

Institut für Visualisierung und Interaktive Systeme  
Universität Stuttgart  
Pfaffenwaldring 5B  
D-70569 Stuttgart

**Visuelle interaktive Exploration  
relationaler Daten mit  
geographischem Bezug**

Roberto Fontanarosa  
Masterarbeit

Studiengang:	Informatik
Prüfer:	Prof. Dr. Thomas Ertl
Betreuer:	Dr. Steffen Koch Dr. Dennis Thom
Beginn der Arbeit:	29.09.2017
Ende der Arbeit:	29.03.2018



## **Kurzfassung**

Für die Visualisierung und Exploration von Daten mit geographischem Bezug gibt es bereits viele verschiedene Methoden, die je nach Aufgabenstellung und dem zugrundeliegenden Graphen unterschiedlich gut geeignet sind. Netzwerke mit sowohl regionalen als auch transkontinentalen Relationen stellen für gewöhnliche Techniken häufig eine Herausforderung dar. In dieser Arbeit wird ein Verfahren vorgestellt, bei dem der Fokus auf der schrittweisen Exploration von Graphen mit verschiedenen Relationen liegt. Es beruht auf einem Konzept, bei dem Minimaps verwendet werden, um Orte zu repräsentieren, die sich außerhalb des Viewports befinden. Für die Positionierung der Minimaps wird ein kräftebasierter Ansatz verwendet. Zu diesem Konzept wurde ein Softwareprototyp implementiert, der anhand einer Nutzerstudie, die in dieser Arbeit vorgestellt wird, evaluiert wurde. Die Ergebnisse zeigen, dass dieser Ansatz besser als ein simpler Übersicht + Details-Ansatz geeignet ist, um die Aufgaben aus der Nutzerstudie zu lösen.

## **Abstract**

Many approaches exist already when it comes to visualizing and exploring data with a geographical context. Their feasibility changes based on the given task or the graph that needs to be explored. Networks with regional and transcontinental relations can often pose a challenge for simple approaches. The approach presented in this paper that is based on a concept that uses minimaps in order to display nodes that are placed outside of the viewport. The positioning of the minimaps is done using a force-based approach. The main focus is the stepwise exploration of graphs with many different relations. A software-prototype has been implemented based on this concept. It is used for a user-study that is presented in this paper in order to evaluate that concept. Results show that this approach is superior compared to a simple overview + details approach when it comes to solving the tasks from the user study.





# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>11</b>
<b>2</b>	<b>Visualisierung relationaler Daten auf Weltkarten</b>	<b>13</b>
2.1	Verwandte Arbeiten . . . . .	13
2.1.1	Übersicht + Details und Zooming + Panning . . . . .	14
2.1.2	Fokus und Kontext . . . . .	16
2.1.3	Cue-basierte Ansätze . . . . .	19
2.2	Offscreen-Visualisierung . . . . .	21
<b>3</b>	<b>Konzept</b>	<b>27</b>
3.1	Definition und Schreibweise . . . . .	27
3.2	Beschreibung des Konzepts . . . . .	28
3.3	Positionierung der Minimap . . . . .	31
3.3.1	Kräftebasierter Ansatz zur Darstellung von Knoten-Kanten Diagrammen . . . . .	34
3.3.2	Anpassung der Kräfte . . . . .	41
3.3.3	Überlagerung von Minimaps . . . . .	45
3.4	Clustering . . . . .	53
3.5	Anpassung der Kanten . . . . .	63
3.6	Interaktion . . . . .	67
<b>4</b>	<b>Evaluation</b>	<b>75</b>
4.1	Setup . . . . .	75
4.2	Ergebnisse . . . . .	81

4.2.1	Komplexität bei Graphen mit regionalen und transkontinentalen Relationen . . . . .	81
4.2.2	Nutzer-Feedback zum Tool . . . . .	85
4.2.3	Direkter Vergleich der beiden Ansätze . . . . .	93
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>96</b>
5.1	Ausblick . . . . .	97

## Abbildungsverzeichnis

1	Übersicht + Details im Spiel League of Legends . . . . .	15
2	Fisheye Lens für die Darstellung eines Metro-Netzes . . . . .	17
3	Quelle: Sarkar et al. (SB92) . . . . .	18
4	Quelle: Sarkar et al. (SB92) . . . . .	19
5	Links: Offscreen-Objekte mit Halo dargestellt. Rechts: Die gleichen Offscreen-Objekt mit Wedges repräsentiert. Quelle: (GBGI08) . . . . .	20
6	a) Proxy-Elemente werden als rote Knoten dargestellt. b) Der rote Inset für Orte in Nord- und Mittelamerika ist geöffnet und zeigt genauere Informationen an. Quelle: (BKA <sup>+</sup> 16) . . . . .	22
7	a) Proxy-Elemente für Offscreen-Elemente sind am Rande platziert. b) Zwei Offscreen-Insets für Asien und Nordamerika werden angezeigt. Quelle: (BKA <sup>+</sup> 16) . . . . .	24
8	Simpler Ansatz zur Visualisierung von Informationsflüssen . . . . .	29
9	Minimap zur Darstellung von Grönland und Island in Europa. . . . .	30
10	Die Minimap am Rande des Kartenausschnitts zeigt den Zielort Rio de Janeiro an. . . . .	32
11	Fall ohne Überdeckung der Minimaps mit anderen Objekten . . . . .	33
12	Fall mit Überdeckung von zwei Minimaps. . . . .	34
13	Pseudocode für den Algorithmus von Fruchterman-Reingold . . . . .	37
14	Ergebnis nach 100 Iterationen . . . . .	39
15	Ergebnis nach 500 Iterationen . . . . .	39
16	Ergebnis nach 1000 Iterationen . . . . .	40
17	Ergebnis mit neuem Kräftemodell nach 100 Iterationen . . . . .	43
18	Ergebnis mit neuem Kräftemodell nach 1000 Iterationen . . . . .	44

19	Überlappung von Minimaps auf Grund des gewählten Kräftemodells . . . . .	45
20	Grafik aus (BKA <sup>+</sup> 16): Linkes Bild: Zustand vor Kollisionsbehebung. Rechtes Bild: Zustand nach Kollisionsbehebung . .	47
21	Algorithmen aus Bordkorb et al., die für die Kollisionsbehebung verwendet werden . . . . .	48
22	Kollision zwischen vier Minimaps . . . . .	50
23	Platzierung der Minimaps nach Behebung der Kollision aus 22	50
24	Minimaps werden nicht absteigend nach Breitengrad untereinander platziert . . . . .	52
25	Minimaps werden richtigerweise absteigend nach Breitengrad untereinander platziert . . . . .	53
26	Minimaps vor dem Clustern . . . . .	55
27	Die Minimaps für New York City und Philadelphia aus Abbildung 26 sind zu einer Minimap geclustert worden. . . . .	56
28	Die farbigen Kreise signalisieren die Orte, die durch die gleichfarbigen Minimaps repräsentiert werden. . . . .	59
29	Der grüne Cluster wurde mit der gelben Minimap zu einem Cluster vereint, der den gelben und grünen Ort enthält. . . . .	60
30	Die gelbe Minimap wurde mit der grünen und roten zu einem Cluster vereint. . . . .	61
31	Visualisierung von größeren Clustern für $p_{curr} = \text{Tokio}$ . . . . .	62
32	Verbindungen zu allen Orten einer Minimap führen zu Visual Clutter. . . . .	64
33	Ausgehende Kante: Pfeilrichtung im Marker . . . . .	65
34	Eingehende Kante: Pfeilrichtung im Marker . . . . .	66
35	Ein- und ausgehende Kante: Pfeilrichtung im Marker . . . . .	66

36	Alle Knoten des Graphen werden beim Öffnen des Tools angezeigt. . . . .	67
37	Vor und nach Paninteraktion nach links . . . . .	69
38	Roter Marker bei Stuttgart zeigt den vorher ausgewählten Knoten an. . . . .	71
39	Sidebar links zeigt zusätzliche Informationen zum aktuell sichtbaren Teilgraphen an. . . . .	72
40	Elemente der Sidebar . . . . .	74
41	Auszug aus dem Fragebogen für die Nutzerstudie . . . . .	77
42	Simpler Fall, bei dem alle Knoten im Viewport enthalten sind.	78
43	Abbildungen zu Aufgaben 5 und 7 des Fragebogens . . . . .	79
44	Zeitmessungen für Aufgaben 3, 5 und 7 . . . . .	82
45	Schwierigkeitseinstufungen von Nutzergruppe A für Aufgabe 3, 5 und 7 . . . . .	84
46	Schwierigkeitseinstufungen von Nutzergruppe B für Aufgabe 3, 5 und 7 . . . . .	84
47	Links: Runde Form. Rechts: Quadratische Form . . . . .	89
48	Links: Mit Schatten. Rechts: Ohne Schatte . . . . .	89
49	Links: Pfeil innerhalb des Markers. Rechts: Pfeil zu Beginn oder Ende der Verbindungslinie zum Knoten. . . . .	91

## Tabellenverzeichnis

1	Zeitmessungen für das Lösen der Aufgabe 3, 5 und 7 . . . . .	82
2	Schwierigkeitseinstufungen für Aufgabe 3, 5 und 7 . . . . .	83
3	Zeitmessungen fürs Lösen der Aufgabe 9 . . . . .	85
4	Schwierigkeitseinstufungen für Aufgabe 9 . . . . .	85
5	Bewertung für die Platzierung der Minimaps . . . . .	87
6	Ergebnisse zu Frage 16 . . . . .	88







# 1 Einleitung

Während es für die Visualisierung von Graphen viele verschiedene Methoden und Layoutalgorithmen gibt, gestaltet sich die Visualisierung von Netzwerken relationaler Daten mit geographischem Bezug deutlich schwieriger. Der geographische Bezug verhindert eine freie Platzierung der Knoten, oder schränkt sie zumindest ein. Die visuelle Analyse solcher Netzwerke erfordert vor allem dann zusätzliche Interaktion, wenn Orte mit stark variierenden Entfernungen exploriert werden sollen. Dieses Problem verstärkt sich insbesondere dann, wenn die zu explorierenden Relationen sowohl regional als auch transnational oder interkontinental sind. Grund dafür ist, dass Details schwer erkennbar sind, wenn Verbindungen weit entfernter Orte gezeichnet werden. Werden wiederum Details regionaler Verbindungen dargestellt, lassen sich stattdessen weit entfernte Relationen nicht oder nicht mehr vollständig anzeigen.

Mithilfe sequentieller Interaktionstechniken aus der Geovisualisierung, wie zum Beispiel Zooming und Panning, ist die Exploration solcher Graphen per se möglich, aber in der Praxis vor allem dann nicht praktikabel, wenn viele Relationen mit stark variierenden Distanzen existieren.

Im Rahmen dieser Arbeit wird eine neue Interaktionsmethode vorgestellt, die genau dieses Problem adressiert und die Möglichkeit bieten soll, für einen selektierten Knoten die Exploration seiner Relationen zu erleichtern. Der Lösungsansatz beruht auf der Benutzung von Minimaps. Dabei handelt es sich um kleine Kartenausschnitte, die weit entfernte Orte darstellen und genau dann angezeigt werden, wenn weit entfernten Relationen nicht mehr sichtbar wären. Verschiedene Ansätze wurden dafür zur Wahl und Platzierung der Minimaps erst konzipiert, anschließend mit Web-Technologien implementiert und abschließend evaluiert.

Die Arbeit ist so strukturiert, dass zu Beginn existierende Konzepte und Lösungsansätze für die visuelle Exploration relationaler Daten vorgestellt und verglichen werden. Das Konzept der Visualisierungstechnik, die im Rahmen

dieser Arbeit entstanden ist, wird daraufhin vorgestellt, indem die Grundidee erläutert und dann schrittweise erweitert wird, um existierende und neu auftretende Probleme zu beheben. Nach der Beschreibung des Konzepts folgen genauere Informationen zur Umsetzung und Implementierung, deren Ergebnisse anhand von Beispielen verdeutlicht werden. Sowohl das Konzept als auch die Implementierung werden durch eine Nutzerstudie evaluiert. Die Evaluation und ihre Ergebnisse sind im vorletzten Teil dieser Arbeit genauer beschrieben. Im letzten Teil wird die Arbeit zusammengefasst, und es folgt ein Ausblick, in dem mögliche Schritte für die Optimierung der Visualisierungstechnik aufgelistet werden.

## 2 Visualisierung relationaler Daten auf Weltkarten

Mit der Visualisierung relationaler Daten auf Weltkarten beschäftigen sich Wissenschaftler bereits seit vielen Jahren. Es geht dabei darum, gerichtete oder ungerichtete Informationsflüsse zwischen mehreren Orten anschaulich wiederzugeben, um dem Nutzer ein möglichst gutes Verständnis über das visualisierte Netzwerk zu ermöglichen.

Schwierig wird diese Visualisierung vor allem dann, wenn Relationen zu Orten angezeigt werden sollen, die sich in unterschiedlich großen Entfernungen vom aktuell sichtbaren Kartenausschnitt befinden. Ist der Nutzer beispielsweise daran interessiert sowohl regionale Details, wie Straßen- oder Städtenamen, aber auch interkontinentale Relationen zu sehen, so reichen simple Visualisierungsmethoden nicht mehr aus. Eine erhöhte Zoomstufe ermöglicht zwar das Lesen von Details, sie verringert aber gleichzeitig auch die Größe des sichtbaren Kartenausschnitts, weshalb weit entfernte Orte dadurch nicht mehr sichtbar sind.

Interaktionen vonseiten der Nutzer, wie beispielsweise Zooming und Panning, sind notwendig, um solch komplexe Graphen zu explorieren, benötigen allerdings viel Zeit und sind oftmals umständlich.

Es gibt bereits viele Arbeiten, die sich mit der Problematik Offscreen-Objekte anzuzeigen, beschäftigen haben.

### 2.1 Verwandte Arbeiten

Für User-Interaktionen bezüglich des Explorieren von Knoten-Kanten Diagrammen im Allgemeinen angeht, gibt es bereits eine Vielzahl von Ansätzen, die sich unter anderem in folgende Kategorien, wie von Frisch et al. (FD13) vorgeschlagen, unterteilen lassen:

1. Übersicht + Details und Zooming + Panning

2. Fokus und Kontext
3. Cue-basierte Ansätze
4. Off-Screen-Visualisierung

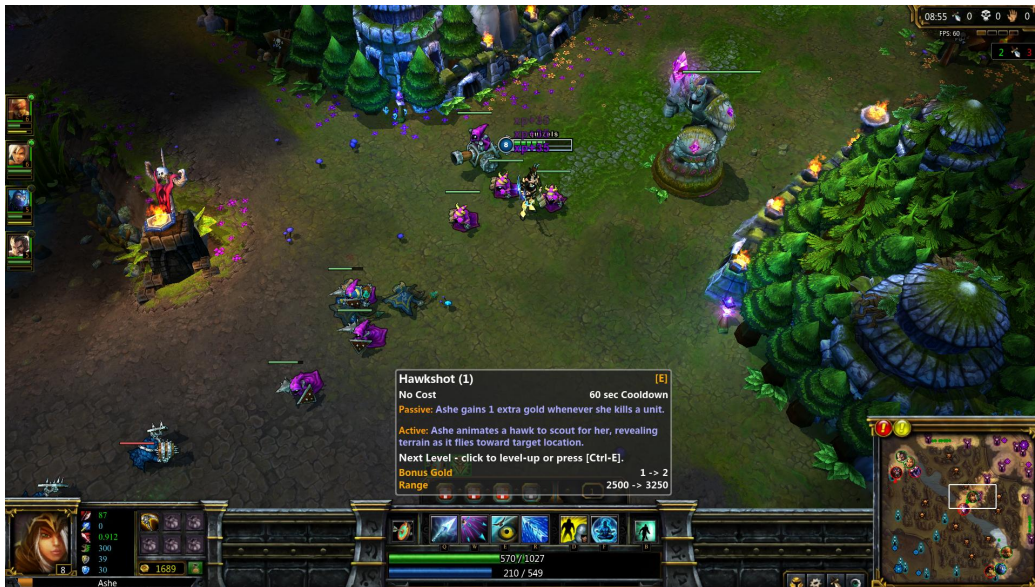
### 2.1.1 Übersicht + Details und Zooming + Panning

Der Ansatz Übersicht + Details und Zooming + Panning ist einer der häufigst verwendeten Ansätze, wenn es darum geht, Details eines Graphen zu erforschen und dabei trotzdem einen Überblick des gesamten Graphen zu behalten. Das Prinzip besteht darin zwei Arten von Fenstern anzuzeigen. Eine zeigt eine Übersicht, während die andere Details des Graphen zeigt. In diesen Fenstern lässt sich mit Zoom- und Panoperationen der dargestellte Graph explorieren.

Eine von vielen und weltweit bekannte Anwendung, die genau diesen Ansatz verfolgt, ist Google Maps (goo).

Der Graph wird bei diesem Ansatz in einem Fenster angezeigt, den man mit Hilfe von Zoom- und Paninteraktionen genauer explorieren kann (Details). Dabei wird, typischerweise in einer Ecke des Ausschnitts, der den zu explorierenden Graphen enthält, ein zusätzliches, aber verkleinertes Fenster hinzugefügt, das den gesamten Graphen aus einer deutlichen geringeren Zoomstufe anzeigt (Übersicht).

Ein zusätzliches Rechteck im Inneren des Fensters signalisiert dabei den gerade sichtbaren Bereich. Bei Pan-Operationen auf das Hauptfenster verschiebt sich dieses Rechteck, während es beim Zoomen die Größe entsprechend verändert. Diese Technik wird auch häufig bei Computerspielen, wie beispielsweise bei League of Legends (lol), wiederverwendet (siehe Abbildung 1). Eine Übersichtskarte ist in der unteren rechten Ecke platziert. Der im Detail gezeigte Kartenausschnitt wird durch ein weißes Rechteck im Inneren der Übersichtskarte signalisiert.



Quelle: <https://bothgunsblazingblog.files.wordpress.com/2013/11/league-of-legends-2.jpg>

Abbildung 1: Übersicht + Details im Spiel League of Legends

Neben der Tatsache, dass das Fenster, das den gesamten Graphen in einer kleineren Zoomstufe anzeigt, zusätzlichen Platz beansprucht, ist die mentale Beanspruchung von Nutzern bei der Interaktion hoch: Beide Fenster lassen sich unabhängig voneinander bedienen, haben aber jeweils einen Einfluss auf das entsprechend andere. Mit dieser Problematik haben sich unter anderem Cockburn et al. beschäftigt (CKB09).

Ein weiteres Problem ist, dass sich Details weit entfernter Orte nicht ohne Interaktion anzeigen lassen. Möchte der User gleichzeitig Details mehrerer entfernter Orte sehen, lässt sich das ohne diesen Ansatz anzupassen, nicht realisieren.

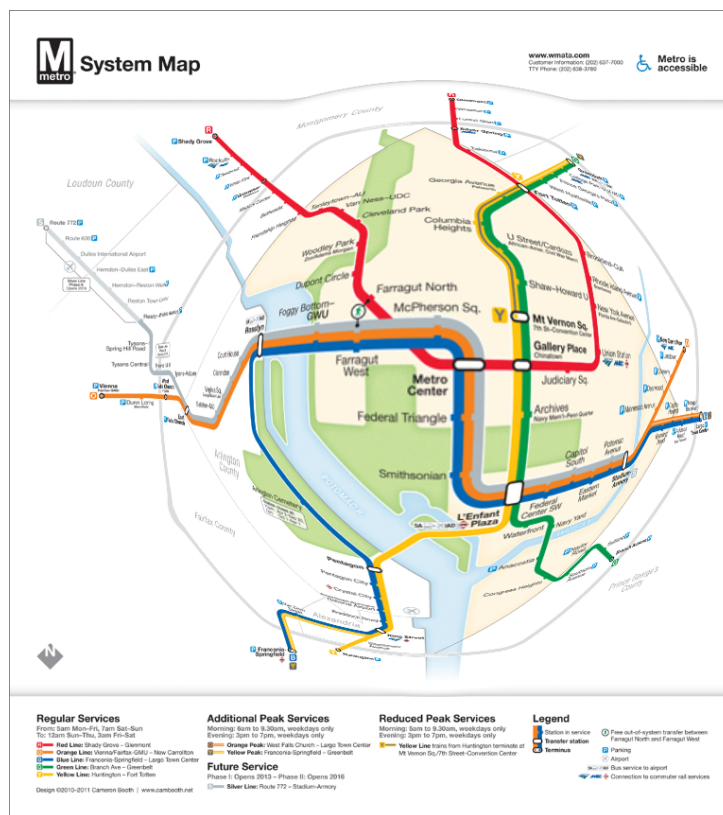
Die Technik für Übersicht + Details und Zooming + Panning wird ebenfalls bei der Visualisierung von Graphen mit Daten ohne geographischen Bezug verwendet und findet Einsatz in Software wie zum Beispiel Microsoft Visio (vis).

