inhalte eingesetzt. Die Touch-Methode ist dabei sowohl bei direkten Interaktionen mit dem Display als auch für die kombinierten Interaktionen mit dem Mobilgerät verbreitet. Bei räumlichen Interaktionen werden öfter die Gesten-Konzepte umgesetzt, da sie keine zusätzliche Hardware benötigen. Andere räumliche Methoden existieren in Form von Konzepten und werden von relativ wenigen Forschern umgesetzt. Ansatz zur räumlichen geschichteten Interaktion ist konzeptuell unter anderem für die temporalen Daten entwickelt und bildet die Grundlage für die Gestaltung des interaktiven Raums in vorliegender Arbeit. Touch- und räumliche Gesten verfügen über gutes Potential für Steuerung der Funktionalitäten in der Konzeption.

Bei einigen existierenden interaktiven Konzepten für Graphen gibt es nach dem Stand der Recherche wenige, bei denen die zeitliche Dimension explorativ untersucht werden kann, und keine, bei denen dafür die räumlichen Interaktionen mit Einbezug der Fokus + Kontext-Methodik eingesetzt werden. In dieser Arbeit soll versucht werden, neue Konzepte zu entwickeln, die durch Einsatz der kombinierten Anwendung Displaywand + Mobilgerät diese Lücke füllen.

3 Konzeption

Im folgenden Kapitel werden die grundlegenden Konzepte für die Interaktion mit dynamischen Graphen ausgearbeitet. Diese beabsichtigen die Möglichkeit einer explorativen Analyse der zeitlichen Dimension in dynamischen Graphen durch geeignete Interaktionen und Visualisierungen. Im Abschnitt 3.1 wird ein Konzept für die Abbildung der zeitlichen Dimension sowie die Eingliederung anderer Datendimensionen im interaktiven Raum beschrieben. Abschnitt 3.2 fokussiert sich auf geeignete Techniken der Visualisierung von Datenänderungen im Graphen. Im Abschnitt 3.3 wird auf die für andere Konzepte hilfreiche Funktion der Einfrierung eingegangen. Abschnitt 3.5 präsentiert ein Konzept für multiple Ansichten.

3.1 Abbildung der zeitlichen Dimension im Interaktionssystem

Im Abschnitt 3.1.1 werden die Rollen der Bestandteile des Interaktionssystems beschrieben. Abschnitte 3.1.2 und 3.1.3 behandeln die Gestaltung der zeitlichen Dimension und ihrer Einordnung im interaktiven Raum.

3.1.1 Bestandteile des Interaktionssystems

Das grundlegende Interaktionssystem besteht aus der Display-Displaywand, dem Mobilgerät und dem Nutzer (Abbildung 3.1). Der Koordinatenursprung befindet sich in der linken oberen Ecke der Displaywand. Das System soll die Möglichkeiten für die Durchführung der Interaktionsaufgaben anbieten.

Der Nutzer spielt die zentrale Rolle bei der Gestaltung des interaktiven Raums. Dabei sind mehrere

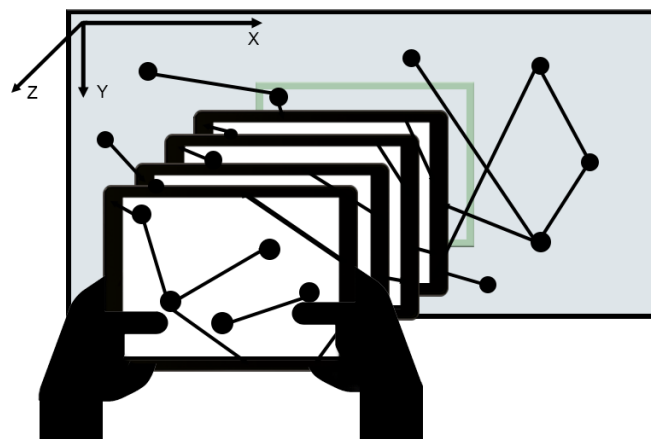


Abbildung 3.1: Zeitschichten entlang der z-Achse.

Aspekte wie die Wahrnehmung, bestimmte Absichten, Präferenzen und physiologische Eigenschaften zu beachten.

Die Rolle der Display-Displaywand besteht darin, den räumlichen oder den zeitlichen Kontext für den Nutzer bereitzustellen. Bei dem räumlichen Kontext handelt es sich um die räumliche Einordnung des Teilgraphen in seinem Zustand. Dafür ist es sinnvoll an der Displaywand den gleichen Zustand G_i des gesamten Graphen wie im Mobilgerät zu visualisieren. Beim zeitlichen Kontext kann die Displaywand den neusten Zustand des Graphen repräsentieren, so dass ein Vergleich zum ausgewählten Zustand direkt möglich ist. Der neueste Zustand des Graphen wird zur Default-Einstellung des Anzeigemodus der Displaywand, da der Fokus des Systems zeitliche Exploration ist. Für Vergleiche zwischen zwei beliebigen Zuständen kann eine Einstellung des Zustandes des Graphen an der Displaywand hilfreich sein. Das System soll somit eine Möglichkeit zur Umschaltung zwischen den Zuständen an der Displaywand anbieten.

Durch die physische Navigation des Nutzers in x- und y-Richtung entlang der Displaywand wird der Fokusbereich ausgewählt und im Mobilgerät angezeigt. Das Gerät soll zudem einen zeitlichen Zustand des ausgewählten Fokusbereichs präsentieren. Das Hauptziel des Einsatzes des Mobilgeräts ist die Exploration der einzelnen Zeitzustände des ausgewählten Teilgraphen, Vergleiche zwischen benachbarten Zuständen sowie Vergleiche mit dem an der Displaywand dargestellten Zustand.

Die Interaktionsaufgaben können allgemein in Manipulation der Daten und Manipulation der Ansicht der Daten unterteilt werden. Es ist sinnvoll, die schreibenden Manipulationen der Daten, beispielsweise das Hinzufügen von neuen oder das Löschen von alten Elementen, lediglich im jüngsten Zustand des Graphen zu erlauben, da die Änderung der Vergangenheit allgemein gegen die existierenden physikalischen Regeln verstößt. Manipulation der Ansicht dient der Analyse der Daten und soll in allen Zuständen möglich sein.

3.1.2 Zeitliche Dimension

In der Informationsvisualisierung gibt es mehrere Möglichkeiten der Zeitvisualisierung. Die Zeitvariable kann dabei verschiedene Eigenschaften haben. Die Zeitvariable in dynamischen Node-Link-Diagrammen hat eine lineare Struktur und kann entweder kontinuierlich verlaufen oder in einzelne Intervalle aufgeteilt werden.

In den meisten existierenden Ansätzen wie Timeline, Scatterplots usw. wird die zeitliche Variable auf eine der Achsen abgebildet, häufig auf der x-Achse. In den in Related Work präsentierten interaktiven Anwendungen [SFR01], [KG06] wird für die Steuerung der Zeit-Dimension ein Slider eingesetzt. Bei diesem Ansatz müsste man für die Navigation in Zeit und Raum sowohl räumliche als auch die Touch-Interaktion zur Bewegung des Sliders einsetzen. Die Richtung der Bewegung des Sliders würde parallel zu einer der Navigationsrichtungen verlaufen. Zwar gibt es nach dem Stand des Wissens noch keine Belegstudien, aber es lässt sich vermuten, dass die Verwendung der gleichen Achse für die Navigation in Zeit und Raum in bestimmten Situationen zu kognitiven Störungen führen könnte.

Für die Abbildung der zeitlichen Zustände eignet sich die zur Oberfläche der Displaywand orthogonale z-Achse. Basis dieser Überlegungen sind die in Related Work aufgeführten Ansätze zur räumlichen geschichteten Interaktion [STSD10] und vergleichbarer Zonen-Interaktion [Leh15]. Die Raumrichtung für die Navigation in der zeitlichen Dimension kann in den interaktiven Raum mit

einer physischen Navigation entlang des Graphen organischer eingeführt werden.

Bei dem Einsatz der z -Achse bewegt sich der Nutzer vor der Displaywand vor und zurück und verwendet parallel das Mobilgerät für die Exploration der Zeitschichten. Der Abstand zur Displaywand legt die Zeitachse fest. Die Zeitrichtung kann in die Richtung zur Displaywand oder in die entgegengesetzte Richtung verlaufen. Bei der Richtung zur Displaywand ist t_0 am weitesten von der Displaywand entfernt. Die Bewegung auf die Displaywand zu repräsentiert die Bewegung zum neusten Zustand des Teilgraphen. Dies ist dadurch zu begründen, dass die Bewegung nach vorne die Richtung der Zeit in die Zukunft symbolisiert und die Bewegung rückwärts die Vergangenheit. Wenn der Nutzer nah an der Displaywand steht, positioniert er sich somit zeitnah an der Gegenwart des Graphen. Wenn an der Displaywand jedoch nicht der neuste Zustand des Graphen repräsentiert wird, kann es zu Störungen des mentalen Bilds der Interaktion mit der Zeitvariable kommen, wodurch die Notwendigkeit zusätzlicher Navigationshilfe entsteht. Es ist jedoch davon ausgehen, dass die Intuition auch in diesem Fall beibehalten bleibt, da die Bewegung nach vorne allgemein mit der Bewegung in die Zukunft assoziiert wird.

Die Zeitzustände des Graphen werden in n Intervalle aufgeteilt. Die Änderungen der Daten entlang eines Intervalls werden in einem Schritt zusammengefasst, der eine zeitliche Schicht im Informationsraum darstellt. Die Entscheidung, die Schritte nicht in Zeitpunkten, sondern intervallweise zu definieren, ist dadurch zu begründen, dass bei hohen Änderungsraten der Daten die Länge der Schritte entlang der limitierten Bewegungsstrecke zu schmal und ihre Anzahl zu hoch werden könnte.

Abhängig von der durchzuführenden Aufgabe am Graphen kann eine bestimmte Intervalllänge und Intervallanzahl sinnvoll sein. Möchte der Nutzer möglichst genau die Zeitzustände eines Fokus untersuchen, so wird eine hohe Intervallanzahl benötigt. Wenn die Absicht der Interaktion die Exploration eines bestimmten selektierten Zeitzustands oder eine gezielte Suche ist, müssen bei der Aufteilung der z -Achse die Bewegungen entlang der Displaywand-Ebene berücksichtigt werden. Die Intervalllänge soll dabei so ausgewählt werden, dass die unbeabsichtigten Bewegungen wie das Handzittern keine Auswirkungen auf die Selektion des Zustandes haben, was zur Verkleinerung der Intervallanzahl führt. Bei kombinierten Zielen müssen Kompromisse gefunden werden, beispielsweise durch Einsatz von anderen interaktiven Techniken wie Einfrierung. Allgemein existieren nach dem Stand des Wissens noch keine Studien über die räumliche geschichtete Interaktion im Zusammenhang mit vertikalen Displays und Ganzkörper-Navigation. Spindler et al. untersuchen lediglich die Interaktionen mit horizontalen Oberflächen. Die Studie von Müller et al. [MRJR15] hat gezeigt, dass die Navigation entlang einer vertikalen Oberfläche einen erhöhten physischen Einsatz beansprucht, da die horizontalen Ganzkörperbewegungen und die vertikale Bewegung des Mobilgeräts gefordert sind. Auf der horizontalen Oberfläche konnte man sich in der Studie bewegen, wodurch die Bewegung des Mobilgeräts implizit gegeben war. Spindlers Tabletop ist wesentlich kleiner und erfordert somit weniger Ganzkörper- und mehr Handbewegungen.

Bei der Aufteilung in Intervalle kann zudem die vorliegende Datenstruktur beachtet werden. Statische Aufteilung der z -Achse, dargestellt in der Abbildung 3.2 a), berücksichtigt die Gesamtlebensdauer des Graphen. Die einzelnen Änderungen werden durch schwarze Punkte dargestellt. Die Timeline fängt bei der Graph-Definition an und endet bei der letzten Graph-Änderung, die nicht in der ausgewählten Knotenmenge enthalten ist. Vorteilhaft ist dabei, dass die Anzahl und die Länge der Intervalle unabhängig vom ausgewählten Fokusbereich gleich bleibt, wodurch der Nutzer ihre grobe Anordnung mithilfe seines räumlichen Gedächtnisses merken kann. In der Abbildung sind auf 3 Intervallen keine

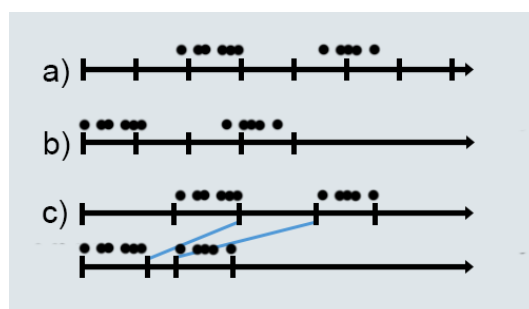


Abbildung 3.2: a) Die Timeline ist statisch aufgeteilt. b) Die Timeline ist statisch aufgeteilt, fängt bei der ersten Änderung der ausgewählten Knotenmenge an und endet bei der letzten. c) Die Timeline ist dynamisch aufgeteilt, fängt bei der ersten Änderung der ausgewählten Knotenmenge an und endet bei der letzten. Die Länge des Bereichs ohne Änderungen ist verkürzt.

Änderungen vorhanden. In ungünstigen Fällen, wenn die Änderungen ungleich verteilt sind, kann die statische Aufteilung zu schlechter Platzausnutzung entlang der z-Achse führen. Außerdem wird eine Hilfe zur Orientierung in Intervallen vom System benötigt, da bei den nicht vorhandenen Änderungen die Intervalle gleich aussehen.

Eine mögliche Verbesserung der Platzausnutzung bringt die Berücksichtigung des ausgewählten Teilgraphen, Abbildung 3.2 b). Die Timeline fängt bei der ersten Änderung der gewählten Knotenmenge an und endet mit der Letzten. Frühere oder spätere Zeitintervalle des Graphen, in denen keine Änderungen der Knotenmenge stattfinden, werden nicht dargestellt. Somit kann die freigestellte Länge für eine höhere Anzahl von Änderungen verwendet werden. Problemhaft wird es dabei mit der Intervallanzahl. Sie kann für verschiedene Fokusbereiche abweichend sein und wird schlechter gemerkt. Während die zeitlichen Längen der Intervalle gleich bleiben, werden sie somit in verschiedenen Fokusbereichen auf verschiedenen räumlichen Längen abgebildet, was das mentale Bild der ganzheitlichen Zeitschichten des Graphen stört. Die mittleren Intervalle werden nicht berücksichtigt: In der Abbildung sind zwei Intervalle fast leer.

Um den Platz noch besser auszunutzen, kann die Länge der Intervalle dynamisch abhängig von der Änderungsaufteilung bestimmt werden. Die Abbildung 3.2 c) repräsentiert eine Variation der dynamischen Aufteilung: Leere Zeitintervalle sind auf kürzere Länge abgebildet, wodurch mehr Platz für nicht leere Intervalle bleibt. Die Timeline verwendet einen verzerrungsbasierten polyfocalen Ansatz mit lokalen Fokussen auf Bereiche mit vorhandenen Änderungen. Dabei entsteht zusätzlich zu den Problemen des vorherigen Ansatzes noch ein Problem: Die Auffassung der Zeit als linearer Variable wird beeinträchtigt. Das System soll den Nutzer über diese gestörte Zeitlinearität benachrichtigen.

Tabelle 3.1: Aspekte der Intervalleaufteilungen.

	Statisch	Dynamisch, zeitl. Grenzen d. Teilgraphen	Dynamisch, Änderungsaufteilung
Platznutzung	0	+	++
Mentales Bild der zeitl. Schichten	++	0	0
Zeitlinearität	++	++	0

In der Tabelle 3.1 sind Aspekte bei drei beschriebenen Intervallaufteilungen zusammengefasst. Die Bewertung der Unterstützung der gewünschten Eigenschaften ist in 3 Stufen aufgeteilt: 0 für keine oder sehr geringe Unterstützung, + für teilweise und ++ für gute Unterstützung. Abhängig von den Eigenschaften der Datenstruktur eignet sich der erste Ansatz für Daten mit relativ gleich verteilten Änderungen. Der zweite Ansatz ist bei hohen Änderungsraten gut und der letzte Ansatz nur für die Daten mit sowohl hohen Änderungsraten als auch ungleicher Verteilung.

Die Aufteilung der Zeit in Intervalle ergibt ein Visualisierungsproblem bei mehreren Änderungen einer Eigenschaft eines Objekts auf einem Intervall. Es stellt sich die Frage, ob die Zwischenänderungen visualisiert werden. Da der Fokus auf den Gesamtfluss der dynamischen Daten gelegt wird, wird angenommen, dass die kurzen Zwischenänderungen weniger relevant sind und bei der Hauptvisualisierung nicht dargestellt werden. Für die Betrachtung des genauen Änderungsverlaufs kann eine separate Ansicht verwendet werden.

Die Zeitskala kann relativ oder absolut sein. Bei der absoluten Zeitskala handelt es sich um konkrete Zeitangaben, beispielsweise *20.07.16 13:37 Uhr*. Diese Skala steht in Abhängigkeit vom Geburtszeitwert des ältesten (t_0) und vom Wert der letzten Änderung des jüngsten Knoten ($t_n - 1$) im Teilgraphen. Die relative Zeitskala bezieht sich auf die Intervalle und ihre Gesamtzahl und kann zeigen, in welchem Intervall man sich gerade befindet, beispielsweise Intervallnummer 2 von 10. Die Kenntnisse sowohl über den relativen Bezug auf die Anzahl der Intervalle als auch über die absoluten Zeitwerte können für die Analyse bedeutend sein.

3.1.3 Der interaktive Raum

Im Allgemeinen besteht der interaktive Raum aus dem physischen Raum, in welchem sich die Displaywand und der Nutzer mit dem Mobilgerät befinden, sowie dem virtuellen Raum der Umgebung mit Daten. Die physischen Interaktionen im System werden auf die virtuellen Funktionalitäten des Systems übertragen. Sie können allgemein nach ihrem Bezugssystem unterteilt werden. Für die Exploration der Zeitschichten ist die Displaywand das Bezugssystem. Je nachdem, wo sich der Nutzer bezüglich des Koordinatenursprungs befindet, wird auf dem Mobilgerät ein anderer Zustand des Graphen sowie ein anderer Teilgraph dargestellt.

Der Nutzer kann ein weiteres Bezugssystem repräsentieren. Im Gegenteil zu der statischen Position der Displaywand kann sich die Position des Nutzers ändern, und das Bezugssystem ist dynamisch. Die räumliche Interaktion im System kann somit Displaywand- oder körperbezogen sein. Die räumlichen körperbezogenen Bewegungen können eine Abbildung bestimmter Funktionalitäten darstellen. Neben den temporalen geschichteten Daten existieren noch weitere Arten: zoombare, wenn im Graph semantisches oder geometrisches Zoomen möglich ist, oder volumetrische bei der dreidimensionalen Struktur des Graphen. Durch die Verwendung des Körperbezugs können diese Daten durch Abbildung auf eine Raumrichtung in den interaktiven Raum integriert werden. Die Distanz beispielsweise auf der z-Achse zwischen dem Gerät und dem Nutzer, begrenzt auf die physische Länge der Arme, kann Schichten dieser Daten repräsentieren. Dazu wird zusätzlich zu den Kenntnissen der Position des Geräts die Position des Nutzerkörpers benötigt.

In der Abbildung 3.3 ist die Vereinigung beider Bezugssysteme zu sehen. Die Distanz zwischen dem ersten Nutzer und dem Gerät ist maximal. Dies kann beispielsweise die letzte Schicht der volumetrischen Daten oder die letzte Zooming-Stufe bedeuten. Die Distanz zwischen dem zweiten Nutzer

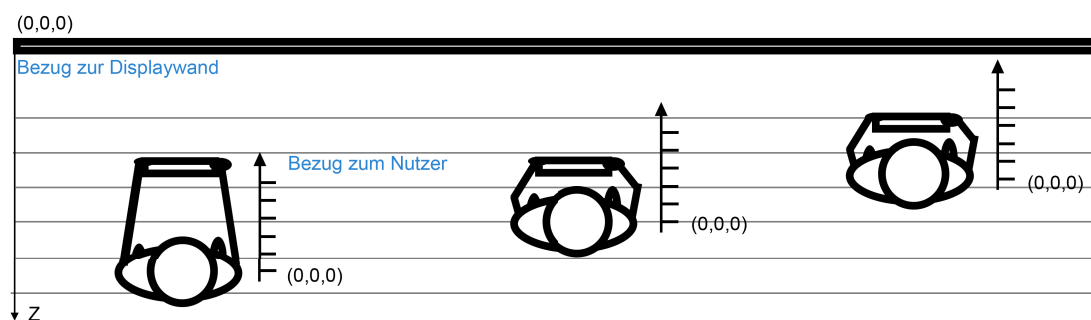


Abbildung 3.3: Bezugssysteme im interaktiven Raum.

und seinem Mobilgerät ist durch die Verbiegung der Arme kleiner als bei dem ersten. Gleichzeitig bleibt das Mobilgerät im gleichen Zeitintervall wie das des ersten Nutzer. Der dritte Nutzer befindet sich im anderen Zeitintervall und in der gleichen körperbezogenen Schicht der Daten wie der zweite.

Durch die Einbeziehung mehrerer Bezugssysteme können mehrere geschichtete Informationen auf die Bewegung abgebildet werden. Verallgemeinert kann man verschiedene geschichtete Daten nach ihrer Wichtigkeit für die Untersuchungen in primäre, sekundäre usw. aufteilen. Das statische Bezugssystem Displaywand erscheint für die primären Daten sinnvoller, da eine größere Anzahl der Schichten darstellbar und der Detailgrad folglich höher ist. Gleichzeitig ergeben sich bei mehreren Bezugssystemen einige Probleme. Die Navigation gleichzeitig in mehreren geschichteten Räumen erfordert einen erhöhten kognitiven Aufwand dadurch, dass der Nutzer nicht nur die Position des Geräts zu achten hat, sondern auch seine eigene. Dies kann einen negativen Einfluss auf die Koordination der Bewegungen ausüben. Zudem kann eine Haltung des Mobilgeräts mit weit gestreckten Armen physisch unbequem sein. Hier kann das Konzept der Einfrierung behilflich sein. Ganz vorne vor der Displaywand muss darauf geachtet werden, dass die Kombination beider Bezugssysteme nicht zu einer zu nahen Positionierung des Nutzers an der Displaywand führt. Dafür kann das entsprechende Intervall etwas verlängert werden, wie in der Abbildung 3.3 gezeigt.

Alternativ zum Einsatz mehrerer Bezugssysteme können mehrere geschichtete Informationen auf eine Raumrichtung abgebildet werden. Dafür ist eine geeignete Umschaltung nötig. Die Koordination des Nutzers wird dabei im Vergleich zu einem Mehrbezügesystem vereinfacht. Verschiedene Schichtenarten werden auf jeweils die gleiche Weise exploriert. Gleichzeitig gehen die Abhängigkeiten zwischen den umzuschaltenden Schichtenarten für die Wahrnehmung des Nutzers verloren, da die Interaktionen mit ihnen nicht gleichzeitig durchgeführt werden können.

Zoombare Informationen sind eine besondere Art der geschichteten Daten, deren Verwendung für die Mobilgeräte sehr verbreitet ist. In mobilen Browsern und zahlreichen anderen Anwendungen wird für das Zoomen die Pinch-Geste gebraucht. Im System kann sie alternativ zu den räumlichen Ansätzen für die Exploration der Zooming-Stufen eingesetzt werden. Potentiell ist die Exploration gleichzeitig mit der Exploration der Zeitschichten verwendbar. Jedoch wird durch das gleichzeitige Durchführen mehrerer Interaktionsaufgaben eine höhere Konzentration erfordert.

Die räumlichen Gesten mit dem Gerät können je nach ihrer Funktion mit statischem oder dynamischem Bezug umgesetzt werden. Die Gestaltung der Gesten im statischen Bezugssystem kann sich beispielsweise auf die horizontale oder die vertikale Orientierung des Geräts beziehen sowie auf die Informationen an der Displaywand. Die körperbezogenen Gesten können als Gesten aufgefasst wer-

den, wenn ein Wechsel der Haltung des Geräts stattfindet. In den Konzepten, in denen die Gesten eingesetzt werden können, wird jeweils diskutiert, welche dieser Optionen abhängig von der Gestenfunktion sinnvoll ist. Für die Umschaltung zwischen räumlichem und zeitlichem Kontext an der Displaywand ist eine absolute Geste, beispielsweise eine Werfgeste in Richtung der Displaywand sinnvoller, da sie sich auf die absolute Position der Displaywand bezieht.

3.1.4 Anwendungsszenarien

Folgende Anwendungsszenarien sollen das Verständnis dafür erleichtern, wie die entwickelte Konzeption zur Abbildung der zeitlichen Dimension auf die Raumrichtung eingesetzt werden kann:

- Der Nutzer möchte die Zeitzustände in einem gewünschten Fokusbereich ansehen. Gleichzeitig möchte er im Entwicklungskontext des Graphen bleiben. Die Interaktion erfolgt in folgenden Schritten: Zuerst erfolgt die Auswahl der räumlichen Position des Fokusbereichs durch Navigation entlang der x- und y-Achsen. Dann bewegt sich der Nutzer entlang der z-Achse, um sich die einzelnen Zustände des Fokusbereichs auf den Intervallen anzusehen. Durch die Anzeige des letzten Zustandes an der Displaywand kann der Nutzer jederzeit sehen, wie sehr sich die ausgewählten Zeitzustände davon unterscheiden.
- Der Nutzer wählt einen Zeitzustand und möchte den Graphen in diesem Zustand explorieren. Gleichzeitig möchte er im räumlichen Kontext des Graphzustandes bleiben. Für die Auswahl der Zeitschicht navigiert der Nutzer entlang der z-Achse. Dann bewegt er sich entlang der Displaywandebene für die Exploration der einzelnen Bereiche des Graphen. Um den aktuellen Zustand an der Displaywand von der Default-Einstellung zu ändern, verwendet der Nutzer eine Werfgeste in Richtung der Displaywand.
- Der Nutzer sucht nach bestimmten Knotenzuständen, in welchen die Knotengröße besonders groß ist. Dafür lässt der Nutzer die entsprechenden Zeitzustände an der Displaywand durch Werfgesten aus den zugehörigen Intervallen anzeigen.
- Der Nutzer möchte die Zeitzustände von einem dreidimensionalen Graphen aus verschiedenen Tiefenstufen betrachten. Zusätzlich zu der Navigation entlang der z-Achse erfolgt dabei die zum Nutzerkörper relative Bewegung des Geräts.

3.2 Visualisierung der Datenänderungen

Im Folgenden sollen Optionen diskutiert werden, wie die Änderungen der Graph-Daten auf den einzelnen Intervallen dargestellt werden sollen. Dies ermöglicht das Vergleichen der Zustände. Die Änderungen können sich auf die Knoten und Kanten sowie ihre Eigenschaften beziehen und werden in Geburt vom neuen Knoten, neuer Kante, Tod des Knoten oder der Kante und Änderung der spezifischen Eigenschaft unterteilt. Die Gesamtdauer des Lebenszyklus wird Alter genannt. Geburt und Tod sind die Lebenszyklusgrenzen des Elements und bedeuten das Entstehen bzw. die Aufnahme in die Datenmenge sowie das Löschen aus der Datenmenge. Grundsätzlich besteht die Aufgabe, die Änderungen auf bestimmte visuelle Variable abzubilden. Dabei ist zu beachten, dass einige Variablen in der Graph-Visualisierung bereits verwendet werden können, wodurch mehrere Alternativen nötig sind.

Abschnitte 3.2.1 und 3.2.2 befassen sich mit der Visualisierung der Änderungen und ihren Bezugsintervallen. Im Abschnitt 3.2.3 wird näher auf die Optionen der Auswahl der zu verfolgenden Elemente eingegangen. Abschnitt 3.2.4 beschreibt das Konzept für die Exploration der Änderungen innerhalb eines Intervalls.

3.2.1 Visuelle Kodierung in einem Graph-Zustand

Die Geburt auf einem Intervall als Spezialfall des Alters ist dadurch gekennzeichnet, dass der Knoten oder die Kante auf dem vorherigen Intervall nicht existiert hat. Eine Hervorhebung des neu abgebildeten Elements kann sinnvoll sein, wenn man die Tendenzen oder die Ausbreiter der Größenänderungen des Fokusbereichs auf einem Intervall beobachten möchte. Dafür eignen sich beispielsweise der Farbton, Helligkeit, die Form des Knotens bzw. die Linienart der Kante, oder eine Kennzeichnung mit Glyph, da sie keine spezifische intrinsische Bedeutung haben. Die Abbildung 3.4 a) präsentiert den Einsatz von Farbe für eine Geburt.

Anhand von Kenntnissen über die Lebenszykluslängen der Elemente kann Rückschluss über die Eigenschaften der quantitativen Entwicklung der Daten in vorherigen Intervallen gemacht werden. Beispielsweise kann beobachtet werden, welcher Alter dominiert und ob es mögliche Zusammenhänge mit anderen Eigenschaften gibt. Alter ist eine quantitative Variable, die sich gut mit der Länge darstellen lässt, wie im Ansatz Time Shadow [MHK⁺14]. Problematisch wird bei Graphen die Überlappung der Schatten der naheliegenden Knoten. Alternativ lässt sich das Alter einstimmig mit der Kennzeichnung der Geburt durch einen Farbverlauf oder Helligkeitsabstufungen darstellen, wobei sich die Form für die qualitativen Variablen nicht eignet. Eine weitere mögliche Variable für das Alter ist Größe. Angelehnt an das Wachstum lebendiger Organismen mit dem Alter können auch die Größenabstufungen Alter symbolisieren. Durch die Darstellung des Alters kann die Wahrnehmung des zeitlichen Kontextes gestärkt werden. In Abbildung 3.4 b) ist eine solche Kodierung des Alters dargestellt. Es gibt 3 Farbstufen: Hellblau für die jüngsten, auf dem aktuellen Zeitintervall entstandene Elemente, Dunkelblau für Elemente, die auf dem vorherigen Intervall entstanden sind, und Schwarz für ältere Elemente. Blau wurde gewählt, da diese Farbe eine neutrale Wirkung erzeugt [Flo14] und als am wenigsten störende Farbe eingestuft wird. Abhängig davon, wie exakt die Kenntnisse über das Alter erforderlich sind, kann man eine variable Stufenunterteilung vornehmen.

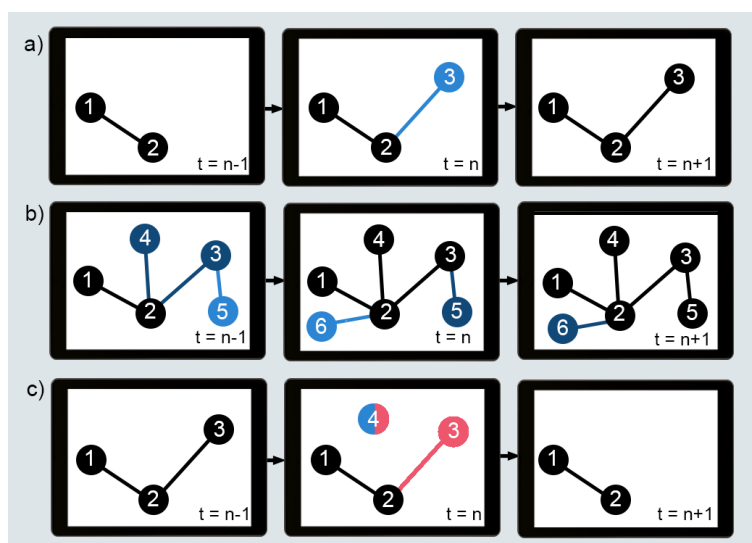


Abbildung 3.4: a) Auf dem Intervall $t = n$ werden neuer Knoten 3 und neue Kante hinzugefügt. b) 1 und 2 sind die ältesten Knoten. Knoten 3 und 4 sind etwas junger. Auf dem Intervall $t = n - 1$ wird neuer Knoten 5 hinzugefügt. Auf dem Intervall $t = n$ wird neuer Knoten 6 hinzugefügt. c) Auf dem Intervall $t = n$ werden Knoten 3 und die dazugehörige Kante gelöscht. Knoten 4 wird geboren und gelöscht.

Gelöschte Knoten und Kanten existieren auf einem vorherigen Intervall und nicht mehr auf dem ausgewählten. Eine Hervorhebung ist genau wie bei neu entstandenen Elementen für die Beobachtung der Tendenzen interessant. Aus dem Verhältnis der gelöschten zu den neu entstandenen Elementen kann Rückschluss über die Art der quantitativen Änderung des Intervalls gemacht werden: wächst die Datenmenge eher, verringert sie sich oder stagniert sie? Hervorgehoben werden können einerseits Elemente, die auf dem aktuellen Intervall gelöscht werden. Andererseits kann man auch die Elemente hervorheben, die im nächsten Intervall nicht existieren. Da das Löschen ein nachfolgendes, materielles Nichtvorhandensein verursacht, kann diese Änderung der Materialität beispielsweise durch Transparenz, Helligkeit oder Farbintensivität dargestellt werden. Transparente Objekte erscheinen als weniger materiell. Ein Beispiel des Einsatzes der Transparenz findet man bei der Visualisierung gelöschter Dateien, die noch nicht überschrieben wurden. Schwächere Farbintensivität symbolisiert inaktive, nicht gültige Objekte im Web-Design. Des öfteren wird für die Assoziationen mit Löschvorgängen die Farbe Rot eingesetzt, beispielsweise bei den Delete-Buttons [Lon15]. In Abbildung 3.4 ist Einsatz der Farbe für den auf dem Intervall gelöschten Knoten zu sehen. Einen besonderen Fall stellt die Situation dar, wenn ein Element auf dem Intervall geboren und gelöscht wurde. Wenn für beide Ereignisse eine Farbvariable eingesetzt wird, bleibt nur das Löschen sichtbar und man braucht eine spezielle Markierung für die Geburt, beispielsweise Färbung einer Hälfte in Geburtsfarbe und anderer Hälfte in Löschungsfarbe, wie auf dem Intervall $t = n$ für Knoten 4 zu sehen ist. Beim Ansatz der Transparenz für das Löschen sind beide Ereignisse unterscheidbar, werden jedoch langsamer wahrgenommen.