Nekrasovski et al. (NBM<sup>+</sup>06) haben diesen Ansatz und den Ansatz für Fokus und Kontext, der im nächsten Kapitel vorgestellt wird, auf großen Baumstrukturen verglichen und festgestellt, dass das Übersichtsfenster keinen Einfluss auf die Performance des Users hat.

### **2.1.2 Fokus und Kontext**

Im Gegensatz zu Übersicht + Details und Zooming + Panning werden bei Fokus und Kontext die Übersicht- und Detailansicht in einer einzigen Ansicht vereint. Dabei werden in einem Fenster, das eine Übersicht über den Graphen anzeigt, Bereiche und Elemente, die genauer untersucht werden sollen, in einer detaillierteren Form angezeigt, indem unterschiedliche Verzerrungen angewandt werden.

Eine in der Informationsvisualisierung häufig genutzte Technik, die auf dem Prinzip von Fokus und Text basiert, ist Fisheye Lens (Fur86). Es handelt sich dabei um Bereiche (typischerweise kreisförmig), die den Inhalt so verzerren, dass Punkte im Innersten der Form am stärksten vergrößert und Punkte, die sich weiter außen befinden, kleiner dargestellt werden. Das Ergebnis für kreisförmige Formen ähnelt einem Fischauge, was dem Ganzen seinen Namen gibt (siehe 5).



Quelle:

[http://holisticsofa.com/wp-content/uploads/2012/02/IMG\\_0230.png](http://holisticsofa.com/wp-content/uploads/2012/02/IMG_0230.png)

Abbildung 2: Fisheye Lens für die Darstellung eines Metro-Netzes

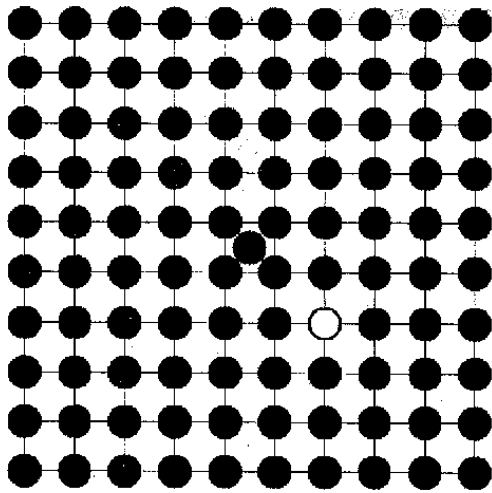
Verzerrungstechniken bei Fisheye-Lenses lassen sich, wie beispielsweise in der Arbeit von Frisch et al. (FD13), in globale und lokale Verzerrungstechniken kategorisieren.

Bei globalen Verzerrungstechniken wird der gewählten Knoten und der Bereich um ihn herum so verzerrt, dass er vergrößert wird. Dadurch werden mehr Details des Knotens und seiner Umgebung erkennbar. Im Gegenzug werden weit entfernte Knoten so verzerrt, dass sich Details schlechter erkennen lassen. Sarkar et al. (SB92) wenden in ihrer Arbeit zwei Arten von Transformationen an:

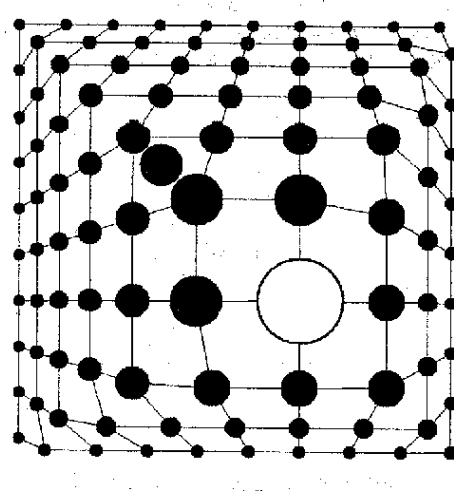
- Polare Transformation

- Kartesische Transformation

Bei der polaren Transformation, siehe Abbildung 3, wird der Graph kreisförmig verzerrt, während bei der kartesischen Transformation, siehe Abbildung 4, eine Verzerrung entlang der Achsen stattfindet.



(a) Graph ohne Verzerrung



(b) Polare Verzerrung des linken Graphen für den weißen Knoten

Abbildung 3: Quelle: Sarkar et al. (SB92)



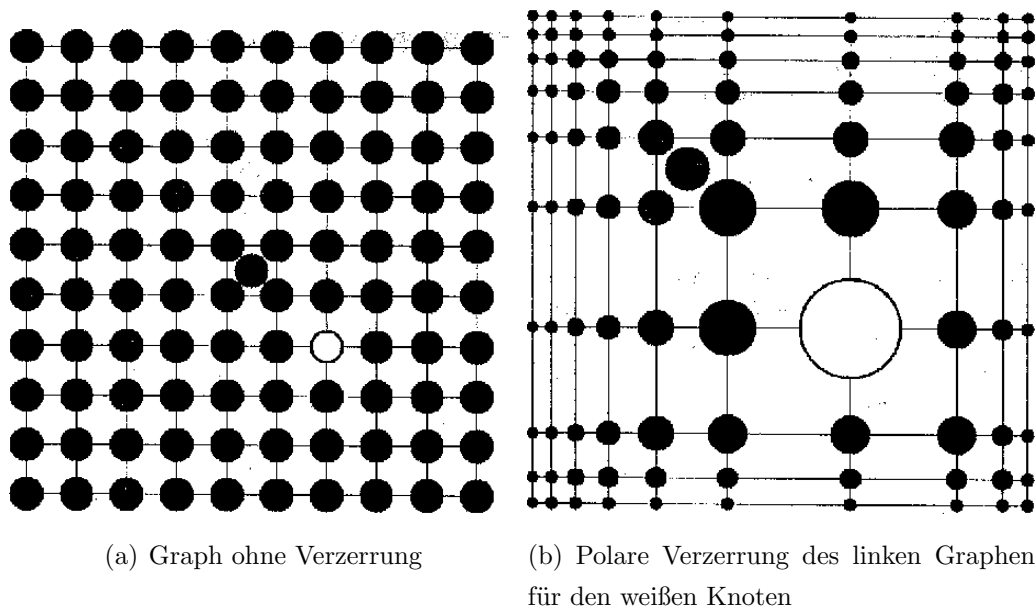


Abbildung 4: Quelle: Sarkar et al. (SB92)

Bei der lokalen Verzerrungsstrategie wird im Gegensatz zur globalen Verzerrungsstrategie nicht der gesamte Graph, sondern stattdessen lokal nur der Bereich innerhalb der gewählten Lupe verzerrt. Tominski et al. (TAVHS06) nutzen in ihrer Arbeit unterschiedliche Arten von Linsen, um einzelne Knoten genauer untersuchen zu können oder Knoten in der näheren Umgebung des zu untersuchenden Knoten genau zu analysieren.

### 2.1.3 Cue-basierte Ansätze

Bei Cue-basierten Techniken werden Elemente, die nicht mehr im sichtbaren Bereich (im folgenden als Viewport bezeichnet) liegen, durch Proxy-Elemente signalisiert. Diese sogenannten Proxy-Elemente können beispielsweise Ausschnitte von Kreisen (BR03) oder Dreiecken (GBGI08) sein. In der Arbeit von Gustafson et al. (GBGI08) werden Wedges vorgestellt. Es handelt sich dabei um Dreiecke, die so platziert werden, dass ihre Spitze beim Offscreen-Element liegt und die Grundseite ein Stück weit in den Viewport hineinragt.

Das ermöglicht dem User zu erkennen, in welcher Richtung das Objekt liegt und grob abzuschätzen, wie weit entfernt es vom aktuellen Viewport ist.

Im Gegensatz dazu werden bei Halo aus der Arbeit von Baudisch et al. (BR03) Kreise verwendet, deren Mittelpunkt auf Offscreen-Objekte gelegt werden. Der Radius des Kreises wird dabei so gewählt, dass der Kreis, ähnlich wie bei Wedges, ein Stück weit im Viewport erscheint.



Abbildung 5: Links: Offscreen-Objekte mit Halo dargestellt.

Rechts: Die gleichen Offscreen-Objekte mit Wedges repräsentiert. Quelle: (GBGI08)

Auch wenn beide Ansätze dem User ermöglichen, die ungefähre Position von Offscreen-Elementen zu ergreifen, empfehlen sich diese Visualisierungstechniken nicht, wenn der User Details der Offscreen-Elemente wissen möchte.

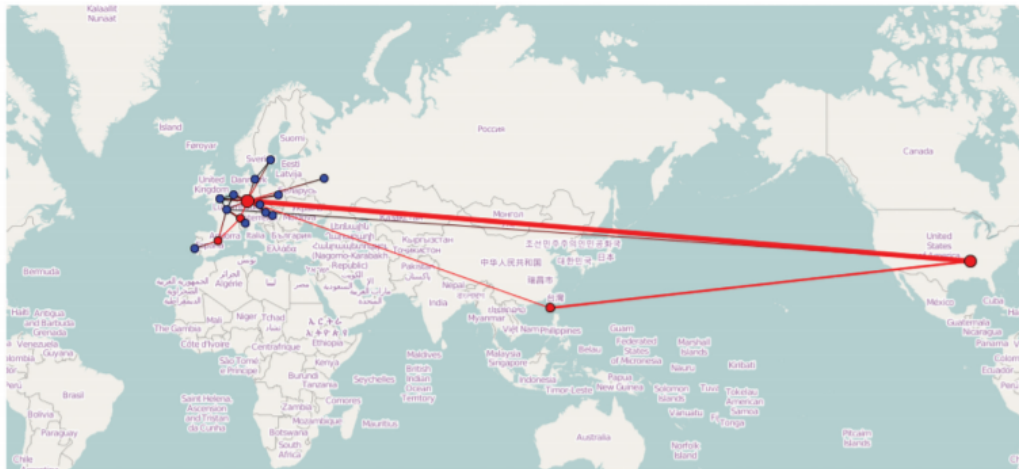
## 2.2 Offscreen-Visualisierung

Bei der Visualisierung von Offscreen-Objekten, wie beispielsweise in der Arbeit von Jäckle et al. (JKK15) beschrieben, geht es darum Objekte - in dieser Arbeit handelt es sich bei den Objekten um Knoten, wie im Falle von Knoten-Kanten-Diagrammen üblich ist - mit Hilfe von Proxy-Elementen zu visualisieren. Typischerweise werden dem Viewport dabei kleine Elemente hinzugefügt, die dem Nutzer möglichst viele Informationen über Knoten geben, die sich außerhalb des Viewports befinden und deshalb nicht an der eigentlich Position gezeigt werden können. Durch Interaktion mit Proxy-Elementen lassen sich Offscreen-Knoten genauer untersuchen um weitere Informationen zu lesen.

Der Ansatz für Offscreen-Visualisierung wird nicht nur für Graphen mit geographischem Bezug verwendet, sondern lässt sich, wie in der Arbeit von Frisch et al. (FD13) für UML-Diagramme, auch auf andere Knoten-Kanten Diagrammen übertragen.

Brodkorb et al. (BKA<sup>+</sup>16) wenden diesen Ansatz an, um Graphen mit geographischem Bezug genauer zu explorieren. Dabei werden Proxy-Elemente genutzt, um sowohl On- als auch Offscreen-Elemente zu repräsentieren.

Bei einer niedrigen Zoomstufe lassen sich Knoten, die nah beieinander liegen, und Details in ihrer Umgebung, oftmals nur schwer erkennen. Solche Regionen werden ebenfalls mit Proxy-Elementen, welche als on-Screen-Insets bezeichnet werden, versehen. Der Nutzer kann on-Screen-Insets durch Interaktion genauer untersuchen, wodurch detailliertere Fenster erscheinen, die den Kartenauschnitt des gewählten Insets mit einer erhöhten Zoomstufe enthalten (siehe Abbildung 6).



(a)



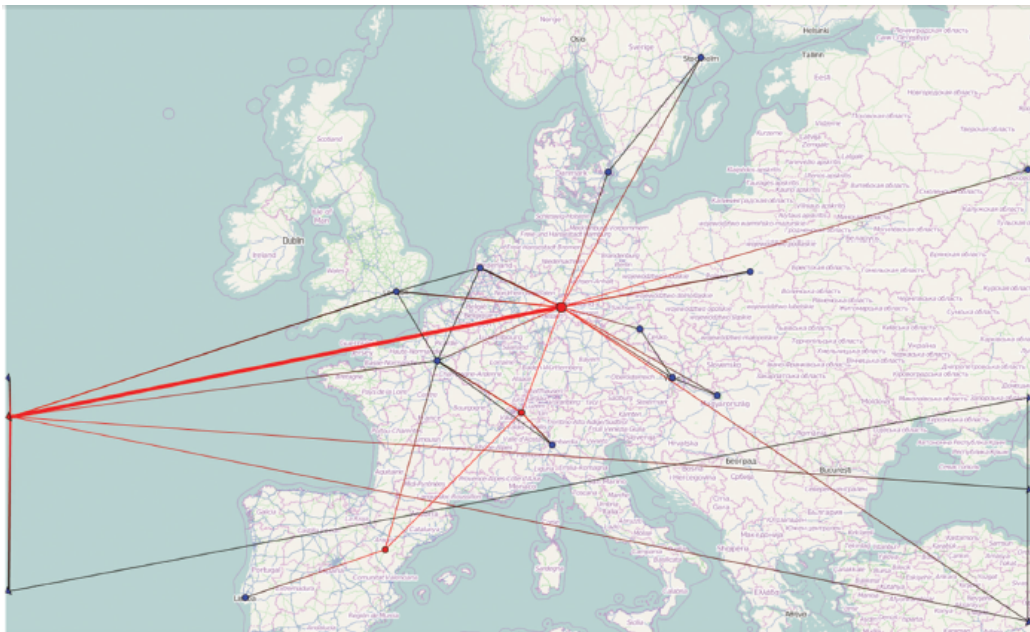
(b)

Abbildung 6: a) Proxy-Elemente werden als rote Knoten dargestellt. b) Der rote Inset für Orte in Nord- und Mittelamerika ist geöffnet und zeigt genauere Informationen an. Quelle: (BKA<sup>+</sup>16)

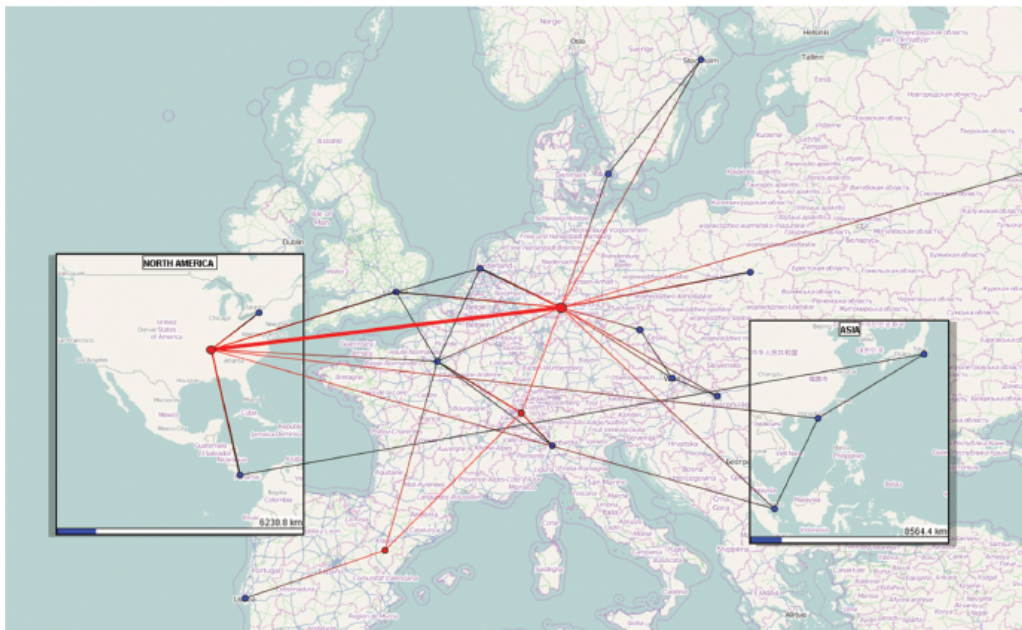
Für die in Abbildung 6 a) gewählte Zoomstufe lassen sich Details für Knoten in Europa nur schwer erkennen. In Abbildung 6 b) ist der Inset für den roten Knoten, der sich im Zentrum Deutschlands befindet, geöffnet. Ein Fenster, das die mit Knoten bedeckte Fläche in Europa überlagert, zeigt einen Kartenausschnitt mit den überdeckten Knoten auf eine höhere Zoom-

stufe, wodurch Details besser erkennbar sind. Das Fenster enthält außerdem einen Schriftzug mit der Aufschrift Europa, das die durch das Fenster repräsentierte Region beschreibt. Am unteren Rand des Fensters ist ein blauer Balken angezeigt, der dessen Zoomstufe wiedergibt.

Knoten, die sich außerhalb des Viewports befinden, werden ebenfalls durch sogenannte Offscreen-Insets angedeutet. Sie werden typischerweise am Rand des Viewports in Richtung des off-Screen-Knotens platziert und lassen sich, ähnlich wie bei on-Screen-Insets mit Hilfe von Interaktion genauer untersuchen (siehe Abbildung 7).



(a)



(b)

Abbildung 7: a) Proxy-Elemente für Offscreen-Elemente sind am Rande platziert. b) Zwei Offscreen-Insets für Asien und Nordamerika werden angezeigt. Quelle: (BKA<sup>+</sup>16)

Welche Orte Offscreen-Insets repräsentieren, lässt sich nur mit Hilfe von Interaktion feststellen. Das im nächsten Kapitel vorgestellte Konzept beruht auf einem ähnlichen Offscreen-Visualisierungsansatz wie der von Brodkorb et al., adressiert dabei aber stärker das Problem, das viel Nutzerinteraktion benötigt wird, um Offscreen-Knoten genauer zu untersuchen. Das Ziel ist es deshalb, eine graphische Repräsentation des Knoten-Kanten-Diagramms zu liefern, die möglichst wenig, im besten Fall sogar keine, Interaktion verlangt.





### 3 Konzept

Obwohl es bereits viele unterschiedliche Techniken gibt, um Netzwerke mit Daten mit geographischen Bezug zu explorieren, wird hier eine neue Technik vorgestellt. Diese wird im Rahmen dieser Arbeit zuerst konzeptioniert, implementiert und abschließend evaluiert. Im Vordergrund soll vor allem die schrittweise Exploration des Graphen stehen.

Um einen Knoten (beziehungsweise einen Ort) genauer zu untersuchen, muss dieser ausgewählt werden, woraufhin alle seine ein- und ausgehenden Verbindungen zu anderen Knoten, die sowohl regional als auch transkontinental verteilt liegen können, visualisiert werden. Vorher eingeblendete Knoten und Verbindungen werden dabei ausgeblendet.

#### 3.1 Definition und Schreibweise

Sei  $\mathcal{P}$  die Menge aller geographischen Orte mit den Elementen  $\{p_0, p_1, \dots, p_n\}$ , von den Informationen aus- oder eingehen. Ein  $p_i \in \mathcal{P}$  enthält dabei folgende Informationen:

- $p_i.city$  (der Städtename des geographischen Ortes)
- $p_i.lat \in [-90, 90]$  (Breitengrad)
- $p_i.lon \in [-180, 180]$  (Längengrad)

Für jedes  $p_i$  gibt  $f : \mathcal{P} \rightarrow \mathcal{D}^{|\mathcal{P}|}$  an, welche Richtung die Relation hat. Dabei sei  $\mathcal{D} = \{none, from, to, both\}$  die Menge der möglichen Richtungen. Hat der  $k$ -te Eintrag des resultierenden Vektors  $f(p_i) = \{d_0, d_1, \dots, d_{|\mathcal{P}|}\}^T$ , also  $d_k := f(p_i)_k$  den Wert *none*, bedeutet dies, dass von  $p_i$  nach  $p_k$  keine Informationen fließen. Für  $d_k = to$  fließen Informationen von  $p_i$  nach  $p_k$ , für  $d_k = from$  fließen die Informationen von  $p_k$  nach  $p_i$ , und für  $d_k = both$  fließen sowohl Informationen von  $p_i$  nach  $p_k$  als auch von  $p_k$  nach  $p_i$ .

Im Folgenden sei  $p_{curr}$  der aktuell ausgewählte geographische Ort, den der Nutzer genauer explorieren möchte, womit  $f_{p_{curr}} := f(p_{curr})$  für alle Einträge  $\neq none$  alle Knoten beschreibt, die mit  $p_{curr}$  in direkter Relation stehen. Die Menge, die alle Orte enthält, zu denen  $p_{curr}$  eine direkte Relation hat, wird im Folgenden als  $\mathcal{P}_{curr}$  bezeichnet.

### 3.2 Beschreibung des Konzepts

Ein simpler Ansatz, bei dem lediglich Pfeile zwischen allen Knoten aus  $\mathcal{P}_{curr}$  und  $p_{curr}$  in eine Weltkarte eingezeichnet werden, kann zum Problem führen, dass sowohl regionale als auch interkontinentale Relationen nur schwer oder sogar gar nicht zu erkennen sind.

Folgendes Beispiel soll dieses Problem verdeutlichen:

Für den Ort  $p_{curr}$  wird ein Marker in eine Weltkarte gezeichnet und der Viewport dabei so gewählt, dass sich  $p_{curr}$  in dessen Mitte befindet. Für jedes  $p_i \in \mathcal{P}_{curr}$  wird ebenfalls ein Marker in dieselbe Weltkarte eingezeichnet und zwar an der Position  $[p_i.lat, p_i.lon]$ . Diese Orte werden schließlich mit Pfeilen verbunden. Diese zeigen dabei entsprechend der Werte aus  $f(p_{curr})$  zu  $p_{curr}$  (wenn  $f(p_{curr})_i = from$ ), zu  $p_i$  (wenn  $f(p_{curr})_i = to$ ) oder sowohl zu  $p_{curr}$  als auch  $p_i$  (wenn  $f(p_{curr})_i = both$ ).

In Abbildung 8 ist der Graph für  $p_{curr} = \text{Stuttgart}$  abgebildet, wobei dieser Ort ausgehende Relationen zu New York City, Rio de Janeiro, Vorkuta und Kallaalit Nunaat hat.

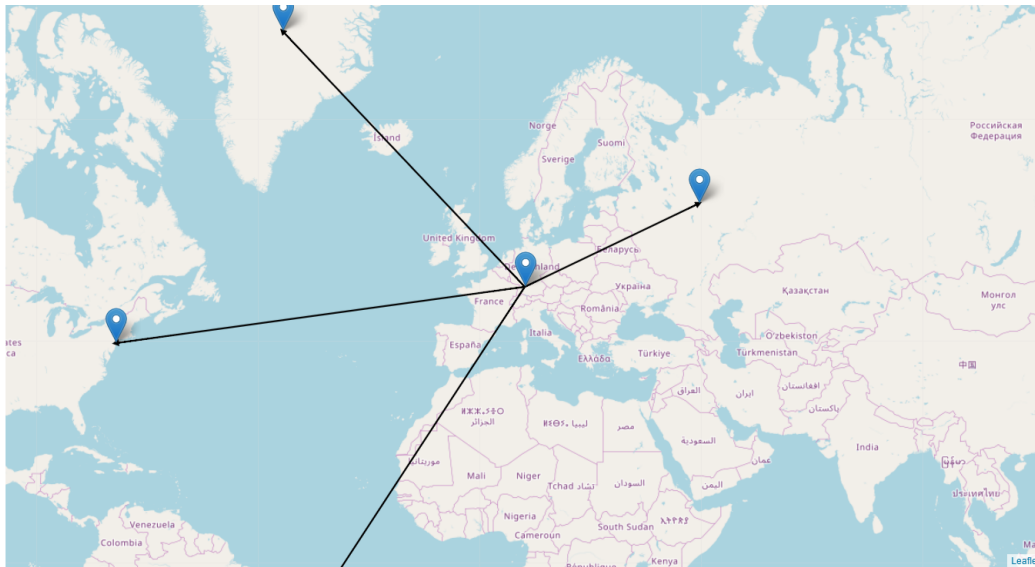


Abbildung 8: Simpler Ansatz zur Visualisierung von Informationsflüssen

Dass dieser simple Ansatz bereits für bei sehr wenigen Relationen zu Problemen führen kann, ist auf Abbildung 8 gut erkennbar. Wohin der Pfeil, der nach Süden geht, zeigt, ist nicht erkennbar, weil der sichtbare Kartenausschnitt den Zielort Rio de Janeiro nicht mehr enthält.

Dieses Problem ließe sich zwar beheben, in dem man weiter rauszoomt - zum Beispiel so lange, bis alle Punkte auf der Karte sichtbar sind. Es würde allerdings dazu führen, dass man Details, wie Städtegrenzen, Flüsse etc. nicht mehr erkennen kann.

Hat ein Ort außerdem nicht nur interkontinentale Relationen, wie im Beispiel aus Abbildung 8 (Europa zu Amerika und Südamerika), sondern würde zusätzlich regionale Relationen beinhalten, würden diese desto schlechter erkennbar sein je mehr man herauszoomt. Orte können sogar so nah beieinander liegen, dass sie sich stark überlagern und dadurch schlecht oder überhaupt nicht mehr erkennbar sind.

Eine der Hauptideen des Ansatzes, der in dieser Arbeit vorgestellt wird, ist die Verwendung von Minimaps, ähnlich wie sie beispielsweise auch in

Nachrichtensendungen Verwendung finden (siehe Abbildung 9).



Quelle:

[http://atlas.tagesschau.de/html/index.php?display\\_id=500000](http://atlas.tagesschau.de/html/index.php?display_id=500000)

Abbildung 9: Minimap zur Darstellung von Grönland und Island in Europa.

Minimaps sind Karten, die in kleinen Fenstern angezeigt und mit in die Hauptkarte eingezeichnet werden. Sie werden, wie in Kapitel 2.1.1, verwendet, um dem Nutzer einen Überblick über den gesamten Graphen zu liefern.

Ein weitere Verwendungszweck jedoch - welcher hier im Fokus steht - ist die Darstellung weit entfernter Orte. Minimaps sollen genau dann erscheinen, wenn ein Zielpunkt  $p_k$  nicht auf der Hauptkarte sichtbar ist. Der sichtbare Kartenausschnitt einer Minimap wird dann so gewählt, dass sich Orte, die mit Hilfe der Minimaps visualisiert werden soll, darin befinden und ebenfalls mit Hilfe von Markern signalisiert werden.

Um Berechnungen für Minimaps zu vereinfachen, wird im ersten Schritt

mit quadratischen Minimaps gearbeitet, wobei weitere Formen im späteren Verlauf der Arbeit vorgestellt und untersucht werden.

Eine quadratische Minimap  $m$  hat dabei folgende Eigenschaften:

- $m.pos$  (die Position des Zentrums der Minimap auf der Hauptkarte in Breitengrad und Längengrad [lat, lon] oder Pixelkoordinaten [x,y])
- $m.rad$  (Die Länge der Hälfte einer Seite der Minimap)
- $m.zoom$  (Das Zoomlevel der Minimap. Ein höheres Zoomlevel zeigt mehr Details an)

### 3.3 Positionierung der Minimap

Die dynamische und automatische Positionierung von Minimaps stellt sich als komplexe Aufgabe heraus. Erste Versuche Minimaps am Rande des Viewports zu positionieren, an dem der Pfeil, der von  $p_{curr}$  zu einem Offscreen-Ort  $p_{offscreen}$  zeigt, lieferten häufig gute Ergebnisse, führten in manchen Fällen aber zu Problemen. Der ursprüngliche Pfeil (von  $p_{curr}$  zu  $p_{offscreen}$ ) wird durch einen neuen Pfeil ersetzt. Dieser behält dabei die gleiche Richtung wie der ursprüngliche, zeigt aber nicht mehr auf den Zielort auf der Hauptkarte, sondern auf den Zielort, der auf der Minimap zu sehen ist. Um einen möglichst guten Überblick über den Zielort und dessen Umgebung zu kriegen, wird der Kartenausschnitt der Minimap so gewählt, dass der Zielort den Mittelpunkt des Kartenausschnitts ausmacht.

In Abbildung 10 ist dies für das gleiche Beispiel aus dem vorherigen Kapitel zu sehen und liefert akzeptable Ergebnisse.

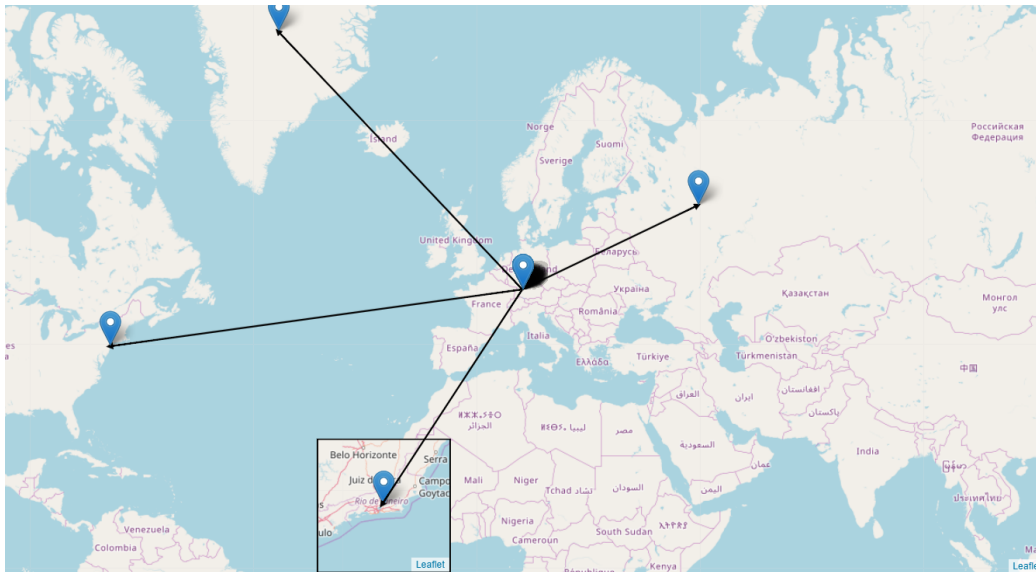


Abbildung 10: Die Minimap am Rande des Kartenausschnitts zeigt den Zielort Rio de Janeiro an.

Eine Minimap wird in Abbildung 10 dabei für Rio de Janeiro am unteren linken Rand gezeichnet, und der Pfeil zeigt auf das Zentrum der Minimap in dem sich der Marker befindet.

Das Problem, dass man weit entfernte Orte ohne Zoom oder Pan-Interaktion nicht erkennen kann, lässt sich mit dieser Methode auch dann beheben, wenn sich mehrere Zielorte nicht mehr im Viewport befinden. Das gilt allerdings nur solange die Orte in unterschiedlichen Richtungen liegen.

In Abbildung 11 werden für ein anderes Beispiel mit  $p_{curr} = \text{Rom}$  für vier der Zielorte (Tokio, Kapstadt, Rio de Janeiro, New York City und Kallaalit Nunaat) Minimaps gezeichnet, weil die Städte nicht mehr im gewählten Kartenausschnitt der Hauptkarte sichtbar sind. Es kommt zu keiner Überlagerung und alle Zielpunkte sind erkennbar, ohne dass User-Interaktionen erforderlich wären.

In vielen Fällen kann es allerdings dazu kommen, dass Minimaps andere Pfeile, Marker und weitere Minimaps überdecken.

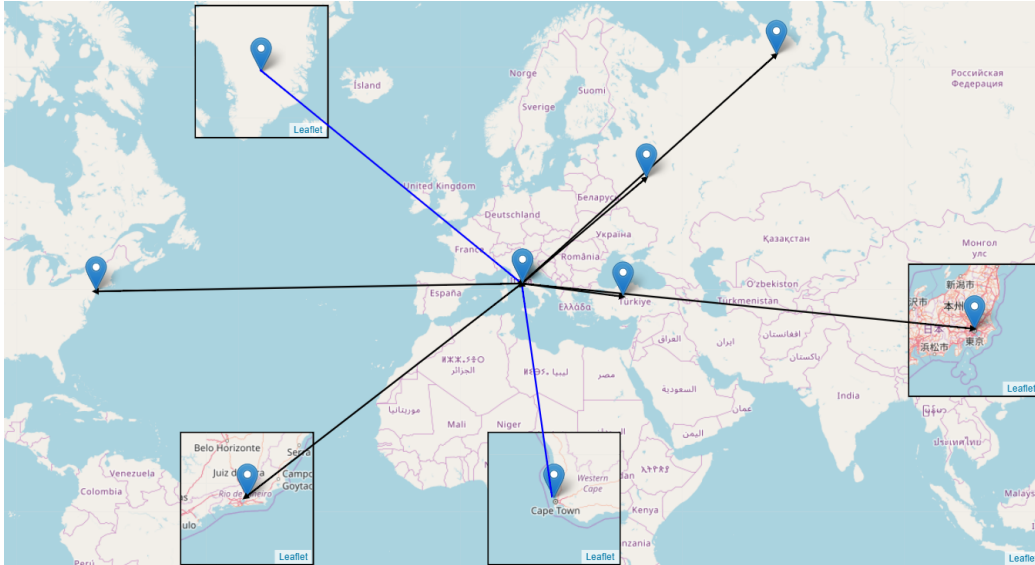


Abbildung 11: Fall ohne Überdeckung der Minimaps mit anderen Objekten

Minimaps überdecken Zielorte, die im sichtbaren Kartenbereich liegen, wenn sich ein solcher Zielort am Rande des Kartenausschnittes befindet und es einen weiteren Zielort gibt, der in der selben Richtung liegt, aber nicht mehr auf der Hauptkarte zu sehen ist.

Zur Überlappung zweier quadratischer Minimaps  $m_1$  und  $m_2$  kommt es, sobald der Winkel zwischen ihren Zielorten und  $p_{curr}$  so klein ist, dass die resultierende Mittelpunkte  $m_1.pos$  und  $m_2.pos$  der zugehörigen Minimaps einen Abstand haben, der kleiner als die Summe der Hälfte einer Seitenlänge der jeweiligen Minimaps ist:

$$\sqrt{(m_1.pos.x - m_2.pos.x)^2 + (m_1.pos.y - m_2.pos.y)^2} < \frac{m_1.rad + m_2.rad}{2}$$

$m_1$  und  $m_2$  liegen in solchen Fällen so nah beieinander, dass sich die Minimaps überlagern.

In Abbildung 12 befinden sich die Zielorte Rom, New York City und Rio de Janeiro außerhalb des Viewports, weshalb sie durch drei Minimaps repräsentiert werden. Die Minimap für Rom überdeckt dabei fast komplett

die Minimap für New York City, wodurch für den User die Verbindung zu New York City nicht mehr erkennbar ist.

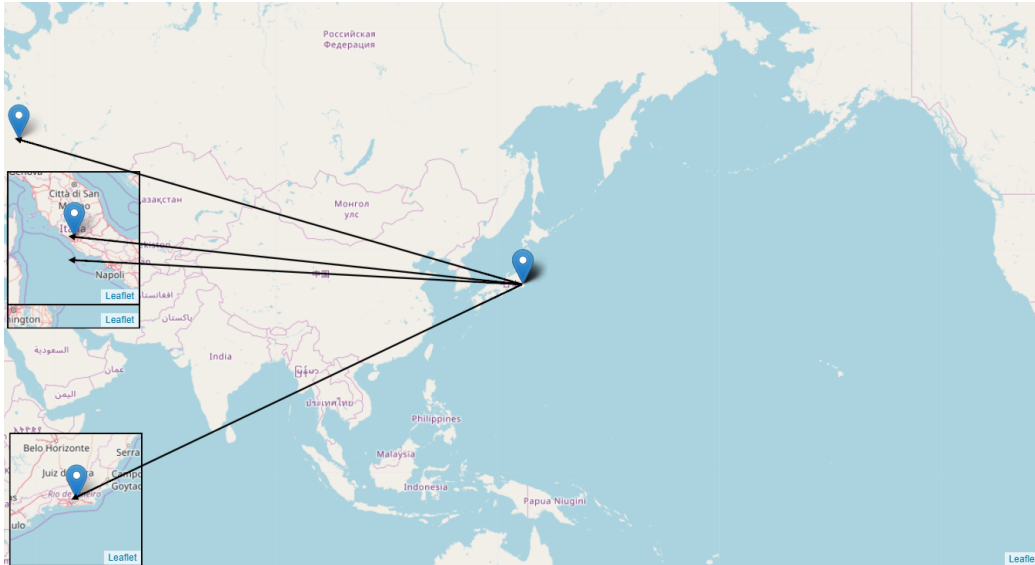


Abbildung 12: Fall mit Überdeckung von zwei Minimaps.

### 3.3.1 Kräftebasierter Ansatz zur Darstellung von Knoten-Kanten Diagrammen

Das Problem Minimaps so auszurichten, dass Sie sich nicht überlagern und ästhetisch aussehen, lässt sich auf ein Graphen-Layoutproblem übertragen. Dabei geht es darum, Knoten eines Graphen so zu platzieren, dass sowohl die Knoten als auch ihre Verbindungen, also ein- und ausgehende Kanten, erkennbar werden.

Ein Graph  $G = (V, E)$  besteht aus der Menge  $V$ , was der Menge der Knoten entspricht, und der Menge der Kanten  $E$  (eine Kante beschreibt dabei die Verbindung zwischen genau zwei Knoten  $\in V$ ).

Wann ein Graph als ästhetisch empfunden wird, wurde bereits von vielen Forschern untersucht. Einer der bekanntesten Arbeiten auf diesem Gebiet ist



die von Fruchterman et al. (FR91). In ihr wird beschrieben, dass ein Graph als ästhetisch empfunden wird, wenn er folgende Eigenschaften erfüllt:

- Gleichverteilung der Knoten im Bild
- Minimale Überschneidung von Kanten
- Einheitliche Kantenlänge
- Reflektierung inhärenter Symmetrien
- Konformität zum Bild

Für die Berechnung von Graph-Layouts gibt es bereits viele verschiedene Algorithmen, wie beispielsweise in der Arbeit von Eades (DBETT94) oder beim Algorithmus von Kamada und Kawai (KK<sup>+</sup>89) beschrieben.

Auch wenn die oben erwähnten Merkmale sich teilweise widersprechen (um Knoten beispielsweise gleichzuverteilen, kann es passieren, dass keine einheitliche Kantenlänge mehr eingehalten werden kann), liefern solche Algorithmen sehr gute Ergebnisse.

In dieser Arbeit wurde für die Positionierung der Minimaps der Algorithmus von Fruchterman-Reingold (FR91) verwendet. Die Grundidee des Algorithmus' basiert auf folgende Prinzipien:

1. Knoten, die über Kanten direkt verbunden sind, sollen nah beieinander gezeichnet werden.
2. Knoten sollen im Allgemeinen einen gewissen Abstand voneinander haben.