Die Eigenschaften der Knoten und Kanten sind abhängig von der Definition der Datenstruktur. Die Visualisierung ihrer Änderungen basiert darauf, ob und wie sie in der Implementierung visualisiert sind. Im Folgenden werden einige verbreitete visuelle Variablen betrachtet sowie die Fälle, in denen die Eigenschaft nicht visualisiert ist:

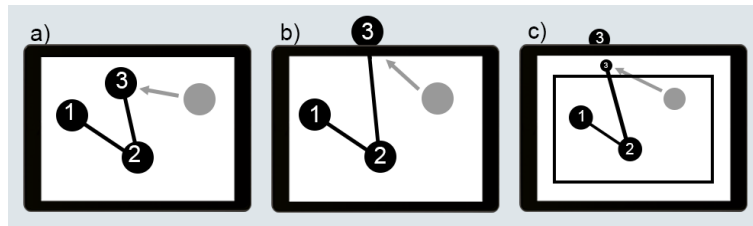


Abbildung 3.5: a) Knoten 3 wird innerhalb des Fokusbereichs verschoben. b) Knoten 3 wird außerhalb des Fokusbereichs verschoben und ist nicht mehr sichtbar. c) Knoten 3 wird außerhalb des Fokusbereichs verschoben. Seine verkleinerte Ansicht bleibt im Randbereich.

- Die Position stellt eine besondere Eigenschaft dar, die im 2D-Koordinatenraum auf sich selbst abgebildet ist. Die Positionsänderung kann dadurch dargestellt werden, dass sowohl die alte als auch die neue Position sichtbar sind. Ein Pfeil kann die Beziehung von der alten zur neuen Position andeuten. Die Transparenz kann andeuten, dass der Knoten für die Positionsdarstellung nicht existent ist. In Abbildung 3.5 a) ist eine Positionsänderung innerhalb des Fokusbereichs dargestellt. Der Vorteil der Positionsabbildung auf sich selbst liegt darin, dass alle anderen Attribute des abgebildeten Zustands an dieser Position abgebildet werden können, wodurch der aktuelle Zustand des Knotens visuell entlastet wird. Die Fälle, in denen die neue Position außerhalb des Fokusbereichs liegt, werden gesondert im Abschnitt 3.2.2 betrachtet. Die Positionsänderung kann ignoriert werden (Abbildung 3.5 b)). Man kann auch einen zusätzlichen Bereich einfügen, in dem die Position nicht wahrheitsgemäß abgebildet ist, wie in Konzepten von Frisch et al. [FD10] und Hossain et al. [HHL12] (Abbildung 3.5 c)).
- Die visuellen Variablen Form und Farbe sind Ausprägungen des Aussehens von Knoten. Damit man die Veränderung des Aussehens erkennt, muss das entsprechende Element markiert werden. Die auf der Abbildung 3.6 a) dargestellten Markierungen sind als Symbole innerhalb der Knoten platziert, deuten die Änderungen an und kodieren den vorherigen Zustand als zeitlichen Kontext. Ähnlich dazu kann die vorherige Farbe der Kante als kleine Markierung in dieser abgebildet werden.
- Knotengröße und Kantenbreite können maßstabgetreu abgebildet werden. In Abbildung 3.6 b) werden die Größen vor der Änderung durch gestrichelte Linien dargestellt. Die Farbe der gestrichelten Linie soll dabei sowohl vor dem Hintergrund als auch auf dem Knoten unterscheidbar sein.

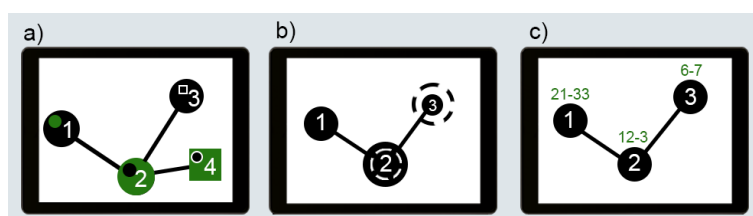


Abbildung 3.6: a) Bei Knoten 1 und 2 wird die Farbe geändert. Knoten 3 und 4 ändern die Farbe und die Form. b) Knoten 2 wird vergrößert, Knoten 3 verkleinert. c) Nicht visualisierte Integer-Werte ändern sich bei den Knoten.

- Die Darstellung der Änderungen von Eigenschaften ohne visuelle Variablen kann textuell erfolgen. In Abbildung 3.6 c) sind Veränderungen der Integer-Werte dargestellt. Diese sind an keine visuelle Variable geknüpft und werden als Texte in Form *alter Wert - neuer Wert* dargestellt.

Wenn sich bei einem Element mehrere Eigenschaften in einem Zeitintervall ändern, ist zu beachten, dass die Kombinationen der Visualisierungen nicht verwirrend oder überlastet wirkt und einzelne Teilvisualisierungen erkennbar bleiben. Wenn die Knotenposition eine der veränderten Variablen ist, kann die Visualisierung durch die Darstellung des alten Knotenzustandes an der alten Position vereinfacht werden.

Bei der Mehrzahl visueller Variablen und bei vielen Änderungen kann die Visualisierung zu unübersichtlich werden. Für die Fälle, in denen der Nutzer die zeitliche Visualisierung von bestimmten Änderungen benötigt, ist eine Funktion für die Auswahl dieser Eigenschaften nützlich. Die Eigenschaften können in einem Interface als Optionen abonniert werden. Ebenso kann es nützlich sein, wenn der Nutzer selbst bestimmen darf, welche visuelle Variable welche Eigenschaften kodiert. Dafür müssen für jede spezifische Eigenschaft eines Graph-Beispiels mehrere implementierte Alternativen der Variablen zur Verfügung stehen.

Diese Technik der Hervorhebung der Änderungen ist für die Durchführung mehrerer Graphaufgaben hilfreich. Die Anomalien und Änderungscluster können visuell bestimmt werden. Die Beziehung zweier Zustände einer Eigenschaft ist sichtbar. Vergleichen zwischen 2 Zuständen als zusammengesetzte High-Level Aufgabe nach [LPP⁺06] und [APS14] wird ermöglicht.

3.2.2 Bezugsintervall der Änderungen

Die Darstellung der Änderungen braucht zum einen ein vom Nutzer ausgewähltes Intervall, in welchem diese Änderungen visualisiert werden, und zum anderen ein Bezugsintervall, welches für die Feststellung der Änderungen verwendet wird. Bei der Darstellung der Änderungen im Bezug auf das vorherige Intervall, in welchem der frühere Zustand des Graphen liegt, wird die temporale Entwicklung untersucht. Dabei wird die Nutzerbewegung entlang der z-Achse für die Auswahl des Intervalls eingesetzt.

Eine Erweiterung stellt die Ausnutzung nicht nur der Nutzerbewegung, sondern auch der Bewegungsrichtung, dar. So kann bei der Bewegung nach vorne zur Displaywand das vorherige und bei der Bewegung nach hinten das darauf folgende Intervall als Bezugsintervall verwendet werden. Diese Erweiterung erlaubt die nutzerspezifische Fokussierung der Entwicklungsexploration auf den neueren oder den älteren Zustand.

3.2.3 Temporale Verfolgung der aktiven Knotenmenge

Ein weiterer Aspekt der zeitlichen Visualisierung ist die Auswahl der Knotenmenge, deren Änderungen verfolgt werden. Da bei den geografischen Karten die Position die primäre visuelle Variable ist, ist es sinnvoll, dass sich auch die Auswahl primär auf die Position bezieht. Dagegen sind bei den Graphen die Knoten als Objekte wichtig, die Position ist lediglich eine von mehreren Eigenschaften. Es stellt sich die Frage, was mit den Knoten passiert, die durch die Änderungen aus dem Fokusbereich auswandern und dennoch vom Nutzer verfolgt werden sollen.

In mehreren Arbeiten werden Fälle behandelt, in denen ein Objekt außerhalb des Fokusbereichs

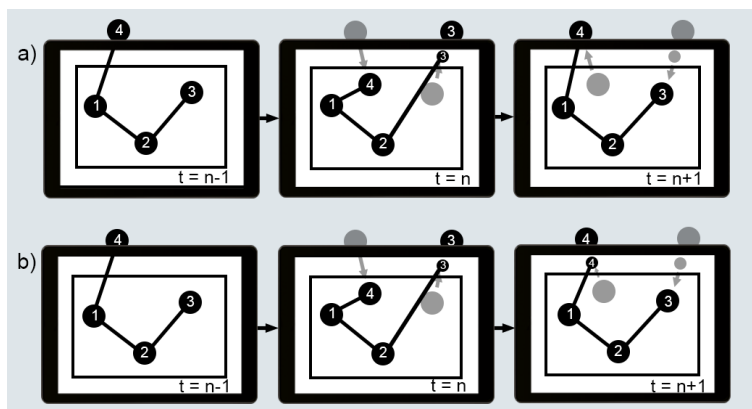


Abbildung 3.7: a) Auf dem Intervall $t = n - 1$ wurde aktive Knotenmenge $\{1, 2, 3\}$ gewählt. Auf dem Intervall $t = n$ wird Knoten 3 außerhalb des Fokusbereichs verschoben. Knoten 4 wird in den Fokusbereich verschoben. Auf dem Intervall $t = n + 1$ wird Knoten 3 zurück in den Fokusbereich verschoben. Knoten 4 wird in den Außenbereich verschoben. b) Gleiches Szenario wie in a), Knoten 4 wird auf dem Intervall $t = n$ in die aktive Menge aufgenommen.

dennoch angezeigt wird. Bei den Ansätzen von Frisch et al. [FD10] und Hossain et al. [HHL12] werden die Randbereiche für die Visualisierung von Off-Screen-Objekten verwendet. Bei Ghani et al. [GRE11] wird für jedes zu verfolgende Off-Screen-Objekt eine kleine Ansicht mit einem Ausschnitt seines Kontexts im Randbereich dargestellt. Diese Ansätze sollen eine Basis für die Visualisierung der Off-Screen-Elemente an den Randbereichen des Mobilgeräte-Displays bilden.

Die Variationen der Behandlung der Änderungen, die Off-Screen-Visualisierungen miteinbeziehen, beruhen darauf, ob die Position des Fokusbereichs und die aktive Knotenmenge fest bleiben.

- *Fester Fokusbereich, feste aktive Knotenmenge:* Zuerst wird durch die Navigation entlang der Achsen die Position des Bereichs in einem Zustand des Graphen festgelegt. Alle Knoten, die sich in diesem Bereich befinden, werden als aktive Knotenmenge definiert. Dann wird durch die Navigation entlang der z-Achse die zeitliche Dimension untersucht. Es werden Änderungen von allen Knoten der aktiven Menge verfolgt und dargestellt, auch wenn diese aus dem Fokusbereich auswandern (Abbildung 3.7 a)). Durch die Änderung der Position in früheren Zeitzuständen können in den Fokusbereich Knoten einwandern, die nicht zur aktiven Menge gehören. Displaywandern diese aus dem Fokusbereich aus, werden ihre Änderungen nicht mehr verfolgt. In der Abbildung ist die Behandlung der Positionsänderung eines zur aktiven Menge zugehörigen und eines nicht dazu gehörenden Knoten dargestellt.
- *Fester Fokusbereich, dynamische aktive Knotenmenge:* In dieser Kombination kann sich die aktive Knotenmenge abhängig von den Positionsänderungen der Knoten im Fokusbereich ändern. Die Knoten der ursprünglichen aktiven Menge können aus dieser gelöscht werden, wenn ihre neue Position außerhalb des Fokusbereichs liegt. Nicht zur ursprünglichen Menge gehörende Knoten können in diese aufgenommen werden, wenn ihre neue Position innerhalb des Fokusbereichs liegt. Ist beabsichtigt, dass alle ursprünglichen Knoten aus der aktiven Menge verfolgt werden können, so kann man das Löschen aus der Menge begrenzen. Der Nachteil ist, dass die aktive Menge auf diese Weise wächst, und die Komplexität der Visualisierung somit steigt. In Abbildung 3.7 b) ist die Aufnahme eines Knotens in die aktive Knotenmenge dargestellt.

- *Dynamischer Fokusbereich, dynamische aktive Knotenmenge*: Im Vergleich zu der vorherigen Variation erfolgen die Änderungen der aktiven Knotenmenge bei gleicher Positionsänderungsrate durch die Verschiebung des Fokusbereichs häufiger, wodurch das Verfolgen von bestimmten Knoten erheblich erschwert wird.
- *Dynamischer Fokusbereich, feste aktive Menge*: Soll die aktive Knotenmenge fest bleiben und darf man den Fokusbereich verschieben, so kann es dazu führen, dass die aktiven Knoten nicht mehr in diesem Bereich vorhanden sind und in den Randbereich wandern. Im ungünstigen Fall wandern alle aktiven Knoten, die für den Nutzer von besonderem Interesse sind, durch die Verschiebung des Fokusbereichs in den Randbereich, wodurch die Platzausnutzung verschlechtert wird.

Im Fall der festen oder der dynamisch wachsenden aktiven Knotenmenge braucht das System die Funktionalitäten für die Manipulationen an ihr. Dazu kann eine zusätzliche Ansicht der aktuellen Menge mit einer Auflistung der dazu gehörenden Knoten und Möglichkeiten für das Hinzufügen, Löschen oder einen Reset nützlich sein. Für das Hinzufügen kann Selektion eingesetzt werden. Das Löschen kann in der Auflistung erfolgen. Für einen Reset kann das Mobilgerät an der Stelle mit den neuen zu verfolgenden Knoten platziert werden.

Durch den Einsatz der virtuellen Verfolgung ausgewählter Elemente wird zusätzliche Navigation für physische Verfolgung und damit die körperliche Belastung minimiert. Gleichzeitig kommt es durch die Verwendung von Rändern zu einer Verkleinerung des Ausschnitts.

3.2.4 Visualisierung der Änderungen in einem Zeit-Intervall

Einen für die Visualisierung der Graph-Zustände problematischen Fall stellt die Situation dar, in der eine Eigenschaft mehrmals in einem Intervall geändert wird. Die im Abschnitt 3.2.1 entwickelten Visualisierungsrichtlinien zeigen den Endzustand des Teilgraphen aber keine Zwischenänderungen.

Eine spezielle Ansicht soll den genaueren Verlauf der Änderungen im ausgewählten Zeitintervall zeigen. Dazu wird ein Timeline-Ansatz verwendet. Jeder Zwischenzustand der Knoten wird in eigener Timeline abgebildet, wodurch ihre allgemeine Entwicklung sichtbar wird. Sollen lediglich die Änderungen sichtbar sein, können auch nur sie dargestellt werden. Die Zeitskala der Timeline ist genauer als in der Graph-Visualisierung, die genauen Änderungszeitpunkte werden erkennbar.

Während Größe, Form oder Farbe auf sich selbst abgebildet werden können, ist dies für die räumliche Position des Elements nicht möglich, da die x-Achse für die Zeit verwendet wird. Die Position wird textuell in Form (x, y) kodiert. In Abbildung 3.8 ist ein Beispiel der Änderungs-Timeline-Visualisierung für drei Knoten zu sehen. Beim Knoten 3 ändert sich die Position auf dem Intervall zweimal. Während bei der Endzustand-Visualisierung die Zwischenänderung nicht abgebildet wird, ist sie in der Timeline dieses Knotens zu sehen.

Möchte man zusätzlich die Entwicklung von Kanten ansehen, so können sie auch in der Timeline-Ansicht dargestellt werden. Kanten sind im Gegensatz zu Knoten längliche Elemente und können im richtigen Maßstab nicht visualisiert werden. Dagegen kann man die längliche Struktur für die Abbildung entlang der Timeline verwenden. In Abbildung 3.8 ist die Visualisierung der Änderung der Kante 1 auf dem Intervall i dargestellt. Kantenänderungen wie Dicke und Farbe können prinzipiell auch auf sich selbst abgebildet werden. Alternativ kann man kleine Kantenfragmente stellvertretend für vollständige Kanten nehmen und ähnlich wie die Knoten darstellen.