Wie nah beziehungsweise weit voneinander Knoten gezeichnet werden, hängt dabei von der Anzahl Knoten und dem vorhanden Platz ab. Die optimale Distanz  $k$  zwischen Knoten wird folgendermaßen berechnet:

$$k = C \sqrt{\frac{area}{|V|}}$$

$C$  ist eine Konstante, die experimentell bestimmt wird und  $|V|$  die Anzahl Knoten.

Der Algorithmus von Fruchterman-Reingold (siehe Pseudocode in Abbildung 13 aus (FR91)) berechnet iterativ für jeden Knoten die Abstoßungs- und Anziehungskraft zu den anderen Knoten (die Anziehungskraft herrscht nur zu Knoten mit denen der zu untersuchende Knoten über eine Kante verbunden ist). Die Anziehungskraft wird berechnet, um dem ersten der beiden Prinzipien zu folgen, während die Abstoßungskraft zur Einhaltung des zweiten Prinzips benötigt wird. Nachdem alle Kräfte, die auf einen Knoten wirken, summiert sind (abstoßende und anziehende Kräfte wirken in die entgegengesetzte Richtung), wird der Knoten in die Richtung der resultierenden Kraft bewegt. Um mögliches Oszillieren von Knoten zu vermeiden, wird die Bewegung in Richtung des resultierenden Vektors mit einer Temperatur  $t$ , die nach jedem Iterationsschritt verringert wird, multipliziert. Das bewirkt außerdem, dass sich Knoten zu Beginn stärker verschieben, was zu verbesserter Performance führen kann, und gegen Ende des Algorithmus nur noch kleine Anpassungen vorgenommen werden. Der Algorithmus stellt außerdem sicher, dass Knoten, die eine so starke Verschiebung erfahren, dass sie außerhalb des Viewports platziert werden würden, stattdessen am Rande des Viewports platziert werden.

```

area := W * L; { W and L are the width and length of the frame }
G := (V, E); { the vertices are assigned random initial positions }
k :=  $\sqrt{\text{area}/|V|}$ ;
function  $f_r(z)$  := begin return  $x^2/k$  end;
function  $f_a(z)$  := begin return  $k^2/z$  end;

for i := 1 to iterations do begin
  { calculate repulsive forces }
  for v in V do begin
    { each vertex has two vectors: .pos and .disp }
    v.disp := 0;
    for u in V do
      if (u # v) then begin
        {  $\delta$  is short hand for the difference }
        { vector between the positions of the two vertices }
         $\delta := v.pos - u.pos$ ;
        v.disp := v.disp + ( $\delta / |\delta|$ ) *  $f_r(|\delta|)$ 
      end
    end

    { calculate attractive forces }
    for e in E do begin
      { each edge is an ordered pair of vertices .v and .u }
       $\delta := e.v.pos - e.u.pos$ 
      e.v.disp := e.v.disp - ( $\delta / |\delta|$ ) *  $f_a(|\delta|)$ ;
      e.u.disp := e.u.disp + ( $\delta / |\delta|$ ) *  $f_a(|\delta|)$ 
    end
  end

  { limit the maximum displacement to the temperature t }
  { and then prevent from being displaced outside frame }
  for v in V do begin
    v.pos := v.pos + (v.disp / |v.disp|) * min(v.disp, t);
    v.pos.x := min(W/2, max(-W/2, v.pos.x));
    v.pos.y := min(L/2, max(-L/2, v.pos.y))
  end
  end
  { reduce the temperature as the layout approaches a better configuration }
  t := cool(t)
end

```

Abbildung 13: Pseudocode für den Algorithmus von Fruchterman-Reingold

Der Algorithmus zum Ausrichten von Knoten und Kanten wird für diesen Fall nicht, wie es eigentlich üblich ist, auf alle Knoten und Kanten angewandt. Stattdessen werden lediglich Knoten (und deren Kanten bewegt), die sich nicht mehr im aktuellen Viewport befinden und somit durch Minimaps

repräsentiert werden; es geht schließlich darum eine geeignete Position für Minimaps zu finden.

Der Algorithmus von Fruchterman-Reingold benötigt eine Vielzahl von Iterationsdurchläufen (*iterations*), um sich der finalen Positionierung der Knoten zu nähern.

In Abbildung 14, 15 und 16 wird für ein Beispiel, das als  $p_{curr}$  Rio de Janeiro und als Zielorte New York City, Reykjavik, München, Ankara und Tokio hat, das Ergebnis des Algorithmus von Fruchterman-Reingold, das für diese Arbeit so angepasst wurde wie oben beschrieben, für eine ansteigende Anzahl Iterationen gezeigt.

In Abbildung 14 sieht man das Ergebnis nach 100 Iterationen. Weil die Zielorte, für die die Minimaps positioniert werden sollen, sich alle nördlich von  $p_{curr}$ , also Rio de Janeiro, befinden, erscheinen alle Minimaps noch auf der oberen Hälfte des Viewports. Die Minimaps für Tokio und Ankara überlappen sich weiterhin in der oberen rechten Ecke.

Nach 500 Iterationen, siehe Abbildung 15, haben sich die Minimaps bereits stärker über den verfügbaren Bereich verteilt. Der Abstand zwischen allen Minimaps hat sich vergrößert, und die Minimap für Ankara befindet sich sogar am unteren Rand des Viewports. Das liegt daran, dass alle Minimaps mit dem zentralen Knoten (Rio de Janeiro) verbunden sind, untereinander allerdings keine weiteren Verbindungen existieren. Die Abstoßungskraft, die zwischen allen Minimaps herrscht, führt dazu, dass sie immer weiter auseinander gehen.

Nach circa 1000 Iterationen hat der Algorithmus die finale Position der Minimaps berechnet. Abbildung 16 zeigt diese für genau 1000 Iterationen. Das Ergebnis erfüllt dabei viele der Kriterien, die einen Graphen als ästhetisch erscheinen lassen. Vor allem die symmetrische Form der Minimaps, die einem Stern ähnelt, ist hier besonders auffällig. Die Winkel zwischen den Minimaps sind sehr groß und keine der Kanten überschneiden sich.

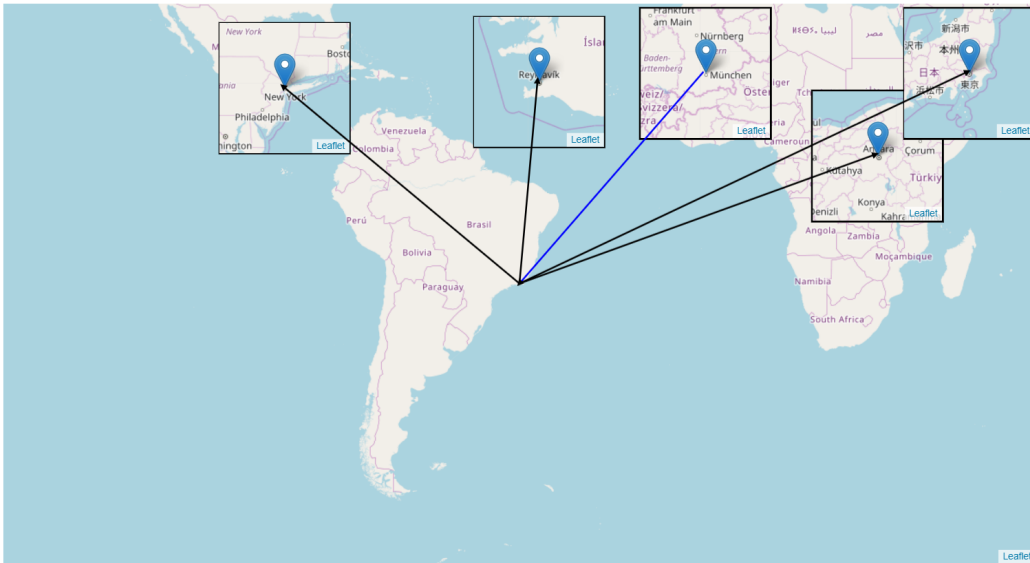


Abbildung 14: Ergebnis nach 100 Iterationen

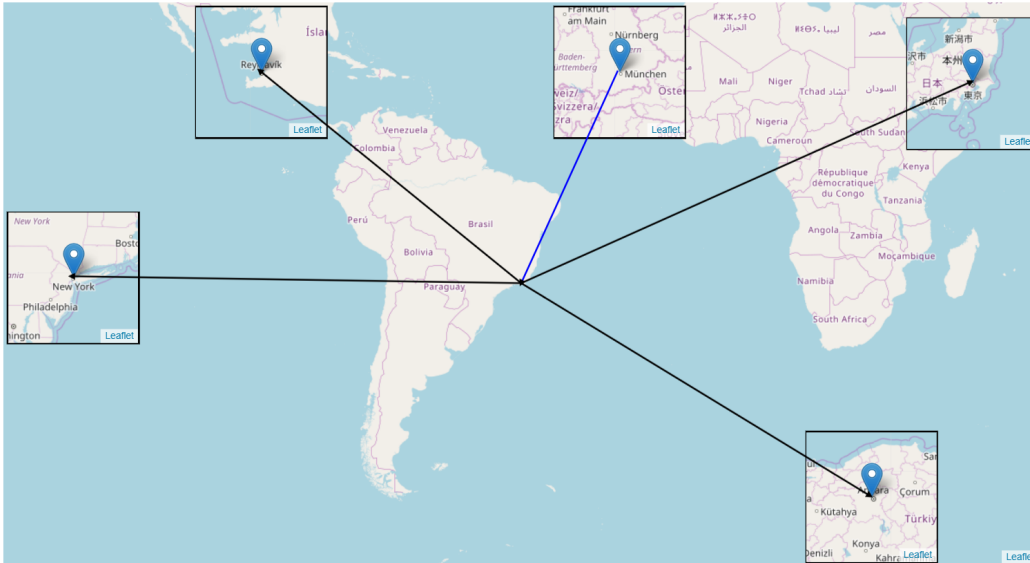


Abbildung 15: Ergebnis nach 500 Iterationen

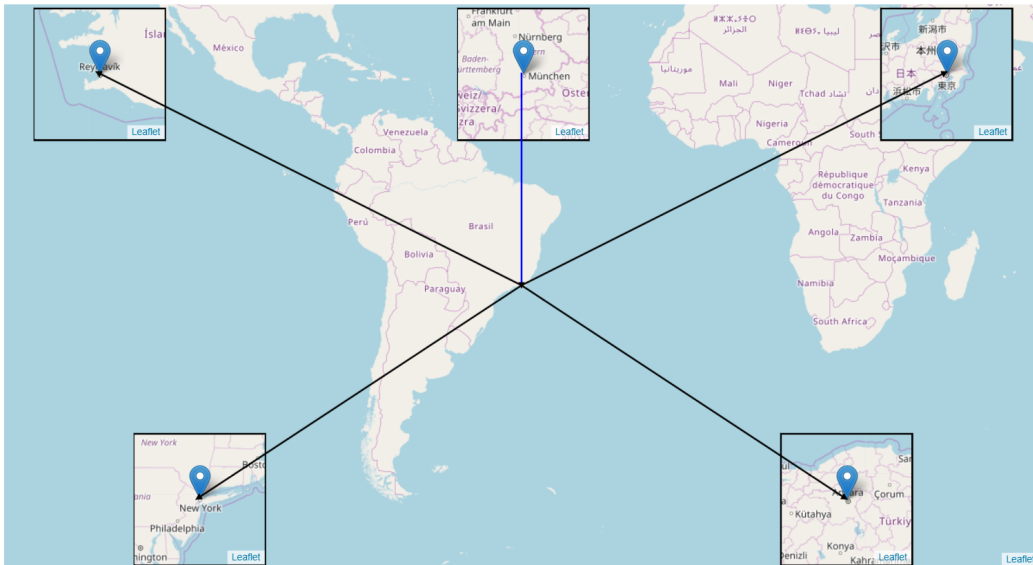


Abbildung 16: Ergebnis nach 1000 Iterationen

Obwohl die resultierende Platzierung der Minimaps viele der Kriterien, die einen Graphen als ästhetisch erscheinen lassen, erfüllt, sind weitere Anpassungen notwendig.

Das Hauptproblem bei einer solchen Platzierung ist, dass der ursprüngliche Ort, an dem eine Minimap platziert werden sollte, nämlich in Richtung des Ortes, der in der Minimap angezeigt wird, nicht in Betracht gezogen wird. Im Beispiel aus Abbildung 14, 15 und 16 wird die Minimap für New York City in der unteren linken Bildhälfte platziert. Genau diese Platzierung erweckt den Eindruck, dass die Stadt, die durch die Minimap repräsentiert wird, sich südwestlich von der Stadt befindet, die genauer untersucht werden soll. Für diesen Fall würde man erwarten, dass eine Minimap, die sich unten links im Viewport befindet, eine Stadt anzeigt, die südwestlich von Rio de Janeiro liegt.

Das gleiche Problem lässt sich auch bei den anderen Minimaps beobachten. Die Minimap für Ankara beispielsweise ist in der unteren rechten Bildhälfte platziert. Hier würde man ebenfalls erwarten, dass ein Ort im

Südosten von Rio de Janeiro, also zum Beispiel in der Antarktis, durch die Minimap repräsentiert werden würde; stattdessen enthält sie einen Ort aus Südosteuropa.

### 3.3.2 Anpassung der Kräfte

Sowohl der gewöhnliche Algorithmus von Fruchterman-Reingold als auch die für unseren Fall angepasste Version können die Position der Knoten so stark verändern, dass Orte, die sich beispielsweise ursprünglich im Osten befanden und somit in der rechten Hälfte des Viewports erwartet werden, hinterher an einer komplett anderen Position erscheinen.

Damit der ursprüngliche Bezug und die Position des Ortes nicht außen vor gelassen werden, führen wir eine neue Kraft  $F_{a_{orig}}$  ein. Bei dieser Kraft handelt es sich um eine Anziehungskraft, die eine Minimap, in die Richtung der Position des Ortes, den sie repräsentiert, bewegt.

- Bestehende Kräfte
  - $F_{r_{map}}$ : Abstoßung von anderen Minimaps mit Radius  $r$
  - $F_{r_{node}}$ : Abstoßung von anderen Knoten im aktuellen Viewport
  - $F_a$ : Anziehung zum zentralen Knoten
- Neue Kraft
  - $F_{a_{orig}}$  Anziehung an ursprüngliche Position des zu repräsentierenden Ortes

Um den Einfluss der einzelnen Kräfte auf das Gesamtmodell anzupassen, wird jede Kraft mit einem Faktor multipliziert. Durch Veränderung der Faktoren ( $k_{map}$ ,  $k_{node}$ ,  $k_a$  und  $k_{orig}$  sind die Faktoren für  $F_{r_{map}}$ ,  $F_{r_{node}}$ ,  $F_a$  und  $F_{a_{orig}}$ ) lassen sich die Kräfteverhältnisse leicht regulieren.

Die Berechnung der Anziehungskraft  $F_{a_{orig}}$  an den ursprünglichen Ort, den eine Minimap repräsentiert, wird ähnlich wie die Anziehung zum zentralen Knoten berechnet. Der folgende Pseudocode zeigt, wie diese Berechnung für den aktuellen Knoten  $p_i$ , für den die wirkenden Kräfte berechnet werden sollen, aussieht:

```

function  $F_{a_{orig}}(p_i)$ 
   $\vec{o} \leftarrow p_i.\text{orig.pos}$ 
   $\vec{c} \leftarrow p_i.\text{pos}$ 
   $\Delta pos \leftarrow |\vec{o} - \vec{c}|$ 
  return  $\frac{\Delta pos^2}{k} \cdot k_{orig}$ 
end function

```

Die neu eingeführte Kraft bewirkt eine starke Veränderung der Positionierung der Minimaps. Das Problem, dass Minimaps den Bezug zur ursprünglichen Position des Ortes, den sie repräsentieren, verlieren, lässt sich immer stärker reduzieren, indem  $k_{orig}$  immer höher gewählt wird. Wird  $k_{orig}$  allerdings zu groß gewählt, verlieren die restlichen Kräfte ihren Einfluss auf das gesamte Modell. In Abbildung 17 ist für das gleiche Beispiel aus Abbildung 14, 15, 16 das Ergebnis nach 100 Iterationen zu sehen.



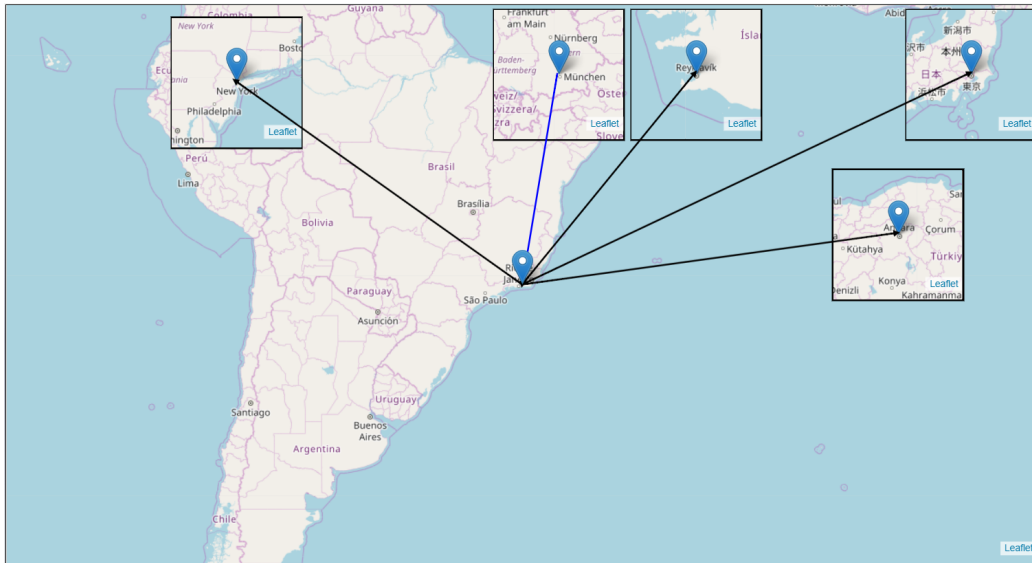


Abbildung 17: Ergebnis mit neuem Kräftenmodell nach 100 Iterationen

Die Positionierung der Minimaps ähnelt der aus Abbildung 14 nach ebenfalls 100 Iterationen. Der Unterschied, den die neu eingeführte Kraft bewirkt, wird erst nach mehreren Iterationen deutlich. In Abbildung 18 ist das Ergebnis nach 1000 Iterationen abgebildet.

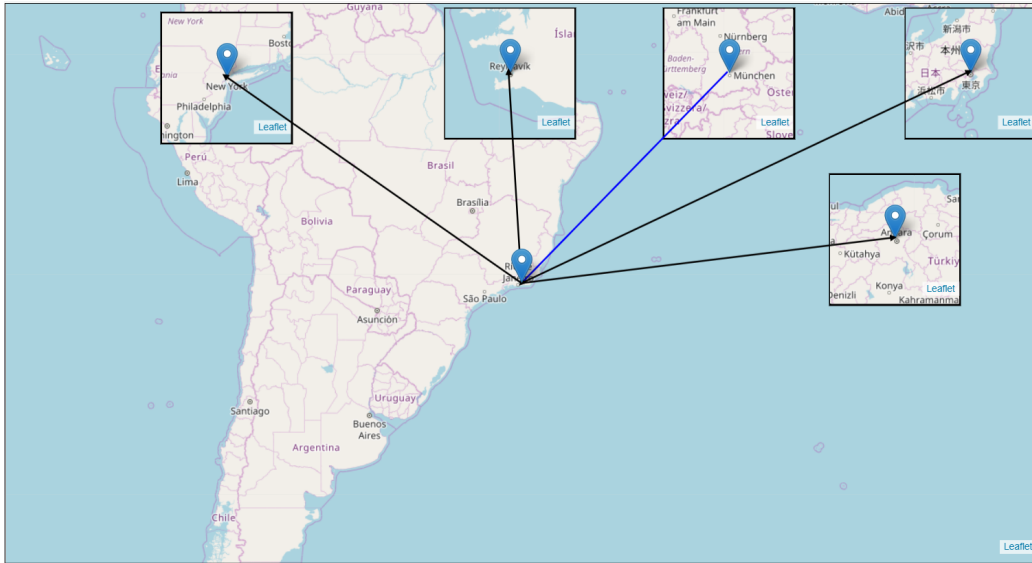


Abbildung 18: Ergebnis mit neuem Kräftemodell nach 1000 Iterationen

Da die finale Position der Minimaps bereits nach weniger als 1000 Iterationen erreicht wurde, würde eine Erhöhung der Iterationen die Position der Minimaps nicht weiter verändern. Anders als in Abbildung 16 befindet sich keine Minimap in der unteren Hälfte des Viewports. Stattdessen befinden sich Minimaps, die Städte repräsentieren, die im Osten des zentralen Ortes liegen, auf der rechten Hälfte des Viewports. Die Minimap für Ankara, die ohne den Einfluss der Kraft  $F_{a_{orig}}$  noch im unteren rechten Bereich des Viewports gezeichnet wurde, befindet sich stattdessen in der rechten Hälfte leicht oberhalb der Mitte des Viewports. Da sich die Stadt Ankara nordöstlich von Rio de Janeiro befindet, hat sich die Positionierung dieser Minimap verbessert.