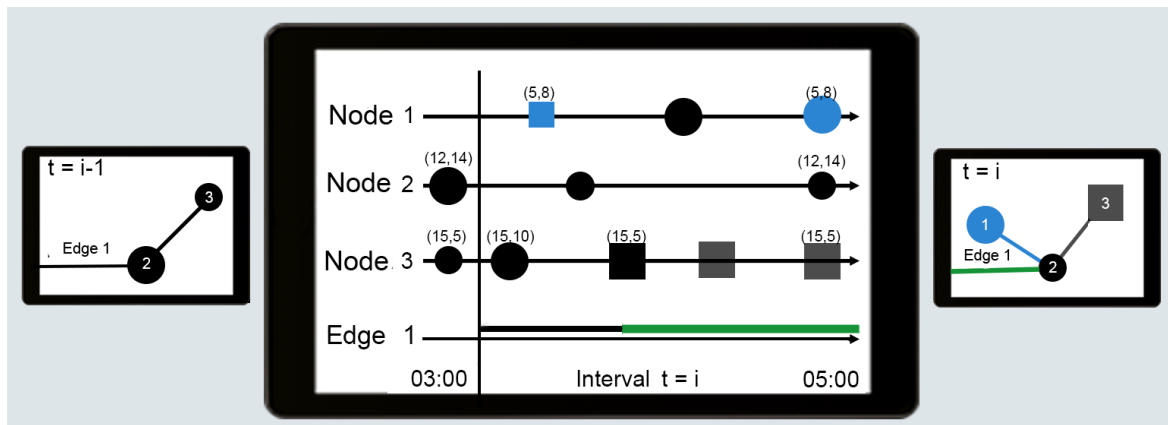


Abbildung 3.8: Knoten 1 wird im Intervall hinzugefügt und ändert seine Form zum Kreis und wird größer. Bei Knoten 2 wird die Größe auf dem Intervall kleiner. Knoten 3 wird größer, ändert seine Position sich zweimal und wird anschließend gelöscht. Kante 1 wird dicker und ändert die Farbe zu Grün.

Das Visualisierungsproblem der Timeline-Ansicht sind Überlappungen bei zu nah aneinander liegenden Änderungen. Unterschiedliche Behandlungsarten sollen das Problem vermindern. Relevant ist die Auswahl der horizontalen oder der vertikalen Mobilgerät-Orientierung. Bei der vertikalen Orientierung sind Timelines für eine größere Anzahl von Knoten darstellbar. Bei der horizontalen Orientierung dagegen sind die Timelines länger, wodurch sich die Skala-Granularität erhöht und mehr Änderungsdarstellungen ohne Überlappungen erlaubt. Diese Haltung wird präferiert, da der Fokus auf der Zeit liegt. Bei sehr vielen Knoten oder Kanten kann man beispielsweise eine Scrollbar verwenden. Eine weitere Behandlung der Überlappungen besteht darin, die kontinuierliche Zeitskala in eine diskrete umzuwandeln, so dass der Abstand zwischen zwei benachbarten diskreten Werten für Änderungsvisualisierungen auf beiden Stellen ausreicht. Als weitere Behandlungsmöglichkeit kann man die Größe aller angezeigten Knoten bei gleich bleibender Timelinelänge durch den Nutzer einstellbar machen.

Der Nutzer kann auswählen, welche Knoten er in der Timeline-Ansicht sehen möchte. Dafür kann die Funktion der Selektion gewünschter Knoten eingesetzt werden. Selektiert der Nutzer keine Knoten, wird die gesamte Knotenmenge des Teilgraphen in der Timeline dargestellt. Bei zu großer Knotenmenge kann Scrollen eingesetzt werden. Wird das Konzept der aktiven Menge verwendet, kann man die aktiven Knoten zuerst anzeigen, da diese von höherer Bedeutung sind. Zudem kann gezielte Selektion für gewünschte Knoten eingesetzt werden.

Die Timeline-Ansicht erlaubt präzise Vergleiche einerseits zwischen verschiedenen Knoten, andererseits zwischen verschiedenen Zuständen eines Knotens in einem Intervall. Aus den Timelines sind Entwicklungsbesonderheiten der Elemente ablesbar.

3.3 Einfrierung eines Teilgraphen-Zustandes

Die Navigation im Graphen und in seiner zeitlichen Dimension erfolgt durch die physischen Bewegungen des Nutzers. Wenn der Nutzer etwas länger mit einem Zustand des Graphen arbeiten möchte, kann das Halten des Mobilgeräts in einer Position anstrengend werden. Unbeabsichtigte Hände- oder

Körperbewegungen können die Usability erschweren. Das Konzept der Einfrierung eines Zustandes des Graphen soll eine Lösung für diese Probleme bieten. In der Studie von Lee et al. [LYK⁺09] wurde bestätigt, dass die Interaktionen mit einem Augmented Reality Gerät durch die Verwendung der Einfrierung exakter werden, insbesondere in Situationen, in denen der Nutzer in komplizierten Posen arbeiten muss.

Unter Einfrierung eines Visualisierungszustandes versteht man den Abbruch der dynamischen Visualisierungsänderung von diesem Zustand. In den Abschnitten 3.3.1 und 3.3.2 werden die Freiheitsgrade der Einfrierungen auf dem Mobilgerät und die Einordnung im visuellen Raum diskutiert. Die Basis wurde konzeptionell bei Spindler et al. [STSD10] präsentiert.

3.3.1 Freiheitsgrade der Einfrierung

Die Freiheitsgrade der Einfrierung basieren auf den Raumachsen. In den zuvor beschriebenen Konzepten wurde die Wichtigkeit dieses Konzeptes einige Male erwähnt. Einfrierung der x- oder der y-Bewegungsrichtungen wird eingesetzt, wenn die Position des Fokusbereichs ausgewählt ist und keine weitere Navigation an der Oberfläche der Displaywand nötig ist. Gleichzeitig kann der Nutzer durch die Bewegung entlang der z-Achse die zeitliche Dimension des ausgewählten Teilgraphen explorieren. Die Position ist somit konstant und die Zeit variabel.

Auch kann die Einfrierung entlang nur einer der Richtungen in der Displaywand-Ebene hilfreich sein. Die Displaywand ist etwas höher als die Größe eines durchschnittlichen Menschen, wodurch die Arbeit im höchsten Bereich ziemlich erschwert ist. Die Arbeit in eher niedrigen Bereichen kann ebenfalls physisch anstrengend sein. Die Einfrierung der y-Achse kann in solchen Fällen Hilfe bieten. Alternativ dazu ist ein Konzept der perspektivischen Abbildung des Auswahlbereichs an die Displaywand, bei der die Neigung des Mobilgeräts verwendet wird. Der Vorteil hier besteht in der Natürlichkeit und Einfachheit: Für die Auswahl extremer Displaywandbereiche muss lediglich das Mobilgerät geneigt werden und anstrengende Dehnungsbewegungen der Arme sind obsolet.

Die Einfrierung der z-Bewegungsrichtung stellt eine umgekehrte Situation dar. Bei einem ausgewählten und festgesetzten Zeitzustand kann man die räumlichen Dimensionen explorieren. Somit ist die Zeit konstant und die Position variabel. Ein weiterer Anwendungsfall der z-Einfrierung ist die Auswahl des Bezugsintervalls. Wenn die Änderungen zu einem bestimmten Intervall angezeigt werden sollen, friert man die z-Achse in diesem Intervall ein, bewegt sich in das Wunschintervall und hebt die Einfrierung auf. Das eingefrorene Intervall dient somit als das vorher angesehene. Auf diese Weise können beliebige Intervalle verglichen werden.

Durch die Einfrierung entlang aller drei Bewegungsrichtungen bekommt man eine bezüglich der Position und der Zeit feste Ansicht des Teilgraphen, die für nähere Untersuchungen verwendet werden kann, zum Beispiel in der hinteren Zone des Raums, wo keine direkten Interaktionen mit der Displaywand stattfinden. Zeit und Position bleiben konstant.

3.3.2 Einordnung im interaktiven Raum

Im interaktiven Raum muss die Funktion der Einfrierung durch eine Interaktion aufgerufen werden. Zudem muss die Aktivierung der Markierung für den Nutzer sichtbar sein. Dafür kann man beispielsweise die Checkboxen einsetzen. Alternativ wäre eine räumliche Geste, beispielsweise eine Schüt-

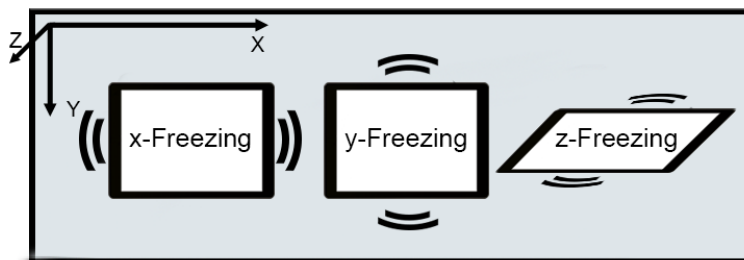


Abbildung 3.9: Einfrierung der Ansicht entlang der räumlichen Achsen durch Schütteln

telgeste als Trigger. Diese ist vor allem bei der Aufhebung der Einfrierung intuitiv. Man schüttelt die eingefrorene Ansicht mit statischer Visualisierung, um dynamische Updates zu aktivieren. Für die Erkennung der zu einfrierenden Achse kann zusätzlich die Orientierung des Mobilgeräts verwendet werden. In Abbildung 3.9 ist Einsatz der Schüttelgeste in Kombination mit Neigung zu sehen. Da während der Durchführung der Geste der Auswahlbereich beeinflusst wird, werden für diese Option spezielle Techniken gebraucht, die die Bewegungen des Geräts ab einer bestimmten Geschwindigkeit nicht als Navigation im Graphen wahrnehmen.

Ist die Einfrierung aktiv und hat der Nutzer die Stelle, an der die Aktivierung erfolgte, verlassen, um beispielsweise den Zustand näher zu untersuchen, so muss das System eine einfache Reidentifizierung der eingefrorenen Stelle gewährleisten. Dies kann durch visuelles Feedback des Mobilgerät-Interface erfolgen. Als Feedback kann beispielsweise eine Miniaturansicht der Achsen verwendet werden, in der die aktuelle Position dargestellt wird.

Mit dem Konzept der Einfrierung lassen sich die Interaktionen im interaktiven Raum komfortabler ausführen. Die nutzerspezifischen Präferenzen werden bei der Navigation beachtet.

3.4 Multiple Ansichten

Die Graph- und die Timeline-Ansicht sind zwei verschiedene Ansichten der Daten des Fokusbereichs. Denkbar ist zudem eine Analyse-Ansicht für diejenigen Daten-Abfragen, die sich nicht visuell darstellen lassen. Diese Ansicht kann unter anderem Statistiken zu dem aktuellen Intervall beinhalten. Dazu gehören beispielsweise die Geschwindigkeit, die Änderungsrate, der Wachstumswert usw. Die Anwendung soll über eine Funktionalität zum Wechseln zwischen den Ansichten verfügen. Eine Möglichkeit ist, die Steuerelemente, beispielsweise Buttons, auf dem Interface zu platzieren und diese mit den Touchgesten zu bedienen. Da die Steuerelemente einen zusätzlichen Platz auf dem relativ kleinen Mobilgerät-Display brauchen, ist der Einsatz von indirekten Touch-Gesten platzsparender. Für den Wechsel zwischen multiplen Ansichten kann eine horizontale Drag-Geste eingesetzt werden. Sie unterstützt das mentale Bild der nebeneinander liegenden Ansichten und braucht keine Steuerelemente auf dem Interface. Eine dritte Option ist die Verwendung räumlicher Gesten. Die Änderung der Position des Geräts, beispielsweise eine Neigung (Abbildung 3.10 a), kann verwendet werden, wenn eine Umschaltung benötigt wird. Da die Haltung des Mobilgeräts während der Durchführung bestimmter Aufgaben nutzerspezifisch ist, kann die Geste nutzerbezogen erfolgen. Erfolgt eine schnelle Änderung der Orientierung um einen bestimmten Grad α , so wird die Geste als solche wahrgenommen. Die absolute Version der Geste, bei der die Interaktion durch eine bestimmte Orientierung im statischen Bezugssystem aufgerufen wird, kann dagegen leichter durch unbeabsichtigte

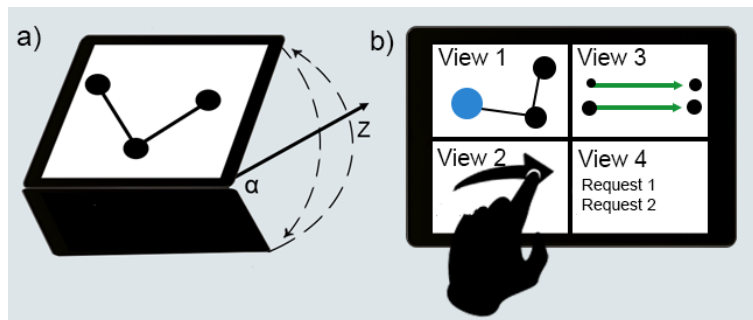


Abbildung 3.10: a) Neigungsgeste zum Umschalten zwischen Ansichten. b) Umschalten zwischen Ansichten im Multifenster durch eine Drag-Geste.

Bewegungen als Geste interpretiert werden.

Während der aktiven Exploration kann eine Speicherung bestimmter Ausschnitte nützlich sein, beispielsweise wenn man diese später näher untersuchen möchte. Zum Speichern könnte Wischen mit der Handkante über das Display eingesetzt werden. Diese Geste wird in gängigen Samsung-Mobilgeräten für die Erstellung von Screenshots eingesetzt. Die Speicherung des Ausschnitts kann als Analogie zu der Erstellung von Screenshots angesehen werden. Jeder Ausschnitt soll Informationen zu der räumlichen und der zeitlichen Position und gegebenenfalls zu der Zoomstufe enthalten. Für die Übersicht der gespeicherten Ausschnitte kann ein Multifenster-Interface behilflich sein. In Abbildung 3.10 b) ist ein Multifenster-Interface mit 4 Ausschnitten zu sehen. Die Ansichten der Ausschnitte werden durch eine Drag-Geste gesteuert. Eine alternative Option mit räumlicher Geste benötigt Selektion des gewünschten Ausschnitts und ist im Vergleich zur Drag-Geste in 2 Schritten ausführbar. Gleichzeitig kann der Einsatz der Selektion für andere Aufgaben mit dem Ausschnitt verwendet werden. Analog zu Datentransferrealisierung durch Werfgeste bei Kurdyukova et al. [KRA12] ist der Transfer des selektierten Ausschnitts an andere Nutzer oder an die Displaywand denkbar.

Verallgemeinernd ausgedrückt können Multifenster mit gespeicherten Ansichten als eine Toolbox für die Kombination mit anderen Filtern und Einstellungen verwendet werden. Verschiedene Ansichten auf selektierte räumlich-temporale Ausschnitte ermöglichen eine vertiefte Analyse.

4 Umsetzung

Dieses Kapitel beschäftigt sich mit der Implementierungsphase. Im Abschnitt 4.1 wird ein Überblick über das System und das bestehende Projekt *MobileLens* gegeben. Abschnitt 4.2 beschäftigt sich mit Implementierung der Datenstruktur, des interaktiven Raums und der Visualisierung der Änderungen.

4.1 System-Überblick

Die Umsetzung der Konzepte erfolgte in einer existierenden Umgebung. Im Folgenden werden die Hardware-Elemente des Systems, zu denen die Displaywand, die Mobilgeräte und die Kameras gehören, sowie das Trackingsystem und die Anwendung *MobileLens* beschrieben.

4.1.1 Verwendete Hardware

Die interaktive Displaywand ist ein Produkt des Softwareentwicklers und -herstellers MultiTaction [Mul]. Die Displaywand der Fakultät Informatik gehört zur Lobby Solution. Sie ist für einen größeren Raum wie das Labor der Professur Multimedia-Technologie gestaltet. Die einzelnen Bestandteile sind 12 MultiTaction Display-Zellen im Landscape-Modus: 3 Zeilen à 4 Spalten. Die Bildschirm-Maße sind 5170mm x 2620mm. Die Arten von Toucheingaben sind Einzelfinger, mehrere Finger, Einzelhand, mehrere Hände. Durch die unbegrenzte Anzahl an simultan arbeitenden Nutzern und Toucheingaben eignet sich die Displaywand gut für Nutzerkollaborationen.

Die im existierenden Projekt verwendeten Mobilgeräte sind die von Google entwickelten und von Asus hergestellten Nexus 7 Version 2013 mit einem Android Betriebssystem [Nex]. Die Maße betragen 200 x 114 x 8.7 mm, die Auflösung des Displays 1200 x 1920px. Multitouch-Eingaben bis zu 10 Fingern sind erkennbar. Zu den eingebauten Sensoren gehören Gyroskop, Beschleunigungssensor, Näherungssensor und Kompass.

Für das Tracking von Positionen der Mobilgeräte werden zwölf Flex 13 Kameras mit der Auflösung bla und der Framerate bla verwendet.

4.1.2 Tracking

Die Tracking Engine gehört zu der Middleware des Systems. Die für das Tracking verwendete Anwendung nennt sich Motive und ist ein OptiTrack-Produkt. Es ist eine für die optischen Bewegungsverfolgungen mit Echtzeit- und Multithreads-Unterstützung entwickelte Plattform. Durch die Kameras wird eine auf dem Mobilgerät befestigte Konstruktion aus retroreflektierenden Kugeln, die als Indikatoren dienen, mit relativ hoher Präzision getrackt. Das Tracking erfolgt durch die Anwendung Motive in sechs Freiheitsgraden: Positionswechsel nach vorne/hinten, oben/unten, links/rechts sowie Rotationen über drei Achsen, die sich Pitch, Yaw und Roll nennen. Die Tracking-Daten mit Position

des Mobilgeräts werden mithilfe der NatNet SDK-basierten Anwendung NAtNetBridge von einem Server an das Client Displaywand-Rechner gesendet. Der Server läuft dabei auf einem separaten Rechner.

4.1.3 MobileLens

Die Anwendung MobileLens ist ein Projekt von Interactiv Media Lab Dresden zur Realisierung von Linsenfunktionalitäten. Das Projekt ist in der universellen höheren Programmiersprache Python 2.7 mit Verwendung einiger Paketen aus dem Python Package Index und Bibliotheken NetworkX und Libavg umgesetzt. NetworkX [Net] ist ein frei verfügbares Python-Paket für die Erstellung, Speicherung, Manipulation und Untersuchungen der Struktur, Entwicklungen und Funktionalitäten komplexer Netzwerke. Libavg [Lib] ist eine durch Interactiv Media Lab Dresden entwickelte high-level Plattform für die Entwicklung von Medienanwendungen. Sie unterstützt eine Vielfalt von Display-Elementen wie Bilder, Text, Vektorgraphiken sowie die Erkennung von Handle-Events, unter anderem für Multitouch-Gesten.

Das Projekt MobileLens ist als eine Client-Server Architektur realisiert. Die Abbildung 4.1 stellt die für das Konzept der Tangible Lens wichtigen Elemente des Projekts dar. Die Graph-Daten der Architektur sind im XML-basierten GraphML-Format gespeichert. GraphML-Format unterstützt gerichtete und ungerichtete, hierarchische Graphen und Hypergraphen sowie die Definition von Anwendungs-basierten Attributen für Knoten und Kanten. Durch den Graph-Provider werden die Dateien eingelesen und als projektinterne Datenstrukturen gespeichert. Die Komponente Graph repräsentiert die Graph-Datenstruktur und ihre Ansicht. Das Graph-Objekt erweitert das NetworkX-Graph-Objekt um die gesonderte Behandlung von Knoten und Kanten, wodurch die Manipulationen ohne Konsistenzverlust ermöglicht werden.

Die Komponente Tracking ist für die Kommunikation zwischen dem Tracking-Server und der Displaywand sowie der Displaywand und dem Client Mobilgerät zuständig. Sie identifiziert das Mobilgerät, empfängt durch OptiTrack seine Tracking-Daten und verarbeitet diese. In Debug-Utils wird eine Simulation der Kommunikation nachgestellt, wodurch das Testen der Funktionalitäten ohne direkte Nutzung der Displaywand und des Mobilgeräts möglich ist.

In der Komponente Device werden die Funktionalitäten des Mobilgeräts implementiert. Die Graph-

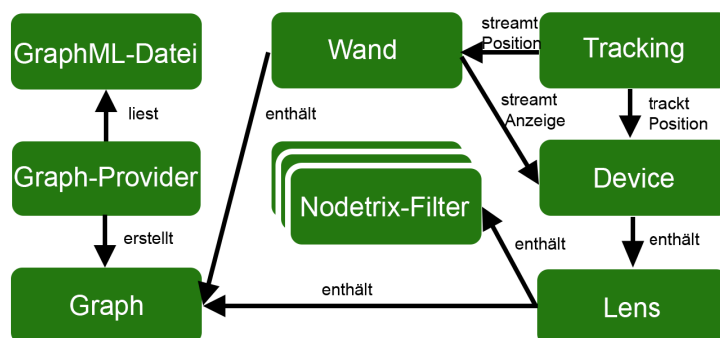


Abbildung 4.1: Elemente der Anwendung MobileLens für die Umsetzung des Konzepts Tangible Lens.

Device-Klasse erweitert die Device-Klasse um die Anzeige des Graphen sowie die Navigation entlang dieses Graphen. Graph-Lens-Device-Klasse ist eine Erweiterung, in der die Linsenfunktionalitäten implementiert sind und verschiedene Filter hinzugefügt werden können. Im Projekt ist es möglich, die Linsen auch ohne Mobilgeräte direkt an der Displaywand zu erstellen. Die Linse und die darin enthaltenen Filter erlauben die Identifizierung der Interaktionen und die Umsetzung der Interaktionslogik.

Die Architektur des Projekts in der Präsentationsschicht basiert auf dem Model-View-Controller Ansatz. MVC ist ein weit verbreitetes Interaktionspattern bei den Anwendungen, in denen die Nutzer-Software-Interaktion eine große Rolle spielt [LR06]. Das Model dient der Speicherung bestimmter Daten, der Kommunikation mit externen Datenquellen, der Gewährleistung des Zugriffs auf die Zustandsdaten usw. Die View visualisiert die Daten des Models. Nachteil des MVC-Ansatzes ist ein Performance-Verlust durch erhöhten Kommunikationsaufwand. Gleichzeitig wird die Komplexität und die Abhängigkeit innerhalb des Systems reduziert, was für ein Projekt, welches von relativ vielen Forschern entwickelt wird, nützlich ist. Der Austausch und die Erweiterung von Komponenten ist durch eine gute Kapselung der Schichten ermöglicht.

Die Informationen über die Änderungen der Zustände der Objekte spielen in MVC-Anwendungen die zentrale Rolle. Nach dem Prinzip der einzigen Verantwortung [LR06] sollte möglichst vermieden werden, dass sich die betroffenen Objekte, beispielsweise Beobachter und Beobachtetes, gegenseitig kennenlernen. Für diese Fälle wird im Projekt das Design-Patter Observer eingesetzt. Durch den Einsatz von Callbacks melden die beobachteten Objekte Änderungen ihrer Eigenschaften an die Beobachter. So werden im Projekt beispielsweise die Views der Objekte bei Änderungen ihrer beobachteten Models geändert.

Das Projekt befindet sich im Entwicklungsstand und verfügt über eine Reihe von bereits implementierten Techniken. Eine Selektion wird durch die Erkennung der Touch-Gesten an einzelnen Knoten sowie ein Lasso-Werkzeug für die Knotenmengen umgesetzt. Das Konzept einer magischen Linse ist in Ausprägungen des klassischen Linsenobjekts mit einer eingestellten Form, einer Bodylens sowie einer Tangible Lens auf dem Mobilgerät implementiert. Zudem verfügt das Projekt über mehrere Filter wie NodeTrix, FishEye, Neighbors usw. Die Idee der räumlichen geschichteten Interaktion findet bei der Umsetzung von Zooming statt Die Position des Nutzers auf der z-Achse bestimmt die Größe des im Mobilgerät angezeigten Ausschnitts. Eine Benutzeroberfläche für die Navigation durch die Funktionalitäten des Systems auf dem Mobilgerät ist in der Entwicklung.

4.2 Implementierung

Im Rahmen einer prototypischen Implementierung wurden die ausgewählten Teile der Konzepte für die Abbildung der Zeit-Variable, die Visualisierung der Änderungen zwischen Zuständen mehrerer Intervalle und Darstellung der Entwicklung von Knoten auf einem Intervall umgesetzt. Somit wurde eine Basis für die Visualisierung und die Exploraion dynamischer Graphen geschaffen. Durch eine Bewegung entlang einer z-Achse soll es dem Nutzer ermöglicht werden, sich die Zustände des Graphen anzuschauen. Zudem sollen die Änderungen zwischen den Zuständen auf geeignete Weise visualisiert werden. In Abbildung 4.2 sind die für diese Ziele neu implementierten Elemente dargestellt. In diesem Abschnitt werden wichtige Aspekte der Umsetzung dieser Elemente dargelegt. Abschnitt 4.2.1 beschreibt die Datenstruktur für die dynamischen Graphen und ein konkretes Beispiel, mit dem

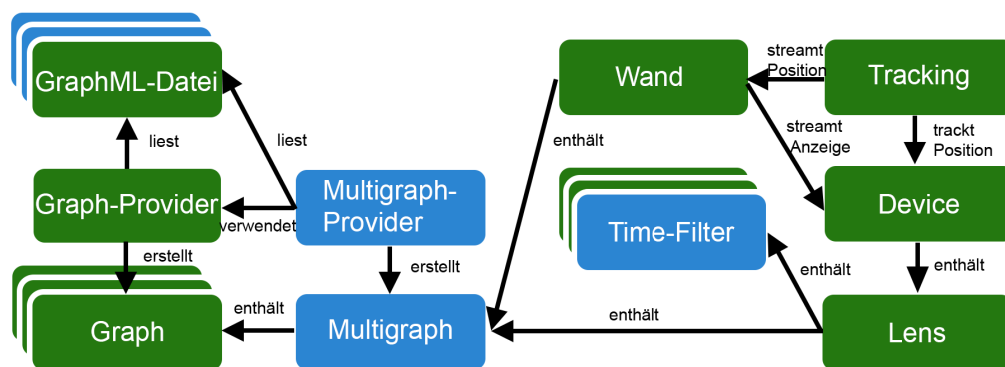


Abbildung 4.2: Vervollständigung der Anwendung MobileLens.

gearbeitet wurde. Im Abschnitt 4.2.2 wird erläutert, wie die Interaktionen für die Ansicht der Zustände umgesetzt wurden. Die Implementierung der Timeline-Ansicht wird im Abschnitt 4.3.4 dargestellt.

4.2.1 Datenstruktur

Die existierende Graph-Datenstruktur des Projekts ist für dynamische Graphen nicht geeignet und erfordert eine Erweiterung. In Graph-Datenbanken werden die Zustände der Graphen oft in separaten Dateien gespeichert. Ähnlich werden einzelne Dateien in einem den Graph repräsentierenden Verzeichnis angelegt. Im Folgenden werden Informationen beschrieben, die dafür in der Datenstruktur benötigt werden:

- Einzelne Graph-Zustände des dynamischen Graphen
- Einzelne Knoten und Kanten mit den sich über die Zeit ändernden Eigenschaften
- Eigenschaften mit den dazugehörigen Zeitänderungen
- Aufteilung der Änderungen in Intervalle

Zur Realisierung der Abbildung dieser Informationen wurden im Rahmen dieser Bachelorarbeit die Klassen *Multigraph* und *Multigraph_Provider* implementiert. *Multigraph_Provider* liest alle Graph-Zustände des dynamischen Graphen und definiert die Intervalle. Zur Zeit wird die Intervallaufteilung als statische Variante aus den Konzepten implementiert. Für den ganzen Graphen wird eine feste Anzahl an Intervallen abhängig von der Anzahl der einzelnen Graph-Zustände festgelegt. Für den Fall mit bis zu 6 Zuständen wird jeweils ein Zustand pro Intervall gespeichert, für den Fall mit von 7 bis zu 14 Zuständen jeweils 2 Zustände pro Intervall und für den Fall von 15 bis zu 21 Zuständen jeweils 3 Zustände pro Intervall. Die höchstmögliche Anzahl an Intervallen ist somit 7. Bei dieser Aufteilung erfolgt keine Anpassung an den ausgewählten Teilgraphen. Die Anzahl an Intervallen und die Anzahl an Zuständen pro Intervall bleiben für den ganzen Graphen konstant.

Die Klasse *Multigraph* verfügt über einige für verschiedene Anfragen an die Daten nützliche Attribute. Das Attribut *intervalgraphs* ist eine Dictionary-Datenstruktur. Durch *Multigraph_Provider* werden alle eingelesenen Graph-Zustände als geordnete Listen-Werte mit Intervallen als Schlüssel gespeichert. Diese Art der Speicherung ist nützlich, wenn genauere Kenntnisse über die einzelnen Zwischenzustände in einem Intervall benötigt werden. Dies ist für die Timeline-Ansicht der Fall, wo

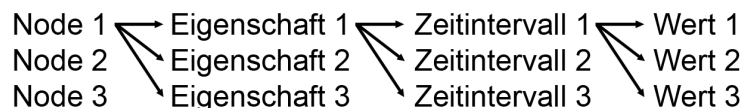


Abbildung 4.3: Aufbau der Datenstruktur.

für jeden Knoten des ausgewählten Teilgraphen die Zwischenzustände auf einem Intervall an einer Timeline angezeigt werden. Die Erstellungszeiten der Zustände können in absoluten Zeitangaben, z.B. durch Date-Objekte aus *datetime*, angegeben werden oder auch in relativen Zeittakten. In der Implementierung werden die relativen Zeittakte verwendet.

Für die Graph-Ansicht benötigt man jeweils nur die neuesten Zustände auf den Intervallen. Das Attribut *graphs* ist eine Dictionary-Datenstruktur mit Intervallnummern als Schlüssel und den neuesten Graph-Zuständen auf diesen Intervallen als Werte. Diese Struktur ist besonders nützlich, wenn der anzuzeigende Graph-Zustand auf dem Mobilgerät geändert werden soll, da die für die Anzeige irrelevanten Zustände ausgelassen werden, was bei ständigen Updates der Filter-Anzeige zu einer Performance-Verbesserung führen soll.

Die Verwendung des Attributs *graphs* würde in der Visualisierung der Änderungen auf einem Intervall dazu führen, dass bei jedem Update für jede zu visualisierende Eigenschaft auf 2 Zustände zugegriffen wird, um dort nach Knoten oder Kanten zu suchen. Das Attribut *dynamicalnodes* soll den Zugriff vereinfachen. Dazu wird für jeden Knoten des dynamischen Graphen jede Eigenschaft gespeichert, für jede Eigenschaft wird zu jedem Zeitintervall der Wert gespeichert. In Abbildung 4.3 ist der Aufbau dieser Datenstruktur zu sehen. Durch eine Verschachtelung von 4 Wörterbüchern wird ein leichter Zugriff auf Zustände der Knoten-Attribute gewährleistet. Der gleiche Aufbau wird für die Kanten in *dynamicaledges* verwendet. Der Zugriff auf einen Wert einer Eigenschaft eines Knotens und einer Kante in einem Intervall erfolgt auf folgende Weise:

```

1 value = multigraph.dynamical_nodes[node_id][property][interval]
2

```

Die allgemeine Entscheidung, die Intervallaufteilung in der Datenstrukturebene vorzunehmen, ist dadurch zu begründen, dass die Datenzugriffe vereinfacht werden. Für die Aufteilung erst in der Linse wäre eine zusätzliche Funktion, die bei jedem Zugriff jeweils den richtigen Zustand für einen Intervall ausrechnet, nötig. Gleichzeitig könnten die Steuerungsmöglichkeiten für die Aufteilung als Teil des Filter-Interface implementiert werden. Für die Steuerung in der aktuellen Implementierung müsste die Datenstruktur neu erstellt werden. Dieser Nachteil ist in Kauf zu nehmen, da die Änderungen der Einstellungen eher die Einzelereignisse sind und die Zugriffe auf die Daten bei jedem Update-Takt erfolgen.

Als konkrete Datenstruktur wurde ein soziales Netzwerk genommen. Durch eine Funktion mit Zufallselementen und zum Teil manuelle Anpassungen wurden 18 verschiedene Graph-Zustände erstellt, so dass das Netzwerk insgesamt zu einer Vergrößerung tendiert. Jeweils 3 Zustände wurden zu einem Intervall zusammengefasst, was insgesamt 6 Intervalle ergab. Eine Reihe der visuellen Variablen wurde bei der Erstellung des Netzwerks verwendet:

- Die Knoten repräsentieren Personen.
- Die Kanten repräsentieren Freundschaften zwischen Personen.

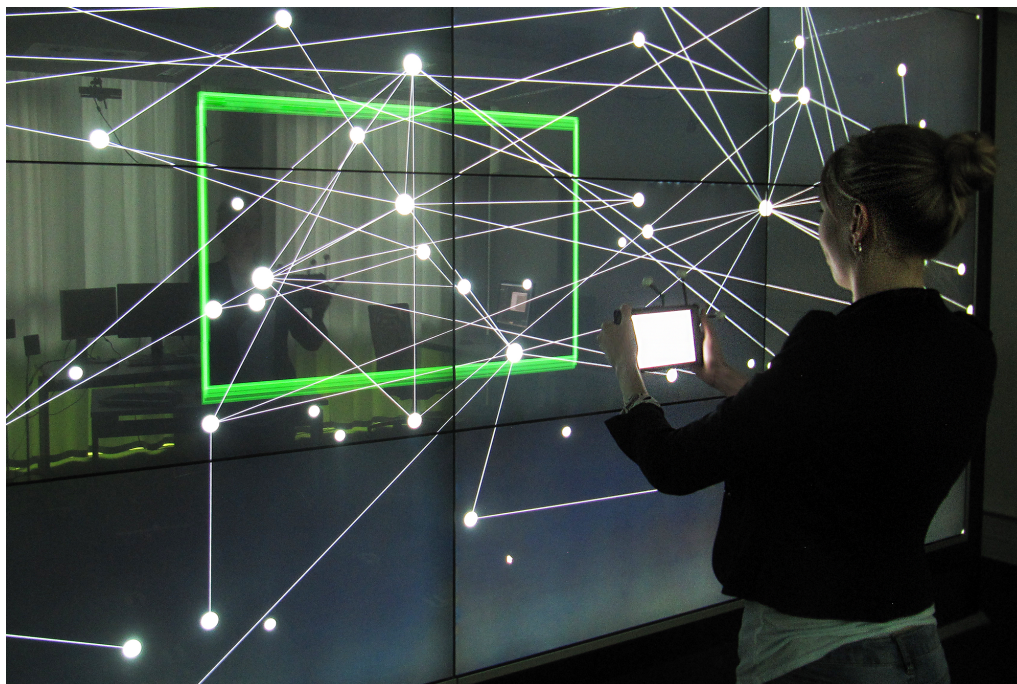


Abbildung 4.4: Der Nutzer im System mit kombinierter Anwendung Displaywand und Mobilgerät.