Das Gleiche gilt auch für die anderen Minimaps, wie zum Beispiel für die, die New York repräsentiert. Diese wurde im Kräftemodell ohne  $F_{a_{orig}}$  in der unteren linken Ecke des Viewports platziert; durch den Einfluss von  $F_{a_{orig}}$  wird sie stattdessen in der Nähe der oberen linken Ecke des Viewports gezeichnet, was einer korrekten Platzierung für Orte wie New York City, die sich im Nordwesten von Rio de Janeiro befinden, entspricht.

Insgesamt führt das neue Kräftemodell zu einer deutlichen Verbesserung der Platzierung der Minimaps.

### 3.3.3 Überlagerung von Minimaps

Die Kräfte des neuen Kräftemodells wirken teilweise gegeneinander - Minimaps sollen sich voneinander abstoßen und gleichzeitig aber auch in Richtung des ursprünglichen Ortes der Minimaps bewegt werden - und führen deshalb zu einem Überlagerungsproblem, das es vor Einführung von  $F_{aorig}$  nicht gab.

Sobald mehrere Orte, die sich außerhalb des Viewports befinden, von  $p_{curr}$  aus gesehen aber in der gleichen Richtung liegen, kann es zur Überlagerung solcher Minimaps kommen (siehe Abbildung 19).

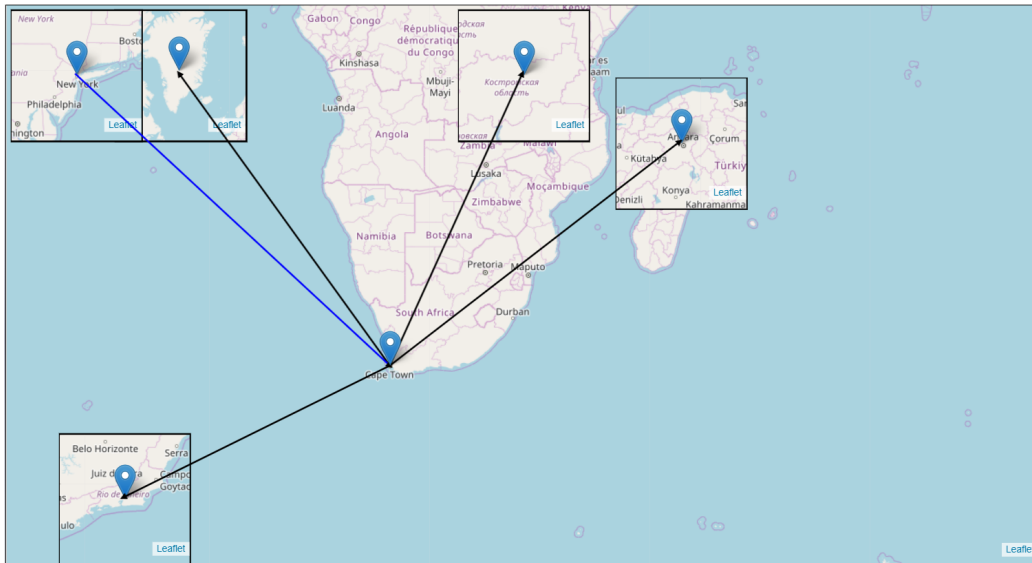


Abbildung 19: Überlappung von Minimaps auf Grund des gewählten Kräftemodells

In Abbildung 19 ist für den Ort Kapstadt als  $p_{curr}$  die Relation zu den Orten New York City, Kallaalit Nunaat, Vorkuta, Ankara und Rio de Janeiro

dargestellt. Zur Berechnung der Position der Minimaps wurde dabei das Kräftemodell aus Kapitel 3.3.2 verwendet.

Grund für die Überlagerung der Minimaps für New York City und Kallaalit Nunaat ist der Einfluss von  $F_{acurr}$ , der die beiden Minimaps in die obere linke Ecke des Viewports zieht. Dies ließe sich zwar beheben, indem der Faktor  $k_{orig}$  für  $F_{acurr}$  verringert, beziehungsweise der Faktor  $k_{map}$  für die Abstoßungskraft  $F_{rmap}$  zwischen Minimaps erhöht wird. Wenn diese Koeffizienten global geändert werden, hätte dies aber einen Einfluss auf alle Minimaps, insbesondere auch auf solche, die eigentlich gut positioniert sind. Eine globale Anpassung der Kräfte könnte zwar für einen Fall das Problem beheben, in einem anderen Fall aber dazu führen, dass sich die Situation verschlechtert.

Obwohl eine dynamische und lokale Anpassung der Koeffizienten  $k_{map}$ ,  $k_{node}$ ,  $k_a$  und  $k_{orig}$ , die sich nur auf Minimaps auswirken soll, die sich überlagern, das Problem beheben würde, wird es stattdessen mit Hilfe von Kollisionsbehebungen beseitigt. Dieser Ansatz wird auch in der Arbeit von Bordkorb et al. (BKA<sup>+</sup>16) verwendet, welcher wiederum auf den Vorarbeiten von Luboschik et al. (LSC08) basiert.

In Abbildung 20 aus (BKA<sup>+</sup>16) sind auf dem linken Bild drei unterschiedliche Arten von Kollisionen mit Insets (rote, grüne und blaue Quadrate) dargestellt, die bei der Positionierung der Minimaps aus dieser Arbeit ebenfalls entstehen können.

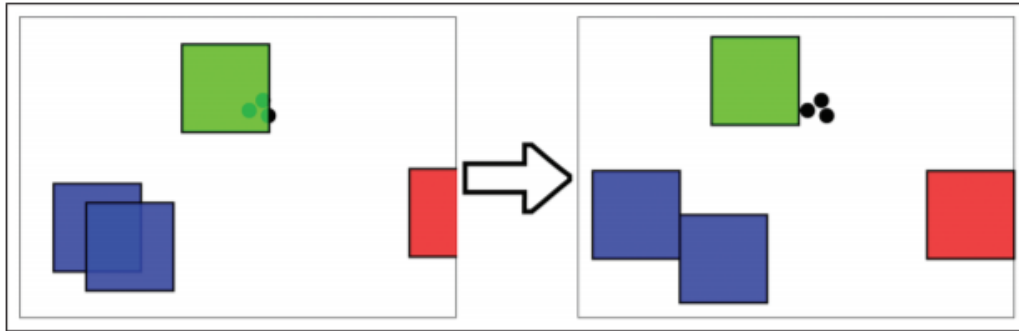


Abbildung 20: Grafik aus (BKA<sup>+</sup>16): Linkes Bild: Zustand vor Kollisionsbehebung. Rechtes Bild: Zustand nach Kollisionsbehebung

Die drei möglichen Kollisionsarten, die bei der Platzierung der Minimaps in dieser Arbeit entstehen können, lassen sich in Abbildung 20 wiederfinden:

1. Kollision zwischen zwei Minimaps (blaue Quadrate)
2. Kollision zwischen Minimaps und Markern im Viewport (grünes Quadrat als Minimap und schwarze Punkte als Marker)
3. Kollision zwischen Minimap und Viewport (rotes Quadrat)

Der Algorithmus 21, der bei Brodkorb et al. verwendet wird, überprüft für jedes Inset mögliche Kollisionen mit anderen Objekten und bewegt dabei schrittweise das kollidierende Inset weg vom Zentrum der Kollision. Beim Wegbewegen der Insets kann es zu neuen Kollisionen kommen, weshalb sich unter Umständen oszillierende Effekte beobachten lassen.

---

**Algorithm 1.** Occlusion prevention algorithm.

---

```
for  $l \in ll$  do
   $FF \leftarrow \phi$ 
   $FF \leftarrow FF \cup \{l(l) - C_l\} \sqrt{a^2 + b^2}$ 
  if  $l \cap S \neq \phi$  then
    moveToScreen( $l$ )
  end if
  for  $l_2 \in ll \setminus \{l\}$  do
    if  $l_2 \cap l \neq \phi$  then
       $M \leftarrow l \cap l_2$ 
      moveAwayFromIntersection( $l, M$ )
    end if
  end for
  for  $N \in NN$  do
    if  $N \cap l \neq \phi$  then
       $M \leftarrow l \cap N$ 
      moveAwayFromIntersection( $l, M$ )
    end if
  end for
  for  $\tilde{F} \in FF$  do
     $l = l + 0.5 \cdot \frac{F \cdot w(\tilde{F})}{w(\tilde{F}) + \sum_{\{\tilde{F} \in FF: \langle \tilde{F}, \tilde{F} \rangle < 0\}}$ 
  end for
end for
```

---

---

**Algorithm 2.** Function moveToScreen( $l$ ).

---

```
 $x_r \leftarrow c \left( x_l, \left[ \frac{w_l}{2}, \left( w_s - \frac{w_l}{2} \right) \right] \right)$ 
 $y_r \leftarrow c \left( y_l, \left[ \frac{h_l}{2}, \left( h_s - \frac{h_l}{2} \right) \right] \right)$ 
if  $C_l - C_r \neq 0$  then
   $FF \leftarrow FF \cup \{C_r - C_l\}$ 
end if
```

---

---

**Algorithm 3.** Function moveAwayFromIntersection( $l, M$ ).

---

```
 $\tilde{P} \leftarrow C_l - C_M$ 
 $s_x \leftarrow \sqrt{w_M^2 + \left( w_M \cdot \frac{y_{\tilde{P}}}{x_{\tilde{P}}} \right)^2}$ 
 $s_y \leftarrow \sqrt{h_M^2 + \left( h_M \cdot \frac{x_{\tilde{P}}}{y_{\tilde{P}}} \right)^2}$ 
 $FF \leftarrow FF \cup \left\{ \|C_l - C_M\| \frac{\|\tilde{P}\| + \min(s_x, s_y)}{2} \right\}$ 
```

---

Abbildung 21: Algorithmen aus Bordkorb et al., die für die Kollisionsbehebung verwendet werden

Das Prinzip der Kollisionsbehebung aus (BKA<sup>+</sup>16) lässt sich leicht auf diese Arbeit übertragen. Für jede Minimap, in (BKA<sup>+</sup>16) als Offscreen-Inset bezeichnet, wird eine mögliche Kollision mit anderen Minimaps, Markern, die Orte markieren, die sich im aktuellen Viewport befinden, und dem Viewport

überprüft. Im Falle einer Kollision wird die Minimap schrittweise von den anderen Objekten mit denen sie kollidiert, wegbewegt.

Da der Kartenausschnitt, nach Auswahl eines Knotens, so gesetzt wird, dass der zu explorierende Knoten im Zentrum des Viewports erscheint, wird in dieser Arbeit davon ausgegangen, dass Informationen im Zentrum des Viewports wichtiger sind als Informationen weiter außen, wie zum Beispiel am Rand des Viewports.

Aus diesem Grund werden Minimaps, um Kollisionen zu beheben, bevorzugt entlang des Randes des Viewports bewegt. Erst wenn eine Platzierung der Minimaps entlang des Randes zu keiner Kollisionbehebung führt, wird nach einer Lösung weiter innen im Zentrum des Viewports gesucht.

Damit entsteht folgende Strategie für die Bewegung der Minimaps, um Kollisionen zu beheben:

Bei einer Kollision von Minimaps am linken oder rechten Rand des Viewports werden Minimaps vertikal bewegt. Führt die Bewegung entlang der Vertikalen zu keiner Lösung, wird nach einer Lösung entlang der nächsten Vertikalen, die sich näher im Zentrum befindet, gesucht. Abbildung 22 zeigt einen Kollisionsfall mit vier Minimaps vor der Behebung. Sie sind als farbige Quadrate (gelb, rot, blau und grün) in der oberen rechten Ecke visualisiert.

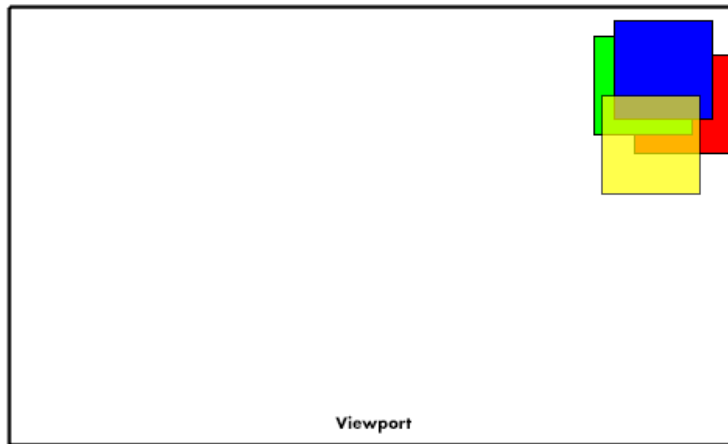


Abbildung 22: Kollision zwischen vier Minimaps

Die Platzierung der Minimaps nach Behebung der Kollision ist in Abbildung 23 gezeigt. Die rote Minimapa kann nicht mehr entlang der selben Vertikalen, auf der sich die blaue, gelbe und grüne Minimapa befinden, platziert werden, weshalb sie auf der nächsten Vertikalen links neben der blauen Minimapa platziert wird.

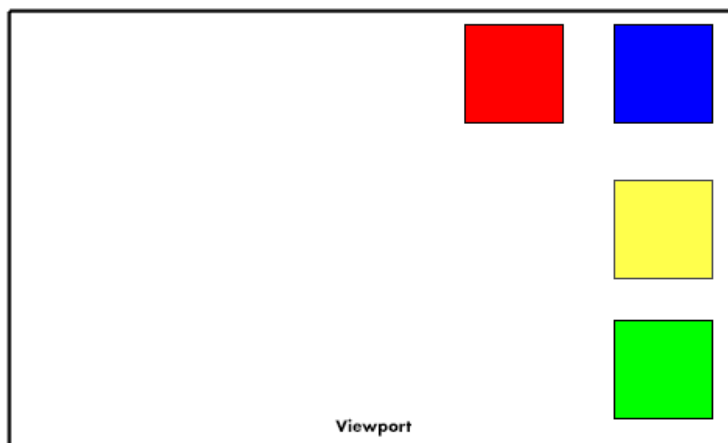


Abbildung 23: Platzierung der Minimaps nach Behebung der Kollision aus 22



Die gleiche Strategie wird ebenfalls für Minimaps am oberen oder unteren Rand des Viewports angewandt, wobei Minimaps nicht entlang der Vertikalen, sondern entlang der Horizontalen platziert werden, bevor nach einer Lösung näher im Zentrum gesucht wird.

Überlagern sich viele Minimaps im selben Bereich, führt der oben beschriebene Algorithmus dazu, dass Minimaps immer weiter von der eigentlichen Position, in der sie mit anderen Minimaps kollidieren, entfernt werden. Wenn sich beispielsweise drei Minimaps  $m_1$ ,  $m_2$  und  $m_3$  in der oberen rechten Ecke des Viewports überlagern, und beginnend mit  $m_3$  die Kollisionen für diese Minimap behoben werden sollen, so wird  $m_3$  so lange aus der oberen rechten Ecke wegbewegt, bis sie, ähnlich wie bei den blauen Quadraten in Abbildung 20, direkt neben  $m_2$  und  $m_3$  landet. Um die Kollisionen für  $m_2$  zu beheben, muss diese im ersten Schritt von  $m_3$  distanziert werden. Das führt aber wiederum zur Kollision zwischen  $m_2$  und  $m_1$ , weshalb  $m_2$  nach der Kollisionsbehebung mit  $m_3$  noch weiterbewegt werden muss, um nicht mehr mit  $m_1$  zu kollidieren. Die Distanz, die eine Minimap im Falle einer Kollision von ihrer eigentlichen Position überbrücken muss, um nicht mehr mit anderen Minimaps zu kollidieren, steigt mit der Anzahl Minimaps, mit der sie kollidiert.

In Extremfällen kann das dazu führen, dass eine Minimap, die eigentlich in der oberen rechten Ecke platziert werden soll, wegen der Kollisionsbehebung in der unteren rechten Ecke des Viewports erscheint. Ohne eine Strategie, die den ursprünglichen Ort, den eine Minimap repräsentiert, berücksichtigt, kann es dazu kommen, dass Minimaps untereinander platziert werden, obwohl die Orte, die sie repräsentieren, nördlich von den weiter oben platzierten Minimaps gezeichnet werden.

In Abbildung 24 ist das Problem am Beispiel von  $p_{curr} = \text{Rio de Janeiro}$  mit Verbindungen zu den Städten Ougadougou, Moskau, Berlin, Stuttgart und Bari illustriert.

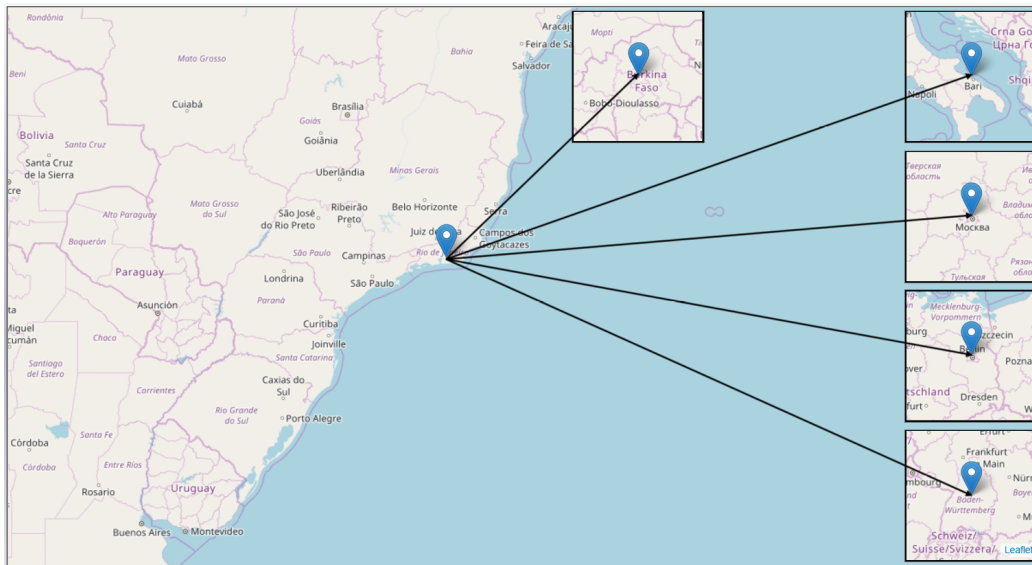


Abbildung 24: Minimaps werden nicht absteigend nach Breitengrad untereinander platziert

Eine Kollision zwischen den Minimaps für Moskau, Berlin, Stuttgart und Bari hat dazu geführt, dass sie untereinander am rechten Rand des Viewports platziert werden. Obwohl Bari der südlichste der vier Orte ist, befindet sich die zugehörige Minimap oberhalb der Minimaps für Moskau, Berlin und Stuttgart.

Um dieses Problem zu beheben, werden die Minimaps, nachdem sie platziert werden, nach den Breiten- und Längengraden der Orte sortiert, die sie repräsentieren. Die sortierten Minimaps werden an den gleichen Positionen an denen sie sich befanden, allerdings in sortierter Reihenfolge neu platziert. Das Ergebnis für das gleiche Beispiel aus Abbildung 24 lässt sich in Abbildung 25 betrachten.