- Die Knotengröße repräsentiert die Anzahl der Freundschaften der Person (Nachbargrad). Insgesamt werden 3 verschiedene Größen verwendet. Diese beispielhafte ordinale Aufteilung wurde für eine bessere Wahrnehmung der Unterschiede vorgenommen.
- Die Position der Knoten repräsentiert die Position der Person.

4.2.2 Dynamischer Graph im interaktiven Raum

Für die Abbildung der erweiterten Datenstruktur im interaktiven Raum erfolgte die Aufteilung der z-Achse in die der Datenstruktur entsprechenden Intervalle. An dieser Stelle wurde eine Begrenzung des maximalen Wertes der z-Achse auf 190 vorgenommen. Dieser Wert erschien bei mehreren Testdurchläufen aus subjektiver Sicht als der weiteste, bei dem der Kontext an der Displaywand noch gut wahrnehmbar ist und eine präzise Navigation entlang der Displaywand-Ebene ohne größere Störungen erfolgt.

In der gegebenen Implementierung blieb die Ansicht des Graphen im Mobilgerät statisch, lediglich die Position des Auswahlbereichs änderte sich mit der Bewegung des Geräts. Für die Änderungen der Ansicht des Graphen wurde in *GraphDeviceView* die Funktion *update_view()* implementiert. Sie berechnet, welches Intervall dem gegebenen z-Wert entspricht und, falls die aktuelle Ansicht nicht im entsprechenden Intervall liegt, wird diese gelöscht und eine neue erstellt. In Abbildung 4.4 ist die Navigation dargestellt. Die Funktion *update_view()* wird nur für den Fall aufgerufen, dass der Graph eine dynamische Natur hat und ein Wechsel zwischen den Intervallen stattfand. Somit werden andere Funktionalitäten von *GraphDevice* bei den statischen Graphen nicht beeinträchtigt und die Updates erfolgen nur, wenn dies nötig ist.

In die Implementierung wurde das Konzept der Einfrierung eingegliedert. Die Einfrierung der räumlichen sowie der physischen Navigation erfolgt durch die Markierung der sich im Interface des



Abbildung 4.5: Umschaltung zwischen dem zeitlichen (neuestes Intervall) und dem räumlichen (aktuelles Intervall) Kontext.

Geräts befindenden Checkboxen. Das Konzept wurde in der Klasse *GraphLensDeviceView* für die Position des Mobilgeräts implementiert und kann bei beliebigen Linsen angewendet werden.

Als Default-Einstellung wird an der Displaywand der letzte, 18. Zustand des Graphen angezeigt. Möchte man den Zustand zum aktuell im Mobilgerät angezeigten wechseln, so verwendet man eine Drag-Geste in Richtung der Displaywand, die als solche aufgefasst wird, wenn die Länge der Berührung größer als $2/3$ der Displayhöhe in horizontaler Orientierung beträgt. Beim darauf folgenden Wechsel des Intervalls wechselt die Anzeige an der Displaywand nicht automatisch. Eine erneute Geste ist notwendig. Diese Entscheidung wurde getroffen, weil dem Nutzer die Möglichkeit der Kombination jedes beliebigen Zustands an der Displaywand mit jedem beliebigen Zustand im Mobilgerät ohne zusätzliche Steuerungselemente im Interface zur Verfügung steht. Analog dazu bewirkt eine gegenseitige Geste, dass der neueste Zustand des Graphen an der Displaywand abgebildet wird. Die Abbildung 4.5 repräsentiert den Vorgang. Damit der Nutzer jederzeit erkennt, welcher Zustand an der Displaywand abgebildet ist, wird in der linken oberen Ecke eine kleine Markierung gezeigt. Die Änderung des Zustands der Displaywand erfolgt in der Implementierung durch die Änderung der Variable *maingraph* in der Multigraph-Struktur.

4.2.3 Visualisierung der Änderungen zwischen den Intervallen

Für die Visualisierung der Änderungen wurde eine neue Filter-Klasse *TimeFilter* implementiert. Die Änderungen werden in zwei Gruppen unterteilt. Zu den absoluten Änderungen gehören das Entstehen und das Löschen eines Knotens oder einer Kante. Alle anderen Änderungen sind relativ und beziehen sich auf das vorher angeschaute Intervall. Bewegt man sich nach vorne, ist das vorher angeschaute Intervall das vorherige, bei der Bewegung nach hinten ist es das darauf folgende. Durch die Nutzung von z-Einfrierung ist die Auswahl eines beliebigen Bezugsintervalls möglich. Dazu muss im gewünschten Bezugsintervall eingefroren werden. Die Änderungen werden im Intervall, in dem die Einfrierung aufgehoben wird, visualisiert.

Damit man sich in den Intervallen besser orientieren kann, wurde in *time-filter-view* eine verkleinerte Timeline der z-Achse als Navigationshilfe für die zusätzliche Unterstützung des zeitlichen Kontextes implementiert. Ein sich entlang der Timeline bewegender Pfeil symbolisiert die Position des Nutzers in Intervallen sowie die Richtung des vorher angesehenen Intervalls. Somit kann der Nut-

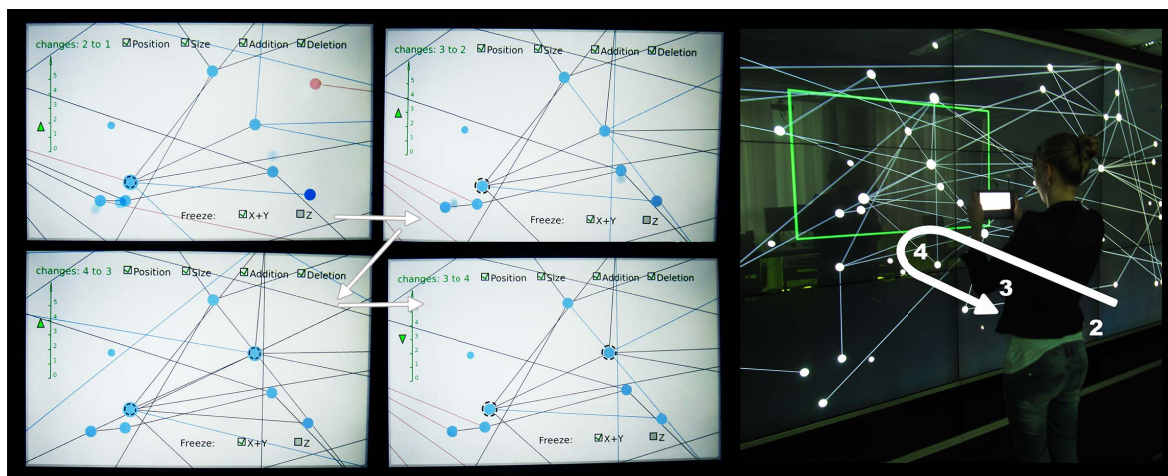


Abbildung 4.6: Visualisierung des Intervalls 3 abhängig vom gewählten Bezugsintervall (2 oder 4).

zer erkennen, die Änderungen zu welchem Intervall gerade angezeigt werden. Zusätzlich wurde eine kleine textuelle Anzeige des aktuellen und des Bezugsintervalls implementiert. Die Visualisierung der Timeline wurde durch eine eigene Klasse implementiert, um so eine Mehrfachverwendung zu ermöglichen. Zu den Einstellungsparametern zählen die Startposition, die Länge, die horizontale oder vertikale Richtung, die Anzahl der Intervalle, die Beschriftungen sowie weitere visuelle Eigenschaften wie die Farbe oder die Liniendicke.

Die Funktion *show_changes()* durchläuft alle sich im Ausschnitt Knoten und überprüft, ob sie neu entstanden sind, und ob sich ihre Eigenschaften geändert haben. Zusätzlich wird geprüft, ob im Intervall die Knoten gelöscht wurden. Sodann überprüft die Funktion alle zugehörigen Kanten auf das gleiche. In einer Einstellungsdatei *time_viz_config* kann man auswählen, mit welchen visuellen Variablen welche Eigenschaft kodiert wird. Je nachdem, welche dieser Eigenschaften bereits für die Visualisierung des Graphen eingesetzt wurden, kann man die Einstellungen anpassen. Die Visualisierung der Änderungen wurde nicht an dem Multigraph-Objekt durchgeführt sondern an seiner Kopie. So bleibt der Kontext an der Displaywand ungestört von der Visualisierung im Filter. Zudem wurden die Manipulationen an der Position der Knoten für alle Zustände außer des neuesten deaktiviert. Die Konzept der aktiven Knotenmenge wurde in der einfachen Variante umgesetzt: Verfolgt werden stets die Änderungen von allen Knoten, die sich gerade innerhalb des Filters befinden. Für das Entstehen und das Löschen wurden als Default-Einstellung die Farbtonkodierungen in Blau und Rot implementiert. Die Änderungen werden stets im Bezug auf das vorherige oder das darauf folgende Intervall visualisiert. Bei der Bewegung nach vorne werden die im aktuellen Intervall gelöschten und geborenen Knoten und Kanten hervorgehoben. Da die gelöschten Knoten und Kanten nicht mehr präsent sind, wird bei ihnen zusätzlich die Transparenz geändert. Bei der Rückwärtsbewegung werden die im vorher angeschauten, zeitlich darauf folgenden Intervall gelöschten und hinzugefügten Knoten und Kanten, die im aktuellen Intervall noch existieren, gefärbt. Da die im späteren Intervall hinzugefügten Knoten noch nicht im aktuellen existieren, wird ihre Transparenz geändert. Die *time_viz_config* erlaubt, die Farbmarkierungen als Glyphen innerhalb des Knotens darzustellen, was bei dem Einsatz des Farbtons für andere Attribute nützlich sein kann. Anstatt des Farbtons bei dem Löschvorgang kann man außerdem die Transparenz einsetzen und bei dem Geburtvorgang die Größe für die Knoten und die Dicke für die Kanten.

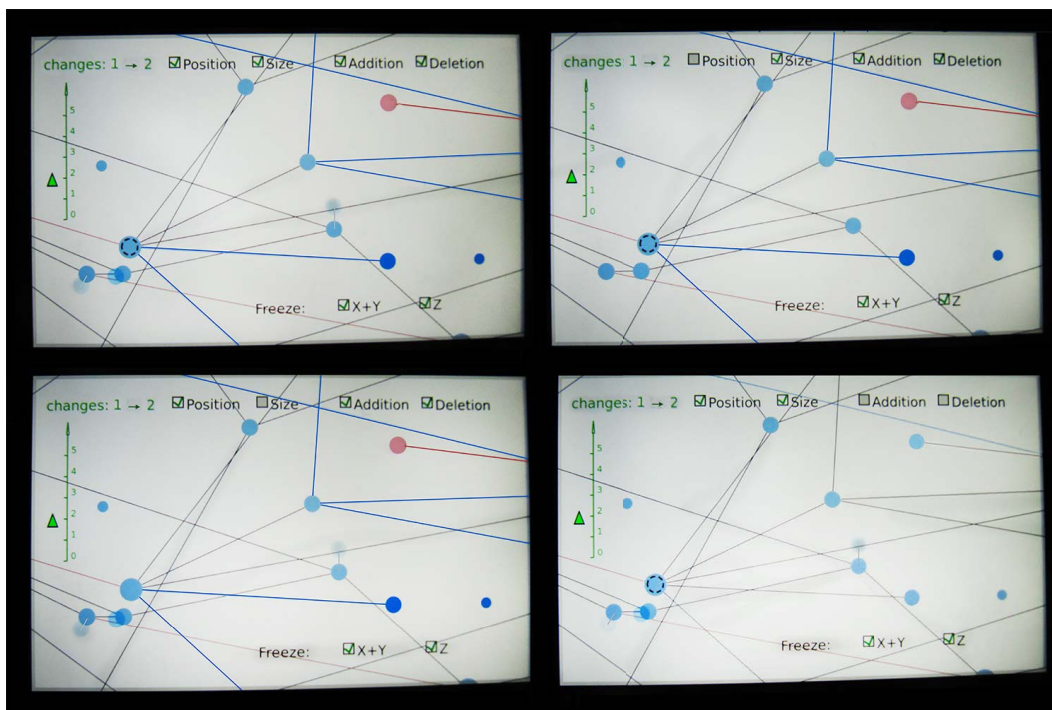


Abbildung 4.7: Nutzersteuerung der Visualisierung der Änderungen durch Checkboxes.

Die Visualisierung der Größenänderung für die Knoten wurde durch einen nicht gefüllten Kreis der Knotengröße mit gestricheltem Rand umgesetzt. Die Farbe des Randes kann schwarz, weiß oder grau sein, abhängig von der Knoten- oder der Hintergrundfarbe.

Für die Positionsänderung wird als Default-Einstellung Transparenz eingesetzt. Hat sich der Knoten bewegt, wird sein vorheriger Zustand transparenter dargestellt. Damit man bei mehreren Positionsänderungen von mehreren Knoten unterscheiden kann, welche vorherigen Zustände zu welchen Knoten gehören, werden sie jeweils durch Kanten verbunden. Damit diese Kanten nicht mit den eigentlichen Kanten des Graphen verwechselt werden, kann man in der Datei *time_viz_config* ihr Aussehen einstellen. In der Default-Einstellung sind die Kanten grau und haben eine geringe Dicke. Für die Fälle, in denen sich eine andere Eigenschaft des Knotens neben der Position geändert hat, in dieser Visualisierung ist es die Größe, wird die Änderung an der alten Position abgebildet. In Abbildung 4.6 folgendes Anwendungsszenario für den Durchlauf der Intervalle zu sehen. Der Nutzer befindet sich im Intervall 2 und wechselt zum Intervall 3. Die Visualisierung der Änderungen bezieht sich somit auf Intervall 2. Nun wechselt der Nutzer zum Intervall 4 und abschließend wieder zum Intervall 3. Die Visualisierung für dieses Intervall hat sich geändert und bezieht sich nun auf das Intervall 4.

Als eine Visualisierungssteuerung wurden im Interface Checkboxes für jedes virtualisierbare Änderungsattribut implementiert. Somit kann der Nutzer nur die für ihn interessanten Attribute visualisieren lassen. Die Abbildung 4.7 präsentiert die Auswirkungen der Steuerung auf die Visualisierung.

Der Graphzustand auf dem Mobilgerät unterscheidet sich durch die Bewegung entlang der z-Achse von dem Graph-Zustand an der Displaywand. Dies kann zur Erschwerung der Orientierung an der Displaywand führen. Eine Abbildung der Position des Mobilgeräts an der Displaywand soll dies vermeiden. Die Abbildung wurde als ein nicht gefülltes grünes Rechteck mit den Maßen des im Mobilgerät angezeigten Ausschnitts implementiert (Abbildung 4.5).

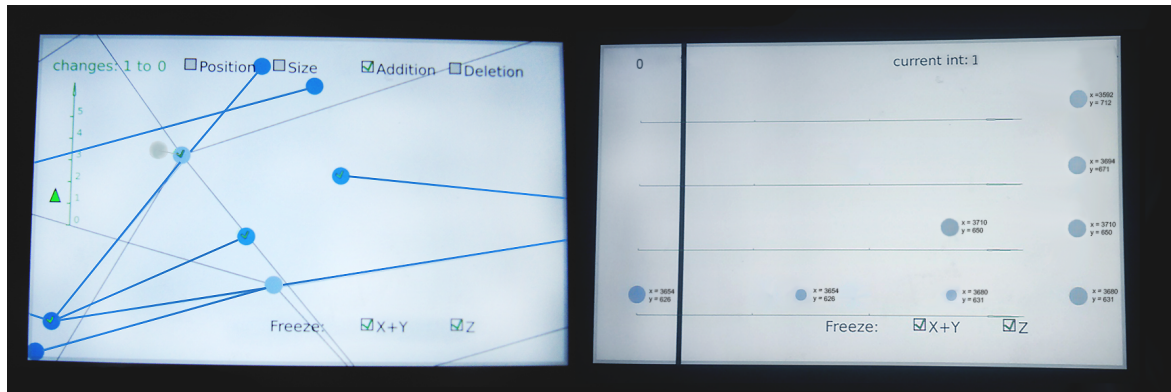


Abbildung 4.8: Wechsel zwischen der Graph- und der Timeline-Ansicht.