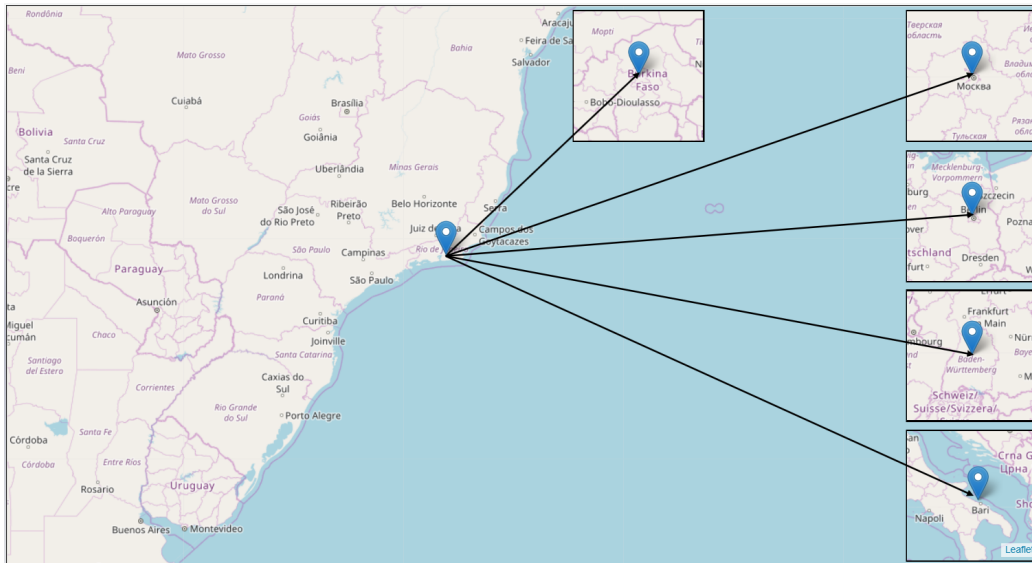


Abbildung 25: Minimaps werden richtigerweise absteigend nach Breitengrad untereinander platziert

Die Minimaps in Abbildung 25 werden korrekterweise von oben nach unten absteigend nach ihren Breitengraden platziert. Die vorherige Reihenfolge von oben nach unten mit Bari, Moskau, Berlin und Stuttgart ist dabei durch die Reihenfolge Moskau, Berlin, Stuttgart und Bari ersetzt worden.

### 3.4 Clustering

Das Problem, dass der begrenzte Platz im Viewport bei der Visualisierung von Graphen im Allgemeinen eine Herausforderung darstellt, lässt sich auch in dieser Arbeit wiederfinden. Auch wenn Verfahren wie beispielsweise die Kollisionsbehebung Überlagerungsprobleme von Minimaps lösen, bringen sie den Nachteil mit, dass mehr Platz des Viewports beansprucht wird (vergleiche Abbildung 24 und 25).

Sobald die Anzahl der Knoten, die sich außerhalb des Viewports befinden groß genug ist, lassen sich Minimaps, die diese Orte repräsentieren, nicht

mehr so platzieren, dass sie sich nicht mehr überlagern. Hat der Viewport  $v$  die Breite  $v_{width}$  und Höhe  $v_{height}$ , so lassen sich Kollisionen nicht vermeiden wenn

$$|Minimaps| > \lfloor \frac{v_{width} \cdot v_{height}}{(m.rad \cdot 2)^2} \rfloor$$

bei einheitlicher Größe  $m.rad$  der Minimaps gilt.

Wie bereits in vielen anderen Arbeiten, wie zum Beispiel von Holten et al. (HVW09) oder Younis et al. (YKR06), wird das Problem, dass zu viele Knoten und Kanten gezeichnet werden müssen und Visual Clutter entsteht, gelöst, indem Elemente geclustert werden. In der Arbeit von Younis et al. (YKR06) werden Wireless-Sensor-Netzwerke, die als Knoten repräsentiert werden, gruppiert, und jede Gruppe von Knoten durch einen neuen Knoten ersetzt, der die gesamte Gruppe repräsentiert. Während sich Younis et al. hauptsächlich mit dem Clustern von Knoten beschäftigen, wird in der Arbeit von Holten et al. die Bündelung von Kanten genauer untersucht. Frisch et al. (FD13) clustern Proxy-Elemente für Knoten, die Elemente in UML-Diagramme repräsentieren. Durch Interaktion mit den Clustern lassen sich die Elemente, die durch das zu untersuchende Cluster gruppiert wurden, genauer untersuchen.

In dieser Arbeit werden, ähnlich wie bei Frisch et al., Proxy-Elemente geclustert, die Offscreen-Objekte repräsentieren. Sowohl die Minimaps, die geclustert werden sollen als auch die geclusterten Minimaps unterscheiden sich dabei von oben erwähnten Clustern dahingehend, dass die Informationen, die sie repräsentieren, keine zusätzliche Interaktion benötigen, um gelesen zu werden. Während Minimaps bereits Details der Offscreen-Objekte enthalten, signalisieren die Proxy-Elemente und Cluster, aus den in diesem Kapitel erwähnten Arbeiten, lediglich die Existenz der Elemente und nicht ihre detaillierten Informationen.

Beim Clustern der Minimaps werden mehrere Orte, die durch Marker repräsentiert werden, nicht mehr durch einzelne Minimaps dargestellt. Stattdessen werden die Minimaps, die geclustert werden sollen, durch eine einzige

Minimap ersetzt, die mehrerer Marker enthält. Sollen beispielsweise die Minimaps  $m_1$ ,  $m_2$  und  $m_3$ , die die Orte  $p_1$ ,  $p_2$  und  $p_3$  repräsentieren, geclustert werden, so wird anstelle der drei Minimaps nur eine einzige Minimap gezeichnet, die für die Orte  $p_1$ ,  $p_2$  und  $p_3$  jeweils einen Marker an den entsprechenden Positionen enthält.

In Abbildung 26 ist für  $p_{curr} = \text{Rom}$  mit neun Verbindungen, wovon sieben durch Minimaps gezeigt werden, ein Beispiel vor dem Clustern der Minimaps für New York City und Philadelphia gezeigt.

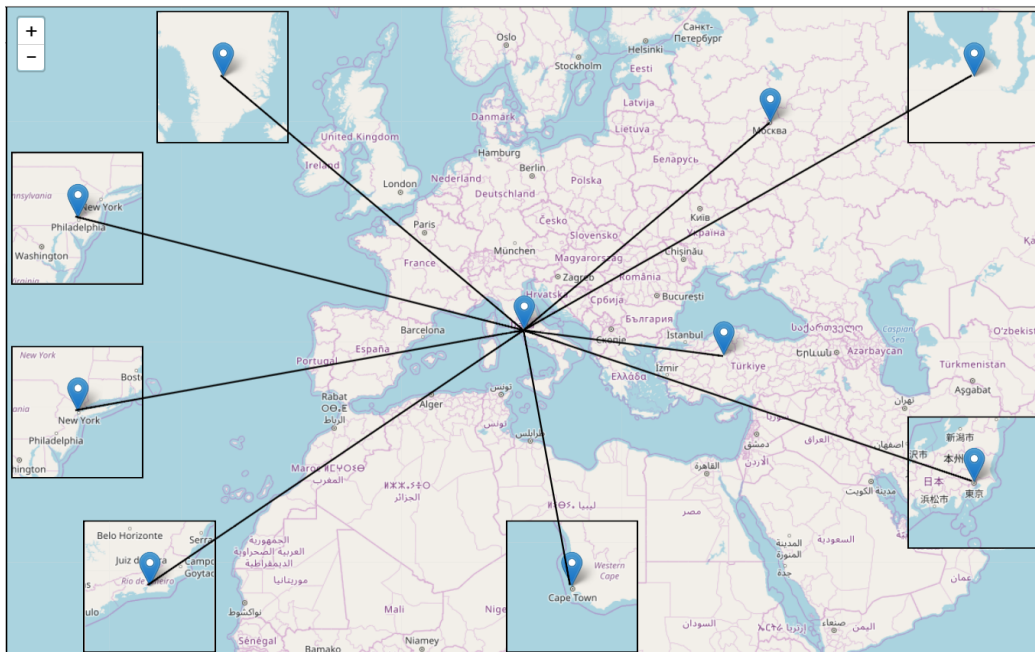


Abbildung 26: Minimaps vor dem Clustern

In Abbildung 27 wurden die Minimaps für New York City und Philadelphia aus Abbildung 26 zu einer Minimap geclustert. Die beiden Minimaps wurden dabei durch eine neue ersetzt, die Marker an den Orten New York City und Philadelphia enthält.

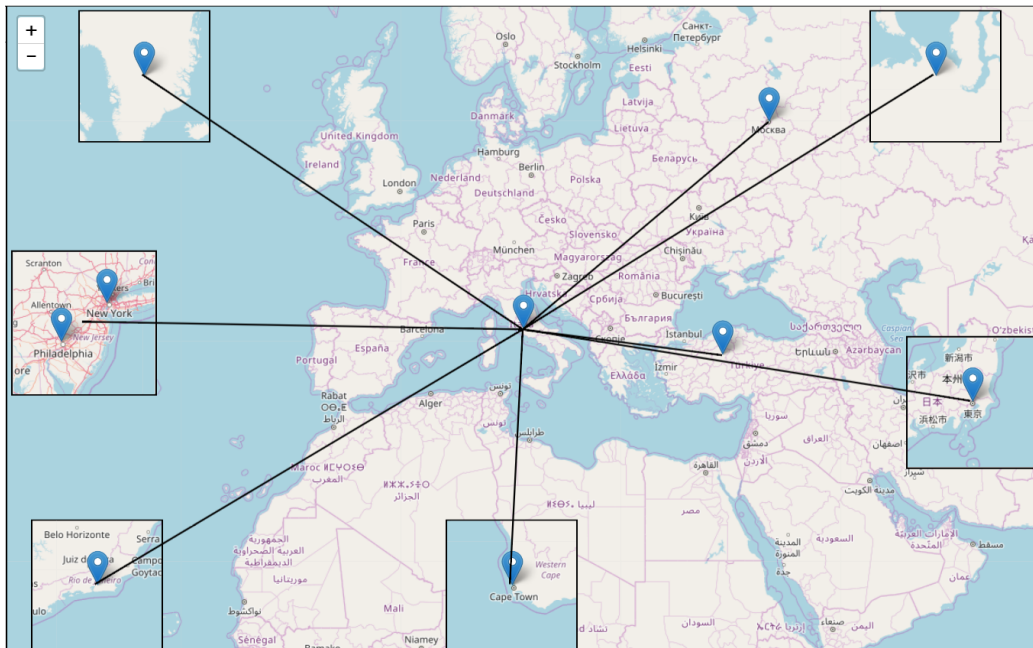


Abbildung 27: Die Minimaps für New York City und Philadelphia aus Abbildung 26 sind zu einer Minimap geclustert worden.

Für die Bestimmung, welche Minimaps geclustert werden sollen, empfehlen sich die beiden folgenden Strategien:

- Clustering nach geo-politischem Kontext
- Clustering nach geographischem Kontext

Beim Clustering nach politischem Kontext werden vorzugsweise Minimaps geclustert, die den gleichen politischen Kontext haben. Zum politischen Kontext gehören beispielsweise Städte-, Länder- oder auch Kontinentgrenzen. Würde nach Ländergrenzen geclustert werden, so würden Minimaps, die Orte aus demselben Land repräsentieren, zu einer Minimap geclustert werden. Dies hat zum einen den Vorteil, dass der Nutzer schnell erkennen kann, mit wie vielen unterschiedlichen Ländern der zu untersuchende Knoten verbunden ist. Allerdings führt die unterschiedlich große Verteilung

der Ländergrößen dazu, dass sehr weit entfernte Orte, wie beispielsweise im Osten und Westen Russlands, zu einer Minimap geclustert werden und für diese Minimap eine sehr geringe Zoomstufe, womit Details schlecht erkennbar wären, gewählt werden müsste. Flächenmäßig kleine Länder, wie beispielsweise San Marino oder Vatikanstadt, die teilweise von einem anderem Land komplett umgeben sind, würden separat geclustert werden, auch wenn sich potentiell weitere Orte, die auf der gleichen Minimap sichtbar wären, aber zu einem anderen Land gehören, in ihrer Nähe befänden. Dieses Problem ließe sich dabei nicht nur bei flächenmäßig kleinen Ländern beobachten, sondern bei allen, sobald sich Orte auf unterschiedlichen Seiten einer Ländergrenze befinden und sehr nah beieinander liegen.

Obwohl diese Strategie in bestimmten Szenarien die bessere der beiden sein könnte, wird in dieser Arbeit nach geographischem Kontext geclustert. Beim Clustering nach geographischem Kontext werden Minimaps geclustert, die Orte repräsentieren, die nah beieinander liegen - unabhängig davon, zu welcher Stadt, welchem Land oder welchem Kontinent sie gehören. Minimaps werden geclustert, wenn Orte unterschiedlicher Minimaps in einer Minimap mit dem bereits gewählten Kartenausschnitt sichtbar sind. Eine Minimap, die eine so geringe Zoomstufe hat, dass die gesamte Weltkarte sichtbar ist, würde dazu führen, dass alle Orte, die durch weitere Minimaps repräsentieren werden, in dieser Minimap geclustert werden.

Das Hauptziel des Clusterings ist nicht nur die Anzahl der Minimaps und somit den benötigten Platz, um Offscreen-Elemente anzeigen zu können, zu reduzieren, sondern auch so viele Details wie möglich, über die zu explorierenden Orte und ihre Verbindungen anzuzeigen.

Das Zoomlevel der Minimaps beeinflusst wie viele Minimaps geclustert werden. Um eine begrenzte Anzahl an Minimaps  $max_{maps}$  zu erreichen, die maximal angezeigt werden sollen, muss das Zoomlevel der Minimaps verringert werden, wenn keine der Minimaps Orte andere Minimaps enthält. Eine Verringerung des Zoomlevels führt dazu, dass weniger Details angezeigt werden, es erhöht aber im Gegenzug die insgesamt sichtbare Fläche der Mi-

nimap.

Werden zwei Minimaps, beziehungsweise zwei Cluster (jede Minimap wird als eigenes Cluster initialisiert), zu einem einzigen Cluster vereint, so wird für alle Orte, die sich im neuen Cluster befinden, das gemeinsame Zentrum gesucht. Anschließend wird der Viewport der neu entstandenen Minimap auf das Zentrum gesetzt und das Zoomlevel so verringert, dass alle Orte im Viewport sichtbar sind.

Befindet sich im sichtbaren Bereich des ausgewählten Kartenausschnitts einer Minimap  $m_1$  der Ort  $p_2$ , der durch eine weitere Minimap  $m_2$  repräsentiert wird, so befindet sich der Ort  $p_1$ , der durch  $m_1$  repräsentiert wird, ebenfalls häufig im Viewport von  $m_2$ . Dies ist insbesondere dann gegeben, wenn Minimaps zu Clustern initialisiert und mit keinen weiteren Cluster vereint wurden. Im Beispiel aus Abbildung 26 und 27 lässt sich das für New York City und Philadelphia beobachten. New York City ist in der Minimap für Philadelphia sichtbar und Philadelphia wiederum in der von New York City.

Zwei Minimaps zu clustern, sobald sie Orte der jeweils anderen Minimap in ihrem Viewport enthalten, kann in manchen Fällen zu einem schlechten Ergebnis führen, das in Abbildung 28 anhand eines Beispiels verdeutlicht wird.



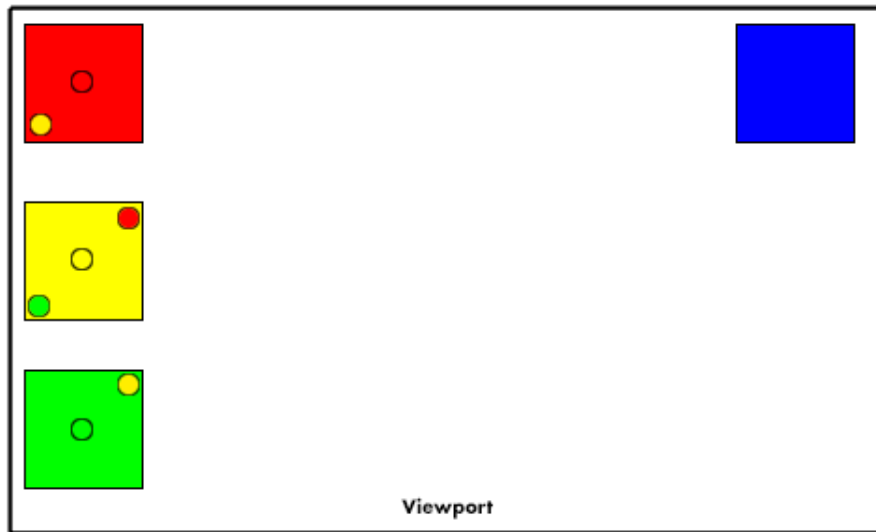


Abbildung 28: Die farbigen Kreise signalisieren die Orte, die durch die gleichfarbigen Minimaps repräsentiert werden.

Man beachte, dass die Quadrate aus Abbildung 28 den Startzustand zu Beginn des Clusterings zeigen und noch nicht geclustert wurden. Im Viewport der gelben Minimap befindet sich sowohl der Ort der roten als auch der grünen Minimap. Die rote und grüne Minimap enthalten zusätzlich zu ihrem eigenen Ort nur den Ort der gelben Minimap. Die gelbe Minimap mit der grünen zu clustern, in dem das grüne Cluster um den gelben Ort erweitert wird, würde zu dem Ergebnis aus Abbildung 29 führen.

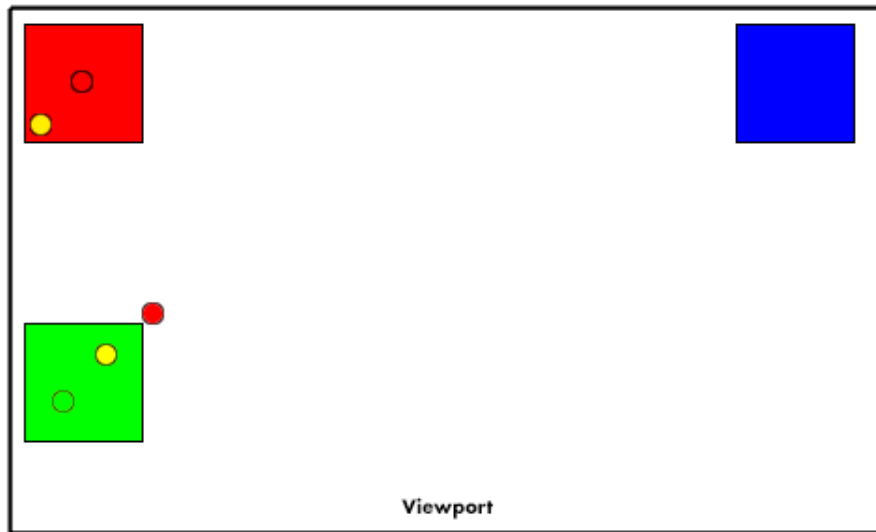


Abbildung 29: Der grüne Cluster wurde mit der gelben Minimap zu einem Cluster vereint, der den gelben und grünen Ort enthält.

Durch die direkte Zusammenführung der gelben und grünen Minimap verschwindet der rote Ort aus dem Viewport des neu entstandenen Clusters, weshalb es zu keinem weiteren Clustering kommt.

Anstelle einer iterativen Vorgehensweise, bei der immer nur zwei Cluster zu einem neuen vereint werden und dessen Zentrum und Zoomlevel neu berechnet wird, empfiehlt es sich, erst alle Cluster zu betrachten, um zu ermitteln, welche Cluster vereint werden sollen, bevor diese n-vielen Cluster direkt zu einem neuen vereint werden.

Hierfür wurde ein Voting-Algorithmus entwickelt, bei dem für jede Minimap gezählt wird, wie viele Orte anderer Minimaps in seinem Viewport enthalten sind. Die Minimap mit den meisten Stimmen wird dabei mit den Orten aller anderen Minimaps, die sich in ihrem Viewport befinden, zusammengeclustert.

Auf das Beispiel aus Abbildung 28 angewandt, folgt folgende Tabelle,

Minimap	Anzahl-Stimmen	Orte von Minimap
Rot	1	Gelb
Gelb	2	Rot, Grün
Grün	1	Gelb
Blau	0	

womit die gelbe Minimap mit den Orten aus der roten und grünen Minimap geclustert wird. Das Ergebnis ist in Abbildung 30 sichtbar.

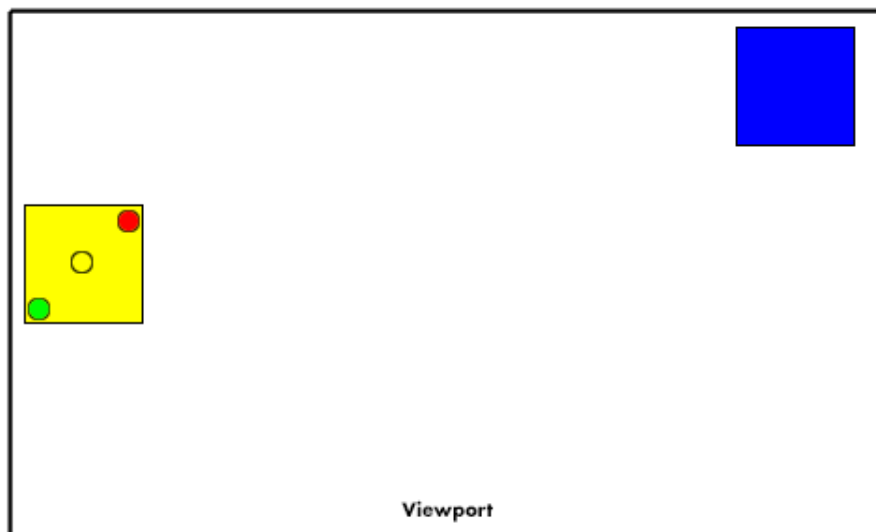


Abbildung 30: Die gelbe Minimap wurde mit der grünen und roten zu einem Cluster vereint.

Für den nächsten Durchlauf des Voting-Algorithmus' folgt folgende Tabelle,

Minimap	Anzahl-Stimmen	Orte von Minimap
Gelb	0	
Blau	0	

womit keine neuen Cluster entstehen. Das Ergebnis aus Abbildung 30

benötigt eine Minimap weniger als bei Abbildung 29 verwendet wird, ohne dass für die Minimaps eine geringere Zoomstufe gewählt werden muss.

Ist die Anzahl der Cluster nach Anwendung des Voting-Algorithmus' größer als  $max_{maps}$ , wird die Zoomstufe der Minimaps so lange verringert, bis sich Minimaps wieder clustern lassen. Sobald die gewünschte Anzahl an Minimaps erreicht ist, wird die Zoomstufe der Minimaps zurückgesetzt, damit Details wieder besser sichtbar werden.

Da geclusterte Minimaps mehrere Marker enthalten, kann es zur Überlappung von Markern innerhalb dieser Minimaps kommen. Aus diesem Grund wird der Radius der Minimap für jeden Ort, der zusätzlich angezeigt wird, etwas vergrößert. Größere Minimaps haben zusätzlich den Vorteil, dass sie dem Nutzer signalisieren, dass sie mehr Informationen als kleinere Minimaps enthalten.

Ein Beispiel für das Clustering und die Wirkung eines größeren Radius' bei Minimaps ist in Abbildung 31 gezeigt.

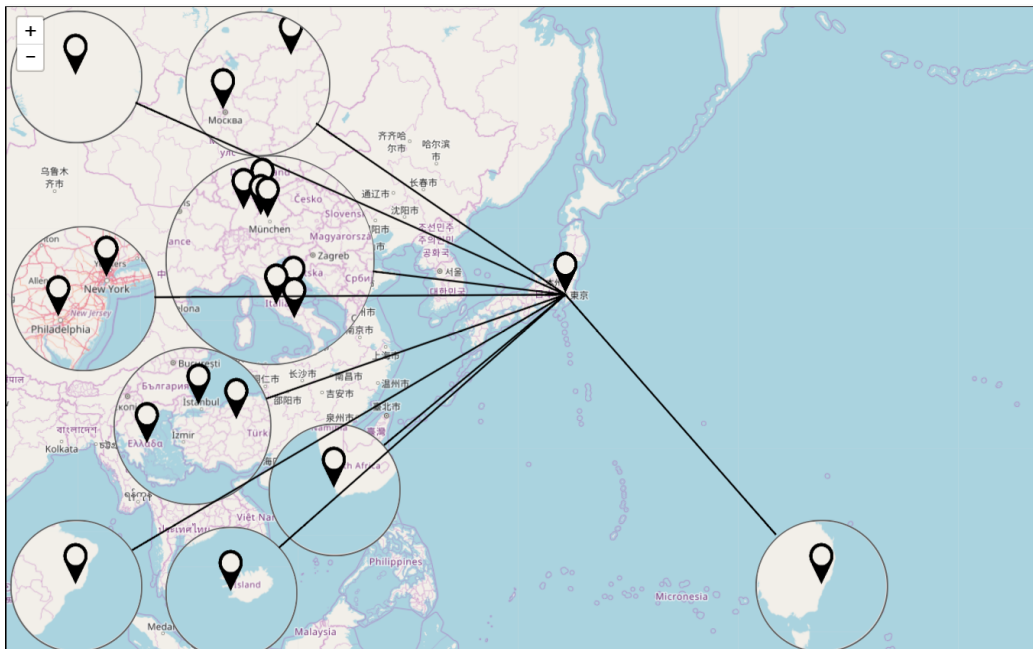


Abbildung 31: Visualisierung von größeren Clustern für  $p_{curr} = \text{Tokio}$

Der vergrößerte Radius der Minimaps, die Orte in Italien und Deutschland geclustert hat, ermöglicht eine höhere Zoomstufe und somit eine detaillierte Ansicht der Umgebung. Für die Positionierung der Minimaps, die auf das Kräftemodell, das in Kapitel 3.3.2 vorgestellt wurde, zurückgreift, wird für die Berechnung der anziehenden Kraft  $F_{a_{orig}}$  zum ursprünglichen Ort, das Zentrum der geclusterten Orte der Minimaps verwendet. Enthält eine Minimaps nur einen Ort, entspricht das Zentrum der Minimaps genau der Position dieses Ortes.

### 3.5 Anpassung der Kanten

Für die Visualisierung der Verbindungen zwischen Knoten gibt es viele verschiedene Techniken, die beispielsweise in der Arbeit von Netzel et al. (NBW14) erwähnt werden. Ein simpler Ansatz, bei dem Knoten über Linien verbunden sind, die je nachdem, ob es sich um ein- oder ausgehende Kanten handelt, Pfeile zu Beginn oder Ende der Verbindung (siehe beispielsweise Abbildung 10) haben. Diese werden auch für Verbindungen zu Knoten, die innerhalb von Minimaps angezeigt werden, erstellt.

Wegen der in diesem Kapitel eingeführten Gruppierung von Orten in einer Minimaps, lässt sich diese simple Herangehensweise nicht verfolgen, ohne dass direkte Verbindungen zu Orten in einer Minimaps zu Visual Clutter führen. In Abbildung 32 lässt sich beobachten, wie Verbindungen, ohne dass Pfeilspitzen eingezeichnet werden, bereits bei kleinen Clustern zu Visual Clutter führen können.

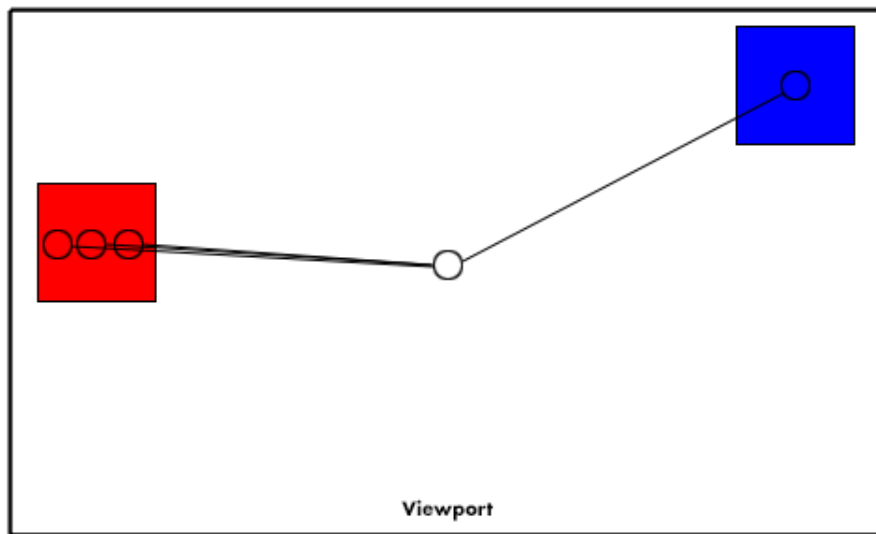


Abbildung 32: Verbindungen zu allen Orten einer Minimap führen zu Visual Clutter.

Anstatt Verbindungen zwischen dem zentralen Knoten und allen Knoten einer Minimap einzuzichnen, werden die Verbindungen lediglich bis zu Beginn der Minimap gezeichnet (siehe Abbildung 31 beispielsweise für die Verbindung zwischen Tokio und der Minimap für Sydney im unteren rechten Viertel des Bildes). Die Form der Minimap wird jetzt kreisförmig und nicht mehr quadratisch gewählt, was folgende Vorteile mit sich bringt:

- Minimaps nehmen weniger Platz ein
- Die wichtigsten Informationen befinden sich im Zentrum der Minimap
- Assoziation mit Lupe vonseiten der Nutzer führt zu schnellerem Verständnis der Minimap
- Ermöglicht schnellere Berechnung des Schnittpunktes zwischen Kreis und Linie

Für Minimaps kommen auch noch weitere Formen in Frage. In der Arbeit von Bordkorb et al. (BKA<sup>+</sup>16) nehmen Minimaps die Form von Rechtecken ein. Dies hat den Vorteil, dass Knoten, die weiter entlang der Breiten- oder Längengrade auseinander liegen, besser mit Hilfe von rechteckigen Minimaps visualisiert werden können, ohne das Zoomlevel zu verringern.

Während sich die Richtung der Verbindungen im Falle von Minimaps, die nur einen Knoten enthalten, erneut mit Hilfe von Pfeilspitzen am Anfang oder Ende der Linie visualisieren lassen, ist dies für Minimaps, die mehrere Orte enthalten, die sowohl über ein- als auch ausgehende Kanten mit  $p_{curr}$  verbunden sind, nicht anwendbar.

Um eindeutig anzuzeigen, welcher Knoten über eine ein- beziehungsweise ausgehende Kante verbunden ist, wird die Pfeilrichtung innerhalb des Markers platziert.

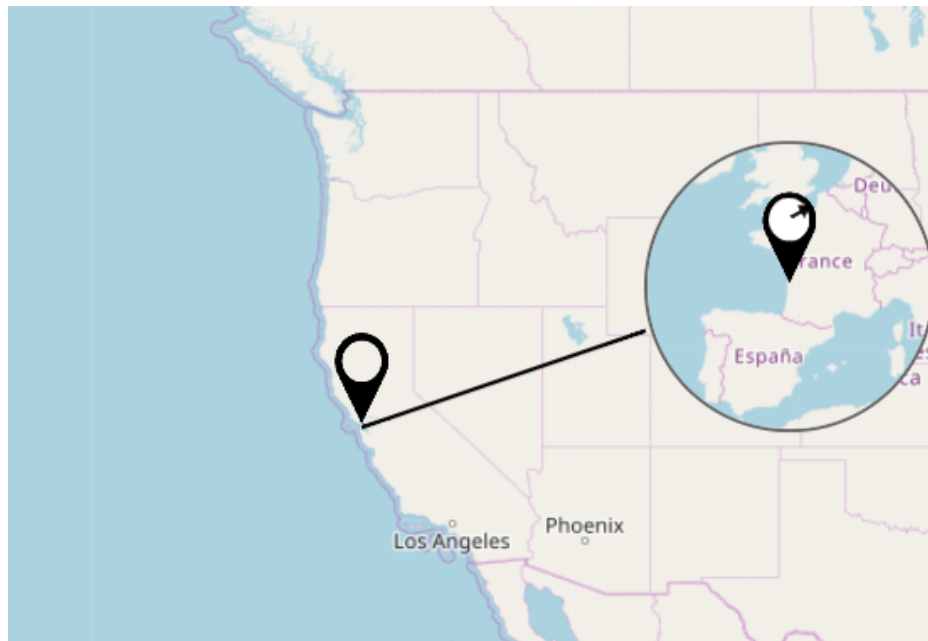


Abbildung 33: Ausgehende Kante: Pfeilrichtung im Marker

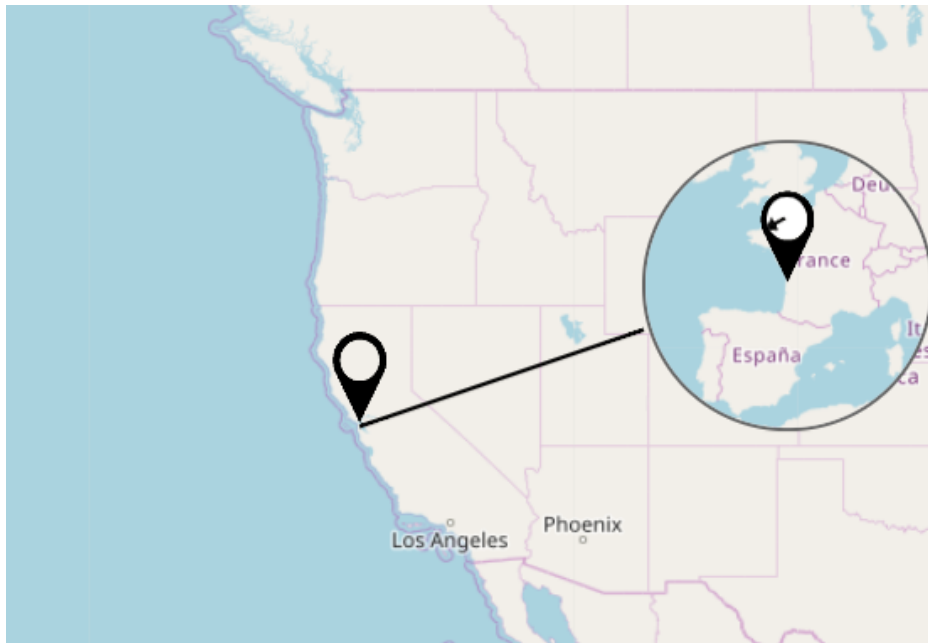


Abbildung 34: Eingehende Kante: Pfeilrichtung im Marker

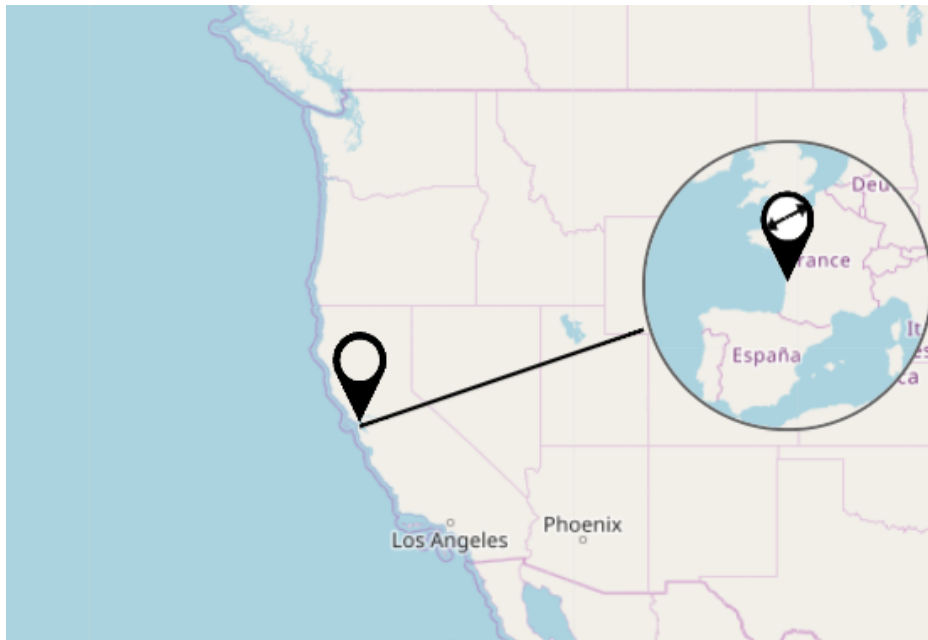


Abbildung 35: Ein- und ausgehende Kante: Pfeilrichtung im Marker



Die Information, ob es sich um ein- oder ausgehende Kanten handelt, im Marker selbst zu kodieren, hat den Vorteil, dass der User den Blick nicht vom Zielort rüber zu  $p_{curr}$  richten muss, sondern die Information direkt am Zielort ablesen kann. Viele eingehende Kanten in  $p_{curr}$  erschweren außerdem das Erkennen, ob es sich um ein- oder ausgehende Kanten handelt, vor allem, wenn sowohl ein- als auch ausgehende Kanten in die selbe Richtung zeigen (siehe Abbildung 32).

### 3.6 Interaktion

Auch wenn es eines der Ziele dieser Arbeit ist, die benötigte Nutzerinteraktion fürs Explorieren des Graphen so stark wie möglich zu reduzieren, werden in diesem Kapitel sowohl diverse Interaktionsmöglichkeiten als auch zusätzliche visuelle Features vorgestellt, die dem Nutzer helfen, sollen die Exploration des Graphen weiter zu vereinfachen.

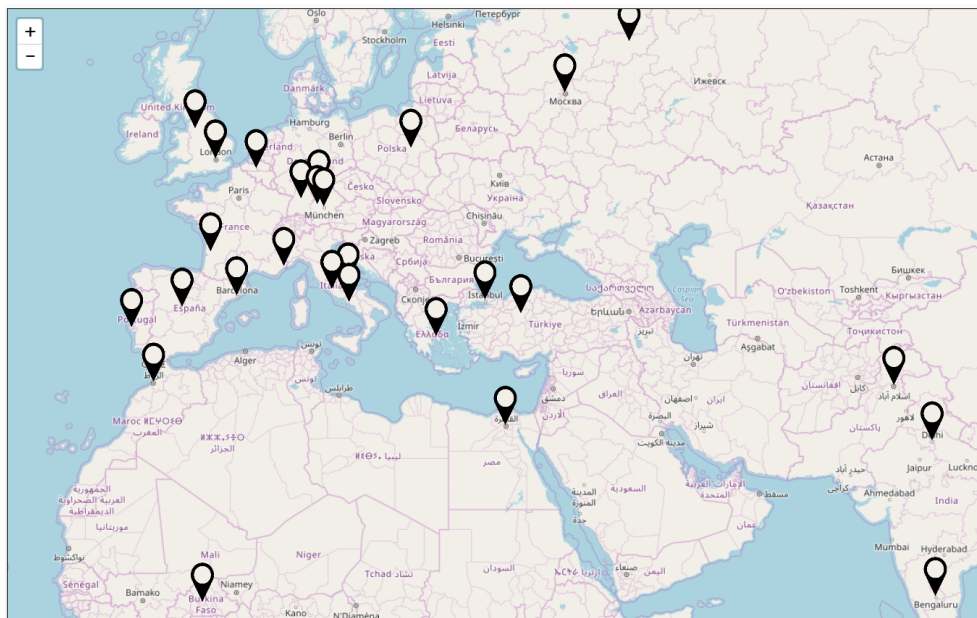


Abbildung 36: Alle Knoten des Graphen werden beim Öffnen des Tools angezeigt.