4.2.4 Timeline-Ansicht

In *TimeFilter* wurde eine Timeline-Ansicht zusätzlich zu der Graph-Ansicht realisiert. Der Wechsel erfolgt durch eine Drag-Geste von links nach rechts. Durch eine schwarze vertikale Linie werden das vorherige und das aktuelle Intervalle abgetrennt. Bei dem vorherigen Intervall wird nur der Endzustand dargestellt, bei dem aktuellen alle Zwischenzustände.

Wenn keine Knoten durch Tap-Gesten selektiert sind, werden für alle Knoten des Fokusbereichs durch eigene Timelines die Umschaltung zu dieser Ansicht erstellt. Die Zwischenzustände, die bei der Graph-Ansicht nicht visualisiert werden, sind entlang der Timeline dargestellt. Durch die Position in Timeline lässt sich die wirkliche Position der Knoten nicht abbilden. Dazu wurde dieses Attribut in textueller Form implementiert. In Abbildung 4.8 sind Graph- und Timeline-Ansicht für einen Fokusbereich dargestellt.

5 Zusammenfassung, Bewertung und Ausblick

In diesem Kapitel werden die im Rahmen dieser Arbeit erarbeiteten Konzepte zusammengefasst und bewertet. Zudem wird ein Ausblick auf die zukünftigen Entwicklungen gegeben.

Für die Exploration der Graphzustände wurde das Konzept zur **Abbildung der zeitlichen Dimension** auf der zur Displaywand orthogonalen z-Achse entwickelt. Bei der Navigation werden die einzelnen Zustände des Fokusbereichs im Mobilgerät angezeigt, während sich an der Wand ein nutzerspezifischer Zustand auswählen lässt. Durch die Aufteilung der Graphzustände in Intervalle soll die Navigation und das Auswählen bei großer Menge der Zustände komfortabler erfolgen. Es wurden Optionen für die Aufteilung vorgeschlagen, die abhängig von der Datenstruktur entweder lediglich die Anzahl der Zustände oder zusätzlich die Anzahl der Änderungen im Zustand berücksichtigen. Die prototypische Implementierung erfolgte für die Option, in der die Aufteilung lediglich von der Anzahl der Zustände abhängig war. Ein weiterer Aspekt des Konzepts war die Verwendung verschiedener Bezugssysteme für die Exploration dynamischer Graphen mit zusätzlichen, anders geschichteten Daten. Während die primären, temporalen Schichten relativ zu der stationären Displaywand platziert werden, kann man die sekundären Schichten relativ zum Menschenkörper platzieren.

Das Konzept unterstützt die Interaktionsaufgaben Overview und Filter. Durch die Navigation wird eine Untermenge anhand des Attributs Zeit gewählt. Die gleichzeitige Darstellung des gesamten Informationsraums als Überblick an der Displaywand ist in der Konzeption nicht möglich. Das liegt daran, dass sich die Zeit als dritte Hauptvariable neben zwei Positionsvariablen auf der zweidimensionalen Oberfläche ohne Einsatz der Interaktion nicht darstellen lässt. Die Konzeption und die darauf basierende Implementierung berücksichtigen eine Möglichkeit der Umschaltung zwischen dem räumlichen und dem zeitlichen Kontext an der Displaywand. Dadurch werden die Nutzerpräferenzen beachtet. Gleichzeitig entsteht ein Problem mit der Nutzerkollaboration. Die Abbildung eines nutzerspezifischen Kontextes kann für andere Nutzer störend sein und ähnlich wie in der im related Work beschriebenen Anwendung mit der Videowand [RLE05] zu Irritationen führen. Die komfortable Nutzerkollaboration ist somit lediglich eingeschränkt möglich, und das Konzept ist eher für einen Einzelnutzer geeignet.

Durch die Implementierung ist deutlich geworden, dass die Aufteilung in Intervalle die Berücksichtigung der Anzahl der Graphzustände bei der Aufteilung in Intervalle nicht unbedingt nutzerspezifische Aspekte impliziert. Abhängig von der durchgeführten Nutzeraufgabe war jeweils eine andere Aufteilung sinnvoller. Bei einfacher Navigation in den Zuständen und gezielter Suche war aus subjektiver Sicht eine geringere Intervallbreite und somit eine höhere Anzahl an Intervallen sinnvoller, während für die Exploration eines gewählten Zustandes deutlich breitere Intervalle benötigt wurden. Dies hat einerseits die Bedeutsamkeit der nutzerspezifischen Einstellungen für die Erhöhung der Komfortabilität der Nutzung und damit der Usability und andererseits die Notwendigkeit der Studien über die für bestimmte Aufgaben geeigneten Intervallbreiten gezeigt.

Die Einbindung des weiteren Bezugssystems Menschenkörper ist in der Implementierung nicht enthalten und kann nur analytisch bewertet werden. Mit diesem Aspekt wird die Aufgabe Filter zusätz-

lich unterstützt. Die Präzision der Auswahl einer Untermenge multivarianter Daten wird verbessert. Gleichzeitig steht die zur Exploration der Zeit parallele Exploration sekundärer Schichtendaten stets in Abhängigkeit zu den Gegebenheiten des statischen Bezugssystems. Diese Abhängigkeit entsteht aufgrund der Tatsache, dass sich der Nutzer als dynamisches Bezugssystem im statischen Bezugssystem der Displaywand befindet. Dies kann in ungünstigen Fällen, in denen beispielsweise die Intervallbreiten im statischen System zu schmal oder die Schichten im dynamischen System zu breit sind, dazu führen, dass die Präzision in beiden Systemen beeinträchtigt wird, was das Gegenteil des Ziels darstellt. In der Implementierung muss deswegen darauf geachtet werden, dass eine Reduzierung der Abhängigkeit durch Ein- oder Ausschalten des Explorationsmodus für einzelne geschichtete Dimensionen möglich ist, und dass die Abbildung der Schichten auf die Achsen des interaktiven Raums in den jeweiligen Systemen proportional zueinander erfolgt.

Das Konzept für die **Visualisierung der Datenänderungen** wurde für Vergleiche zwischen den Graphzuständen und Vergleiche innerhalb eines Graphzustandes entwickelt. Es werden jeweils zwei Intervalle verglichen: das ausgewählte mit dem vorher ausgewählten. Durch die Verwendung der Bewegungsrichtung kann das vorher ausgewählte Bezugsintervall entweder das zeitlich vorherige oder das zeitlich darauf folgende sein. Durch den Einsatz des Hilfskonzeptes **Einfrierung** wird die Auswahl eines beliebigen Bezugsintervalls ermöglicht. Die Unterschiede werden durch geeignete visuelle Kodierungen hervorgehoben. Für typische Eigenschaften wurden jeweils die Alternativen der Kodierung vorgeschlagen. Die Eigenschaft Position wurde gesondert betrachtet, da die Positionsänderung das Auswandern eines für den Nutzer interessanten Knotens aus dem Fokusbereich verursachen kann. Die Implementierung erfolgte für die Änderungen der Eigenschaften Knotengröße, Hinzufügen und Löschen sowie die Positionsänderungen ohne Berücksichtigung des möglichen Auswanderns. Für die Betrachtung der Entwicklungen innerhalb eines Intervalls wurde in der Konzeption eine spezielle Ansicht vorgeschlagen, in der die im Intervall enthaltenen Zustände der selektierten Knoten und Kanten auf den Timelines abgebildet werden. In der Implementierung wurde die Timeline-Ansicht ohne spezielle Kennzeichnung der Änderungen und lediglich für Knoten umgesetzt. Jeder gespeicherte Knotenzustand wird in der Timeline dargestellt.

Mit dem Konzept werden die Interaktionsaufgaben Relate und Details-on-demand unterstützt. Es werden Änderungsbeziehungen zwischen einzelnen Zuständen der Objekte betrachtet, was der High-Level Aufgabe Vergleichen entspricht. Durch die visuelle Kodierung wird der Vergleich von Werten zwischen benachbarten Intervallen ermöglicht. In der Implementierung können die Details über einzelne Änderungen durch Checkknochen ein- und ausgeblendet werden. Für eine größere Anzahl an kodierten Änderungsvariablen wird eine spezielle Einstellungsansicht gebraucht, da sonst die Widgets zu viel wertvollen Mobildisplayplatz einnehmen. Erweiterte nutzerspezifische Einstellungen können in Zukunft die Adaptierbarkeit und somit die Intuitivität verbessern. Das Vergleichen der Änderungen ist auf zwei beliebige Intervalle beschränkt. Durch die Implementierung ist aufgefallen, dass für die Selektion des Bezugsintervalls in der Konzeption vermehrte Bewegungen entlang der z-Achse nötig sind. Soll bei einem gegebenen Intervall das Bezugsintervall geändert werden, wird zusätzlich zu dem eventuellen Einfrieren der z-Achse eine Bewegung zu dem gewünschten Bezugsintervall und eine darauf folgende Rückbewegung benötigt. Das verursacht potentielle Ablenkungen und einen Anstieg der für eine Aufgabe benötigten Zeit. Somit sinkt die Nutzereffizienz. Die Implementierung zeigt die Notwendigkeit der Option für die nicht an die Bewegung geknüpfte Selektion des Bezugsintervalls. Diese ist in der Implementierung nicht vorgesehen.

Durch die Implementierung der Änderungsvisualisierungen sind die Limitierungen der Objektmen-

ge in einzelnen Graphzuständen deutlicher geworden. Die vorgeschlagene Kodierung der Änderungen durch die Visualisierung der Position, der auf dem Intervall gelöschten Knoten oder der auf dem Bezugsintervall entstandenen Knoten benötigt Präsenz des Bezugszustandes im Graphlayout. Dies kann zu Überlappungen, zu der daraus folgenden Senkung der Erkennbarkeit und zur Begrenzung der Selektierbarkeit und damit der Usability führen. Die maximale Knoten- und Kantenanzahl soll folglich abhängig von der Änderungsanzahl kleiner sein als im vergleichbaren statischen Graphen. Im Layout sollen bei der nicht überlappenden Positionierung eines Knotens die Positionen aller Zustände anderer Knoten beachtet werden.

Durch die Timeline-Ansicht erhält man genauere Details über die Entwicklung in einem Intervall. Dies ist in den Fällen sinnvoll, in denen ein Intervall mehrere Zustände beinhaltet. Durch die Platzierung der einzelnen Knoten-Timelines untereinander sind Vergleiche der Zustände dieser Knoten möglich. Die horizontalen Timelines bieten Platz für die Darstellung mehrerer Knotenzustände. In der Implementierung ist die Ansicht mit der horizontalen Drag-Geste aufrufbar. Durch das Testen ist aufgefallen, dass die Zeitrichtung der Timelines bei solchem Aufruf nicht der Zeitrichtung im interaktiven Raum entspricht. Intuitiver wäre die Option mit vertikalem Verlauf der Timelines oder ein Aufruf durch Änderung der Position des Geräts, so dass die Timelines parallel zur z-Achse verlaufen.

Im Konzept **Multiple Ansichten** wurden Optionen der Gesten für die Umschaltung zwischen einzelnen Ansichten auf den gewählten Teilgraphenzustand beschrieben. Zudem wurde das für die vertiefte Analyse praktische Multifenster präsentiert. Das Multifenster kann für den Überblick über die gespeicherten Ansichten, für Kombinationen mit anderen Filtern oder für den Datentransfer verwendet werden. Die implementierten Graph- und Timeline-Ansichten sind eine Umsetzung multipler Ansichten.

Die gesamte entwickelte Konzeption erfordert einen hohen Einsatz räumlicher Bewegungen. Eingesetzte Touchesten werden mit einer Hand durchgeführt, wodurch das Halten des Mobilgeräts mit lediglich einer Hand während der Touchgeste-Zeit benötigt wird. Konzipierte räumliche Alternativen verursachen ebenfalls vermehrten Händeeinsatz. Zu lange Sessions im konzipierten interaktiven Raum können zu körperlicher Überlastung führen. Als Abhilfe wurden die Einfrierung und die Speicherung der Ansichten konzipiert. Die Abbildung der physischen Bewegungen auf die virtuelle Navigation begrenzt die potentiell mögliche Dauer einer Session jedoch im Vergleich zu ausschließlichem Einsatz der virtuellen Navigation. In Zukunft kann die Korrelation zwischen der Session-Länge und der Nutzerperformance im konzipierten System genauer untersucht werden.

Rashid et al. [RNQ12] haben durch eine Studie gezeigt, dass häufige Blickwechsel zwischen mehreren Displays die Performance verlangsamen. Durch die klare Rollenaufteilung des Mobilgeräts als Fokus und der Displaywand als Kontext im konzipierten System ist die Aufmerksamkeit des Nutzers hauptsächlich auf das Mobilgerät gelenkt, die Displaywand wird durch peripheres Sehen wahrgenommen. Ein häufiger Blickwechsel tritt hauptsächlich in Fällen ein, in denen die Displaywand für das Suchen gebraucht wird oder ein Wechsel des Kontextes erfolgt. Somit ist die Menge an Blickwechseln nach subjektiver Einschätzung relativ begrenzt und wirkt nicht sehr stark auf die Effizienz ein.

Die technische Seite der Implementierung ist neben der inhaltlichen Seite ein wichtiger Aspekt, der die Performance und Usability beeinflusst. Die Geschwindigkeit und Qualität der Datenzugriffe hängt von der Datenstruktur ab. Die implementierte Datenstruktur des dynamischen Graphen ermöglicht einfache Zugriffe und beachtet die Gestaltung der Attribute für verschiedene Zwecke. Gleichzeitig ist

die Speicherung jedes vollständigen Graphzustandes nicht der effizienteste Weg. Als platzsparender und schneller zeigen sich im Bereich Big Data Datenstrukturen, die nur die Änderungen als Snapshots speichern. Da die Exploration vor der Displaywand nicht für Big Data gedacht ist, und der Performancegewinn bei kleineren Datenmengen eher gering ist, wurde in der Implementierung auf diesen Weg verzichtet. Eine Optimierung kann zukünftig dennoch durchgeführt werden.

Die Implementierung ist anwendungsfallbasiert. Die Einstellungen und Gestaltung beziehen sich auf ein konkretes dynamisches Netzwerk. Zukünftig kann eine gesonderte Einstellungsansicht implementiert werden, in der zusätzlich zu der Einstellung der Änderungssichtbarkeit Einstellungen für die Auswahl gewünschter visueller Variablen, Intervallaufteilung sowie Interfacegestaltung ermöglicht wird. Zudem ist die Kompatibilität des Filters mit anderen bereits implementierten Filtern nicht automatisch vorhanden, da sie mit einem statischen Graph-Objekt arbeiten. Hierfür wären in Zukunft weitere Anpassungen nötig, um die volle Funktionalität der Anwendung auch für dynamische Graphen nutzen zu können.

FAZIT: Allgemeines Ziel vorliegender Arbeit war die Entwicklung von Konzepten für die Exploration dynamischer Graphen mit dem Schwerpunkt auf der bislang weniger untersuchten zeitlichen Dimension. Als Beitrag zur Interaktion in kombinierten Anwendungen Displaywand und Mobilgerät wurden in der Arbeit Konzepte für die zeitliche Navigation und Selektion der Zustände vorgestellt. Es wurden Möglichkeiten für Vergleiche zwischen Zuständen durch geeignete Visualisierungen beschrieben. Dabei wurden die räumlichen Freiheitsgrade des interaktiven Raums untersucht und für passende Aufgaben eingesetzt. Die Demonstration des Potentials der Konzeption erfolgte durch eine prototypische Implementierung.

Wie oben erwähnt, bedarf die Bewertung der Konzeption weiterer Studien. In vielen Konzepten wurden mehrere subjektiv intuitive Optionen vorgeschlagen, deren Effizienz und User Experience es zu testen gilt. Die zukünftige Entwicklung könnte einen stärkeren Bezug auf Multi-Nutzer Szenarien nehmen. Dies wäre lohnenswert, da die große Displaywandoberfläche ausreichend Platz für kollaborative Arbeit darstellt und die Nutzeridentifizierung durch das Mobilgerät gegeben ist. Denkbar wäre eine Komponente, in der die analytischen Daten visualisiert werden könnten, zumal der Bereich Graphen über eine große theoretische Basis mit vielen Algorithmen auch für dynamische Varianten verfügt. Zusätzlich von Interesse könnte im Gebiet dynamische Graphen die Spezialisierung der Visualisierungen auf bestimmte Inhalte wie chemische Verbindungen, auf bestimmte Graph-Typen, beispielsweise geclusterte dynamische Graphen, sein. Insgesamt werden dynamische Graphen in vielen Domänen eingesetzt und bieten viel Potential für anwendungsbasierte Untersuchungen.

Literaturverzeichnis

- [ABCD15] ARDITO, Carmelo ; BUONO, Paolo ; COSTABILE, Maria F. ; DESOLDA, Giuseppe: Interaction with Large Displays: A Survey. In: *ACM Comput. Surv.* 47 (2015), Februar, Nr. 3, S. 46:1–46:38. – ISSN 0360–0300
- [AES05] AMAR, R. ; EAGAN, J. ; STASKO, J.: Low-Level Components of Analytic Activity in Information Visualization. In: *Proceedings of the 2005 IEEE Symposium on Information Visualization (INFOVIS'05)* (2005), S. 15–15. ISBN 0–7803–9464–X
- [AEYN11] ANDREWS, Christopher ; ENDERT, Alex ; YOST, Beth ; NORTH, Chris: Information Visualization on Large, High-resolution Displays: Issues, Challenges, and Opportunities. In: *Information Visualization* 10 (2011), Oktober, Nr. 4, S. 341–355. – ISSN 1473–8716
- [AMM⁺07] AIGNER, Wolfgang ; MIKSCH, Silvia ; MÜLLER, Wolfgang ; SCHUMANN, Heidrun ; TOMINSKI, Christian: Visualizing Time-oriented data-A Systematic View. In: *Comput. Graph.* 31 (2007), Juni, Nr. 3, S. 401–409. – ISSN 0097–8493
- [APS14] AHN, Jae W. ; PLAISANT, Catherine ; SHNEIDERMAN, Ben: A Task Taxonomy For Network Evolution Analysis. In: *IEEE Transactions on Visualization and Computer Graphics* 20 (2014), Nr. 3, S. 365–376. – ISBN 1077–2626 VO – 20
- [Beh97] BEHRENS, John T.: Principles And Procedures Of Exploratory Data Analysis. In: *Psychological Methods* 2 (1997), Nr. 2, S. 131–160. – ISBN 1082–989X\r1939–1463
- [BEM⁺14] BETELLA, Alberto ; ESCUREDO, Alex ; MARTÍNEZ, Enrique ; OMEDAS, Pedro ; VERSCHURE, Paul F. M. J.: Manipulating Complex Network Structures in Virtual Reality and 3D Printing of the Results. (2014), S. 30:1–30:2. ISBN 978–1–4503–2626–1
- [BLF12] BALDAUF, Matthias ; LASINGER, Katrin ; FRÖHLICH, Peter: Private Public Screens: Detached Multi-user Interaction with Large Displays Through Mobile Augmented Reality. (2012), S. 27:1–27:4. ISBN 978–1–4503–1815–0
- [BN05] BALL, Robert ; NORTH, Chris: Effects of Tiled High-resolution Display on Basic Visualization and Navigation Tasks. (2005), S. 1196–1199. ISBN 1–59593–002–7
- [BNB07] BALL, Robert ; NORTH, Chris ; BOWMAN, Doug A.: Move to Improve: Promoting Physical Navigation to Increase User Performance with Large Displays. (2007), S. 191–200. ISBN 978–1–59593–593–9
- [Bra99] BRATH, Richard K.: *EFFECTIVE INFORMATION VISUALIZATION Guidelines and Metrics for 3D Interactive Representations of Business Data*, University of Toronto, Diss., 1999
- [BSRG06] BAITALUK, Michael ; SEDOVA, Mayya ; RAY, Animesh ; GUPTA, Amarnath: Biological Networks: Visualization And Analysis Tool For Systems Biology. In: *Nucleic Acids Research* 34 (2006), Nr. WEB. SERV. ISS., S. 466–471. – ISBN 1362–4962

- (Electronic)
- [CKB09] COCKBURN, Andy ; KARLSON, Amy ; BEDERSON, Benjamin B.: A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. In: *ACM Comput. Surv.* 41 (2009), Januar, Nr. 1, S. 2:1–2:31. – ISSN 0360–0300
- [CMS99] CARD, Stuart K. (Hrsg.) ; MACKINLAY, Jock D. (Hrsg.) ; SHNEIDERMAN, Ben (Hrsg.): *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1999. – ISBN 1–55860–533–9
- [CSR+03] CZERWINSKI, Mary ; SMITH, Greg ; REGAN, Tim ; MEYERS, Brian ; ROBERTSON, George ; STARKWEATHER, Gary: Toward Characterizing the Productivity Benefits of Very Large Displays. In: *Interact* 3 (2003), S. 9–16. ISBN 1586033638
- [DB09] DACHSELT, Raimund ; BUCHHOLZ, Robert: Natural Throw and Tilt Interaction Between Mobile Phones and Distant Displays. (2009), S. 3253–3258. ISBN 978–1–60558–247–4
- [DFAB04] DIX, A ; FINLAY, J ; ABOWD, G D. ; BEALE, R: *Human-Computer Interaction*. 2004. – 834 S. – ISBN 0130461091
- [DHKQ14] DOSTAL, Jakub ; HINRICHS, Uta ; KRISTENSSON, Per O. ; QUIGLEY, Aaron: SpiderEyes: Designing Attention- and Proximity-aware Collaborative Interfaces for Wall-sized Displays. (2014), S. 143–152. ISBN 978–1–4503–2184–6
- [Die00] DIESTEL, Reinhard: *Graphentheorie*. 2000. – 326 S. – ISBN 3540676562
- [EB06] ECKMAN, B A. ; BROWN, P G.: Graph Data Management For Molecular And Cell Biology. In: *IBM Journal of Research and Development DOI - 10.1147/rd.506.0545* 50 (2006), Nr. 6, S. 545–560. – ISSN 0018–8646
- [EWCP13] EBERT, Achim ; WEBER, Christopher ; CERNEA, Daniel ; PETSCH, Sebastian: TangibleRings: Nestable Circular Tangibles. In: *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2013 (CHI EA '13). – ISBN 978–1–4503–1952–2, S. 1617–1622
- [FD10] FRISCH, Mathias ; DACHSELT, Raimund: Off-screen Visualization Techniques for Class Diagrams. (2010), S. 163–172. ISBN 978–1–4503–0028–5
- [FHD09] FRISCH, Mathias ; HEYDEKORN, Jens ; DACHSELT, Raimund: Investigating Multitouch and Pen Gestures for Diagram Editing on Interactive Surfaces. (2009), S. 149–156. ISBN 978–1–60558–733–2
- [Flo14] FLORIN, Alexander. *Farbe im Web*. 2014
- [GF12] GONCALVES, Paulo ; FONSECA, Manuel J.: Combining Multitouch Gestures And Sketches To Explore Photo Collections. In: *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC* (2012), S. 251–252. – ISBN 9781467308502
- [GPG+15] GRUBERT, Jens ; PAHUD, Michel ; GRASSET, Raphael ; SCHMALSTIEG, Dieter ; SEICHTER, Hartmut: The Utility Of Magic Lens Interfaces On Handheld Devices For Touristic Map Navigation. In: *Pervasive and Mobile Computing* 18 (2015), S. 88–103. – ISSN 15741192

- [GRE11] GHANI, S. ; RICHE, N. H. ; ELMQVIST, N.: Dynamic insets for context-aware graph navigation. In: *Computer Graphics Forum* 30 (2011), Nr. 3, S. 861–870. ISBN 01677055
- [Gre12] Shape identification in temporal data sets. In: *Expanding the Frontiers of Visual Analytics and Visualization* (2012), S. 305–321. ISBN 978–1–4471–2803–8
- [Hea08] HEARST, M. *CourseSIMS 247: Information Visualization and Presentation*. 2008
- [HHLI12] HOSSAIN, Zahid ; HASAN, Khalad ; LIANG, Hai-Ning ; IRANI, Pourang: EdgeSplit: Facilitating the Selection of Off-screen Objects. (2012), S. 79–82. ISBN 978–1–4503–1105–2
- [HMM00] HERMAN, I. ; MELANCON, G. ; MARSHALL, M.S.: Graph Visualization And Navigation In Information Visualization: A Survey. In: *IEEE Transactions on Visualization and Computer Graphics* 6 (2000), Nr. 1, S. 1–21. – ISBN 1077–2626 VO – 6
- [Ish07] ISHII, Hiroshi: Tangible User Interfaces. In: *CHI 2006 workshop* (2007), S. 1–17. ISBN 1420088858
- [KAP11] KÄSER, Dominik P. ; AGRAWALA, Maneesh ; PAULY, Mark: FingerGlass: Efficient Multiscale Interaction on Multitouch Screens. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2011 (CHI '11). – ISBN 978–1–4503–0228–9, S. 1601–1610
- [KG06] KUMAR, Gautam ; GARLAND, Michael: Visual Exploration Of Complex Time-varying Graphs. In: *IEEE Transactions on Visualization and Computer Graphics* 12 (2006), Nr. 5, S. 805–812. – ISBN 1077–2626
- [KRA12] KURDYUKOVA, Ekaterina ; REDLIN, Matthias ; ANDRÉ, Elisabeth: Studying User-defined iPad Gestures for Interaction in Multi-display Environment. (2012), S. 93–96. ISBN 978–1–4503–1048–2
- [KRD14a] KISTER, Ulrike ; REIPSCHLÄGER, Patrick ; DACHSELT, Raimund: Multi-Touch Manipulation of Magic Lenses for Information Visualization. (2014), S. 431–434. ISBN 978–1–4503–2587–5
- [KRD14b] KISTER, Ulrike ; REIPSCHLÄGER, Patrick ; DACHSELT, Raimund: Multi-Touch Manipulation of Magic Lenses for Information Visualization. In: *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces - ITS '14* (2014), S. 431–434. ISBN 9781450325875
- [Leh15] LEHMANN, Anke: Distanzabhängige Interaktion in großen hochauflösenden Displayumgebungen. (2015), S. 270
- [LGB07] LOOSER, Julian ; GRASSET, Raphael ; BILLINGHURST, Mark: A 3D Flexible and Tangible Magic Lens in Augmented Reality. (2007), S. 1–4. ISBN 978–1–4244–1749–0
- [Lib] *Libavg*
- [LML⁺10] LANE, Nicholas D. ; MILUZZO, Emiliano ; LU, Hong ; PEEBLES, Daniel ; CHOUDHURY, Tanzeem ; CAMPBELL, Andrew T.: A Survey Of Mobile Phone Sensing. In: *IEEE Communications Magazine* 48 (2010), Nr. 9, S. 140–150. – ISBN 0163–6804 VO – 48
- [Lon15] LONGO, Luca. *Best Practices For Buttons: The User Experience Of Colors*. 2015

- [LPP⁺06] LEE, Bongshin ; PLAISANT, Catherine ; PARR, Cynthia S. ; FEKETE, Jean-Daniel ; HENRY, Nathalie: Task Taxonomy for Graph Visualization. (2006), S. 1–5. ISBN 1–59593–562–2
- [LR06] LAHRES, B. ; RAÏMAN, G.: *Praxisbuch Objektorientierung: von den Grundlagen zur Umsetzung*. Galileo Press, 2006 (Galileo computing). – ISBN 9783898426244
- [LYK⁺09] LEE, Gun A. ; YANG, Ungyeon ; KIM, Yongwan ; JO, Dongsik ; KIM, Ki-Hong ; KIM, Jae H. ; CHOI, Jin S.: Freeze-Set-Go Interaction Method for Handheld Mobile Augmented Reality Environments. (2009), S. 143–146. ISBN 978–1–60558–869–8
- [Mac86] MACKINLAY. *Automating the Design of Graphical Presentations of Relational Information*. 1986
- [MHK⁺14] MORAWA, Robert ; HORAK, Tom ; KISTER, Ulrike ; MITSCHICK, Annett ; DACHSELT, Raimund: Combining Timeline and Graph Visualization. (2014), S. 345–350. ISBN 978–1–4503–2587–5
- [MRJR15] MÜLLER, Jens ; RÄDLE, Roman ; JETTER, Hans-Christian ; REITERER, Harald: An Experimental Comparison of Vertical and Horizontal Dynamic Peephole Navigation. (2015), S. 1523–1526. ISBN 978–1–4503–3145–6
- [Mul] *Lobby Area*
- [Net] *NetworkX*
- [Nex] *Asus Google Nexus 7 (2013)*
- [PD10a] PREIM, Bernhard ; DACHSELT, Raimund: *Interaktive Systeme Band 2*. 2010. – 423 S. – ISBN 978–3–642–05401–3
- [PD10b] PREIM, Bernhard ; DACHSELT, Raimund: *Präsentation , Navigation und Interaktion*. 2010. – 509–580 S. – ISBN 9783642054020
- [RJMR14] RÄDLE, Roman ; JETTER, Hans-Christian ; MÜLLER, Jens ; REITERER, Harald: Bigger is Not Always Better: Display Size, Performance, and Task Load During Peephole Map Navigation. (2014), S. 4127–4136. ISBN 978–1–4503–2473–1
- [RLE05] RYALL, Kathy ; LI, Qing ; ESENTER, Alan: Temporal Magic Lens: Combined spatial and temporal query and presentation. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3585 LNCS (2005), S. 809–822. – ISBN 3540289437
- [RNQ12] RASHID, Umar ; NACENTA, Miguel A. ; QUIGLEY, Aaron: The Cost of Display Switching: A Comparison of Mobile, Large Display and Hybrid UI Configurations. In: *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI'12)* (2012), S. 99–106
- [SBR13] SEIFERT, Julian ; BAYER, Andreas ; RUKZIO, Enrico: PointerPhone: Using mobile phones for direct pointing interactions with remote displays. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8119 LNCS (2013), Nr. PART 3, S. 18–35. – ISBN 9783642404764
- [SD09] SPINDLER, Martin ; DACHSELT, Raimund: PaperLens: Advanced Magic Lens Interac-

- tion Above the Tabletop. (2009), S. 7:1–7:1. ISBN 978–1–60558–733–2
- [Sem16] SEMERTZIDIS, Konstantinos: Time Traveling in Graphs using a Graph Database. In: *Proceedings of Workshops of the EDBT/ICDT 2016 Joint Conference (March 15, 2016, Bordeaux, France)* (2016), S. 1–6
- [SFR01] STOEV, Stanislav L. ; FEURER, Matthias ; RUCKABERLE, Michael: Exploring the Past: A Toolset for Visualization of Historical Events in Virtual Environments. In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. New York, NY, USA : ACM, 2001 (VRST '01). – ISBN 1–58113–427–4, S. 63–70
- [Shn96] SHNEIDERMAN, Ben: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. (1996), S. 336–. ISBN 0–8186–7508–X
- [SNDC10] SCHMIDT, Sebastian ; NACENTA, Miguel a. ; DACHSELT, Raimund ; CARPENDALE, Sheelagh: A Set of Multi-touch Graph Interaction Techniques. In: *ACM International Conference on Interactive Tabletops and Surfaces* (2010), S. 113–116. ISBN 9781450303996
- [SSG13] SCHMIDT, Dominik ; SAS, Corina ; GELLERSEN, Hans: Personal Clipboards for Individual Copy-and-paste on Shared Multi-user Surfaces. (2013), S. 3335–3344. ISBN 978–1–4503–1899–0
- [SSRG12] SCHMIDT, Dominik ; SEIFERT, Julian ; RUKZIO, Enrico ; GELLERSEN, Hans: A Cross-device Interaction Style for Mobiles and Surfaces. (2012), S. 318–327. ISBN 978–1–4503–1210–3
- [STSD10] SPINDLER, Martin ; TOMINSKI, Christian ; SCHUMANN, Heidrun ; DACHSELT, Raimund: Tangible Views for Information Visualization. (2010), S. 157–166. ISBN 978–1–4503–0399–6
- [TAVS06] TOMINSKI, Christian ; ABELLO, James ; VAN HAM, Frank ; SCHUMANN, Heidrun: Fisheye tree views and lenses for graph visualization. In: *Proceedings of the International Conference on Information Visualisation* (2006), S. 17–24. – ISBN 0769526020
- [TGK⁺14] TOMINSKI, C ; GLADISCH, S ; KISTER, U ; DACHSELT, R ; SCHUMANN, H: A Survey on Interactive Lenses in Visualization. In: *Eurographics Conference on Visualization (EuroVis)* (2014). ISBN <http://hdl.handle.net/10.2312/eurovisstar.20141172.043-062>
- [VDRN⁺15] VEPSÄLÄINEN, Jouni ; DI RIENZO, Antonella ; NELIMARKKA, Matti ; OJALA, Jouni A. ; SAVOLAINEN, Petri ; KUIKKANIEMI, Kai ; TARKOMA, Sasu ; JACUCCI, Giulio: Personal Device As a Controller for Interactive Surfaces: Usability and Utility of Different Connection Methods. (2015), S. 201–204. ISBN 978–1–4503–3899–8
- [War04] WARE, Colin: *Information visualization: perception for design*. Morgan Kaufmann, 2004. – ISBN 1558608192
- [War12] WARE, Colin: *Foundation for a Science of Data Visualization*. 2012. – 1–27 S. – ISBN 1558608192
- [YKSJ07] YI, Ji S. ; KANG, Youn a. ; STASKO, John ; JACKO, Julie: Toward a Deeper Understanding of the Role of Interaction in Information Visualization. In: *IEEE Transactions on Visualization and Computer Graphics* 13 (2007), November, Nr. 6, S. 1224–1231. – ISSN 1077–2626

Danksagung

Ich danke meinen Betreuern Ulrike Kister und Konstantin Klamka für die Unterstützung bei der Anfertigung meiner Bachelorarbeit. Ihre zahlreichen Ideen, Verbesserungsvorschläge und Hinweise für die Umsetzung haben mir sehr geholfen. Ebenso möchte ich mich bei Bernhard Kroll und Simon Hanisch für ihre Mühen beim Korrekturlesen bedanken.