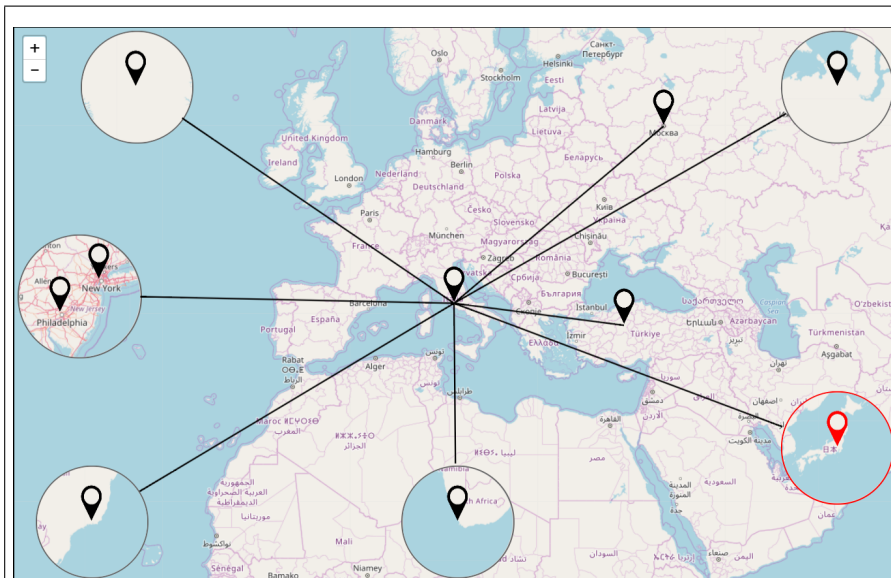
Öffnet der Nutzer das Tool, das sich in modernen Webbrowsern starten lässt, werden alle Marker, ohne ein- und ausgehende Verbindungen, auf der Karte angezeigt (siehe Abbildung 36).

Mit Hilfe von Zooming und Panning lässt sich ein Überblick über die Position aller existierenden Marker und ihrer Verteilung auf der Karte gewinnen. Wird über einen Marker gehovert, erscheint unter ihm ein Label, das den Städtenamen anzeigt. Der Nutzer kann mit der genaueren Exploration des Graphen beginnen indem er auf einen gewünschten Knoten klickt, wodurch dieser zu  $p_{curr}$  gesetzt und der sichtbare Kartenausschnitt so gewählt wird, dass der angeklickte Knoten im Zentrum des Viewports platziert ist.

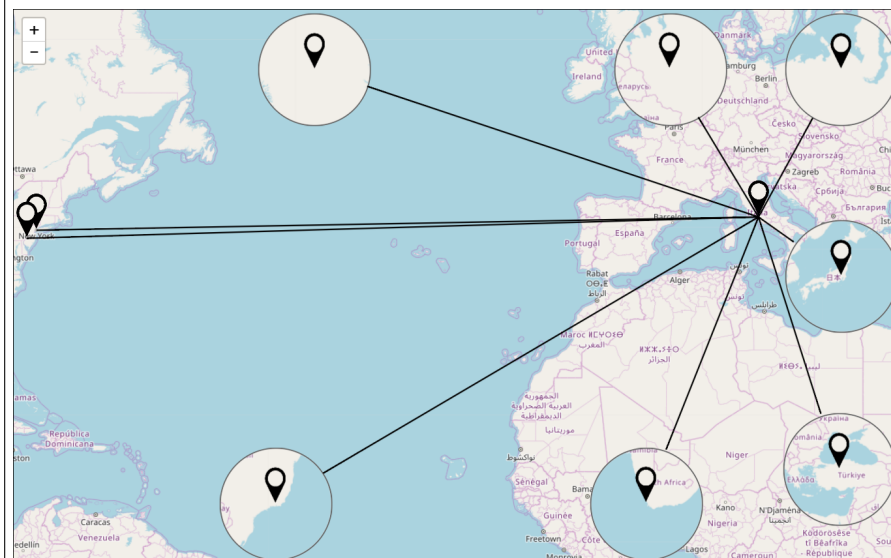
Werden Verbindungen eines Knotens angezeigt, wodurch in vielen Fällen Minimaps im Viewport platziert werden, lässt sich der sichtbare Kartenausschnitt und sein Zoomlevel mit Hilfe von Zoom- und Panoperationen anpassen. Wenn beispielsweise viele Minimaps entlang der linken Kante des Viewports angezeigt werden, ergibt es durchaus Sinn, dass der Nutzer nach Westen navigiert, um einen besseren Überblick über die Verteilung der Knoten auf der westlichen Hälfte der Karte zu erhalten. Minimaps werden überflüssig sobald Orte, die sie repräsentieren, auf der Hauptkarte erscheinen (durch Nutzerinteraktion). Orte, die wiederum vorher auf der Hauptkarte sichtbar waren, können durch Zoom- und Panoperationen aus dem Viewport verschwinden, wodurch für diese Minimaps benötigt werden.

Aus diesen Gründen werden Minimaps und ihre Positionierungen nach Zoom- und Panoperationen neu berechnet. Für die Berechnung der Kräfte aus dem zugrunde liegenden Kräftemodell wird anstelle der Position von  $p_{curr}$  das Zentrum des Viewports verwendet. Die Position von  $p_{curr}$  entspricht zu Beginn immer dem Zentrum des Viewports.

Die Neuberechnung der Minimaps nach Zoom- und Panoperationen kann, wie beispielsweise in Abbildung 37 gezeigt ist, dazu führen, dass bestehende Minimaps verschwinden und durch Marker an den entsprechenden Orten auf der Hauptkarte und entsprechend andersrum ersetzt werden.



(a) Darstellung nach Auswahl von Rom



(b) Nachdem der Nutzer eine Pan-Interaktion in Richtung Westen durchgeführt hat

Abbildung 37: Vor und nach Paninteraktion nach links

Die Minimap, die die Orte New York City und Philadelphia in Abbildung

37 a) repräsentiert hat, ist in Abbildung 37 b) nicht mehr sichtbar, weil sich beide Orte innerhalb des Viewports befinden. Im Gegenzug sind zwei Minimaps, eine für Ankara und eine für Moskau, hinzugekommen, weil sowohl Moskau als auch Ankara außerhalb des Viewports bewegt wurden.

Der Nutzer kann das Zoomlevel einer Minimap mit Hilfe des Mausrads verändern. Nach der Zoomoperation wird das Zentrum der Minimap wieder automatisch auf das ursprüngliche Zentrum der Minimap (wie vor der Zoomoperation) gelegt, weil davon ausgegangen wird, dass das Interesse des Nutzers auf den Markern liegt, die sich im Zentrum (beziehungsweise um das Zentrum herum) befinden.

Es gibt zwei Möglichkeiten, um einen Knoten, der innerhalb einer Minimap dargestellt wird, zu selektieren. Der Nutzer kann zum einen auf den Marker und zum anderen in die Minimap, in der er sich befindet, klicken. Letzteres führt zu einer Animation bei der sich die Minimap erst in die Mitte des Viewports verschiebt und anschließend so lange vergrößert wird, bis sie den gesamten Viewport einnimmt. Ihre Form verändert sich dabei von kreisförmig zu rechteckig.

Befindet sich in der angeklickten Minimap nur ein Knoten, wird dieser am Ende der Animation ausgewählt und alle Verbindungen zu anderen Knoten, die er hat, angezeigt. Wenn sich mehrerer Knoten in der angeklickten Minimap befinden, wird der Knoten gesucht, der den kleinsten Abstand zur Stelle hat, an die der Nutzer innerhalb der Minimap geklickt hat. Für diesen Knoten werden am Ende der Animation alle Relationen angezeigt.

Diese zusätzliche Möglichkeit, Knoten auszuwählen, um sie genauer zu untersuchen, bietet folgende Vorteile:

- Der Nutzer muss den Marker nicht genau treffen, um den entsprechenden Knoten auszuwählen.
- Die Animation hilft die Operation nachzuvollziehen, wodurch die Mental Map des Nutzers nicht gelöscht wird.

- Die Animation wirkt natürlich und erinnert an ein Heraus- bzw Hineinzoomen, was bei kreisförmigen Minimaps die Assoziation zu Lupen erhöht.

Ein weiteres Hilfsmittel, das bei der Evaluation, die im nächsten Kapitel beschrieben ist, untersucht wurde, ist der rotfarbene Marker, der den vorherig ausgewählten Ort anzeigt. Befindet sich ein roter Marker innerhalb einer Minimap, wird sie ebenfalls durch eine rote anstelle einer dunkelgrauen Umrandung angezeigt. Wählt ein User beispielsweise als allerersten Knoten Stuttgart aus, wird kein roter Marker angezeigt; das liegt daran, dass vor Stuttgart kein Knoten ausgewählt wurde. Sobald ein nächster Knoten ausgewählt wird, wie im Falle von Abbildung 38 New York City, wird der Marker für Stuttgart durch einen roten ersetzt. Sobald der nächste Knoten ausgewählt wird, wird New York City wiederum durch einen roten Marker repräsentiert.

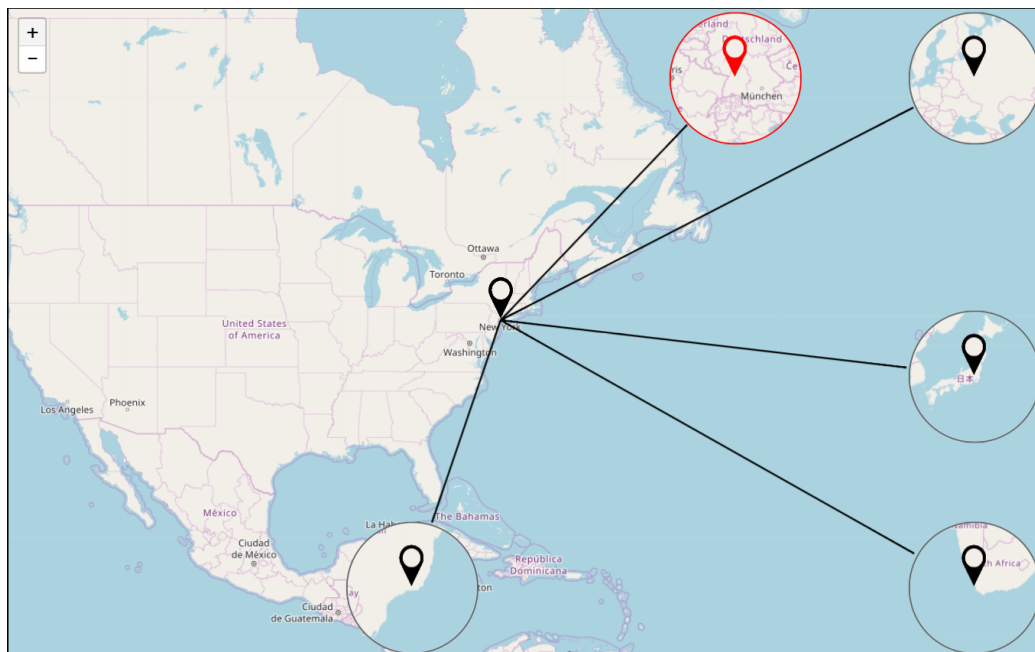


Abbildung 38: Roter Marker bei Stuttgart zeigt den vorher ausgewählten Knoten an.

Dieser soll die kognitive Belastung des Nutzers reduzieren, da es, vor allem wenn viele Knoten angezeigt werden, für den Nutzer schwierig sein kann, den Überblick zu behalten und sich daran zu erinnern, über welche Interaktionssequenz man zum aktuellen Knoten gelangt ist.

Je mehr Verbindungen ein Knoten zu anderen hat, desto schwieriger ist es, den Überblick zu behalten, was die kognitive Belastung weiter erhöht. Eine Sidebar, die links neben dem Viewports platziert ist, soll diesem Problem entgegenwirken, indem der ausgewählte Städtenamen und alle Relationen des ausgewählten Knotens darin aufgelistet werden. In Abbildung 39 ist die Karte mit der Side für das vorher genannte Beispiel, bei dem zuerst Stuttgart und anschließend New York City ausgewählt wurde, dargestellt.

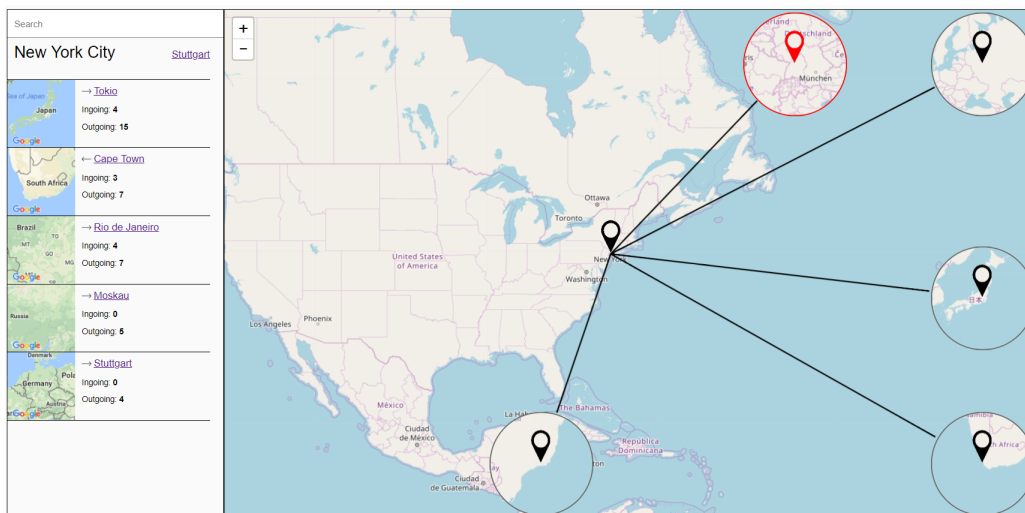


Abbildung 39: Sidebar links zeigt zusätzliche Informationen zum aktuell sichtbaren Teilgraphen an.

Die Sidebar enthält die folgende Elemente:

### 1. Suchfeld:

Ermöglicht Wildcard-Suche nach einem Ort mit dem eingegebenen Städtenamen.

**2. Ausgewählter Ort:**

Der Städtenamenname des gerade ausgewählten Knotens ( $p_{curr}$ ).

**3. Herkunftsort:**

Der Städtenamenname des Ortes, der zuvor ausgewählt wurde (roter Marker).

**4. Liste der Knoten:**

Eine Auflistung aller Knoten, die in Relation zu  $p_{curr}$  stehen.

**5. Land des Knotens:**

Ein Kartenausschnitt des Landes der Stadt des zugehörigen Knotens. Das Bild wird mit Hilfe der Google API für statische Karten

`https://maps.googleapis.com/maps/api/staticmap?parameters`

geladen. Als Parameter für den Service, der 2500 Mal kostenlos pro Tag aufgerufen werden kann, wird der Ländername übergeben, der wiederum mit Hilfe der Google API für Maps Geocoding

`https://maps.googleapis.com/maps/api/geocode/outputFormat?parameters`

ermittelt wird, indem die Längen- und Breitengrade des Knotens als Parameter übergeben werden.

**6. Kantenrichtung:**

Die Kantenrichtung gibt wieder, ob es sich bei dem Knoten um eine eingehende, ausgehende oder bidirektionale Relation handelt. Ein Pfeil nach rechts signalisiert eine ausgehende Verbindung von  $p_{curr}$  zum entsprechenden Knoten in der Liste. Ein Pfeil nach links steht für eine eingehende und ein doppelspitziger Pfeil für eine bidirektionale Verbindung.

**7. Städtenamenname:**

Der Städtenamenname des Knotens in der Liste. Der entsprechende Knoten lässt sich selektieren, indem auf den Städtenamenname geklickt wird.

## 8. Ein- und ausgehende Verbindungen:

Die Anzahl von ein- und ausgehenden Verbindungen des Knotens in der Liste.

Die numerisch aufgezählten Elemente der Sidebar sind in Abbildung 40 markiert:

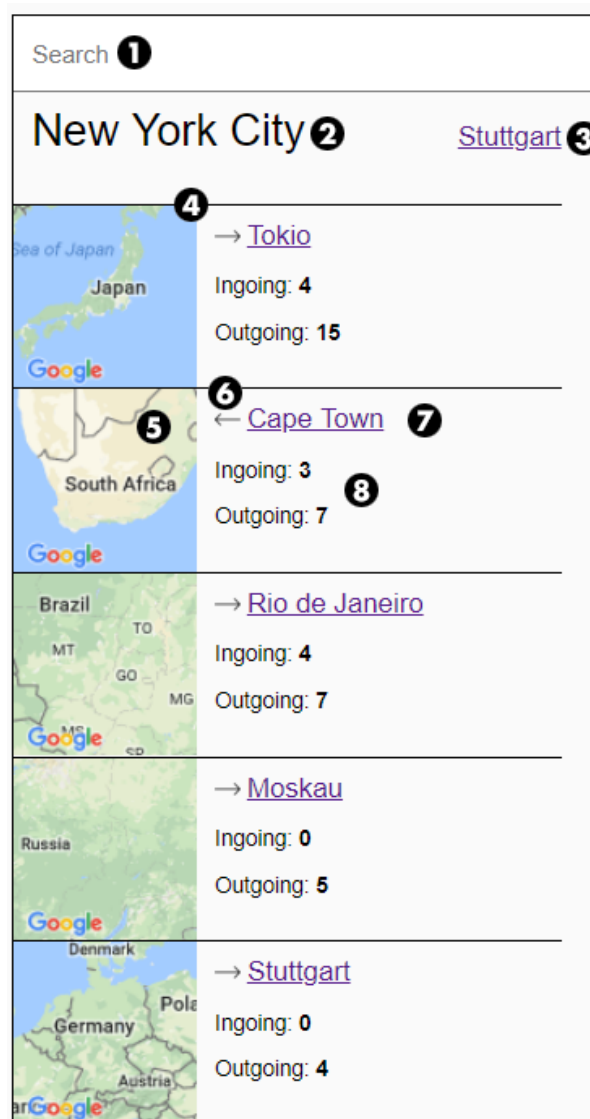


Abbildung 40: Elemente der Sidebar



## 4 Evaluation

Für die Evaluation des im Rahmen dieser Arbeit konzipierten und implementierten Ansatzes für die Exploration von relationalen Daten mit geographischem Bezug, wurde eine Nutzerstudie durchgeführt, die in diesem Kapitel beschrieben wird.

Ziel der Nutzerstudie war es zum Einen die Hypothese, dass der in dieser Arbeit implementierte Ansatz besser geeignet ist als simple Techniken (insbesondere Übersicht + Details Ansätze), um komplexe Graphen mit regionalen und transkontinentalen Verbindungen zu explorieren, zu untersuchen. Der Fokus liegt vor allem auf der schrittweisen Exploration, bei der erst ein Knoten ausgewählt wird und anschließend dessen Verbindungen genauer untersucht werden.

Zum anderen geht es bei der Nutzerstudie darum, Feedback zum Tool zu erhalten, wobei sowohl Verbesserungsvorschläge als auch Aspekte, die vom Nutzer als gut empfunden werden, dokumentiert werden.

### 4.1 Setup

Die Nutzerstudie wurde mit insgesamt 23 Personen (20 männliche und 3 weibliche) durchgeführt. Bei 11 der Personen - Nutzergruppe A - handelt es sich um Mitarbeiter der Universität Stuttgart, die im Fachbereich für Informatik und Softwaretechnik in einer lehrenden oder forschenden Funktion tätig sind. Bei 8 der Personen handelt es sich sogar um Experten der Informationsvisualisierung, die am Visualisierungsinstitut der Universität Stuttgart angestellt sind.

Die zweite Gruppe - Nutzergruppe B - besteht aus 12 Probanden, die zwischen 23 und 52 Jahre alt sind und das Arbeiten mit dem Computer sowohl aus dem Alltag als auch aus dem Berufsleben kennen. Sie arbeiten in unterschiedlichen Bereichen (Journalismus, Medizin etc.) und haben, mit

Ausnahme einer Personen, einen Hochschulabschluss oder eine vergleichbare Qualifikation.

Um einen möglichst guten Vergleich zwischen der Variante, bei der Minimaps benutzt werden, um weit entfernte Orte, die sich außerhalb des Viewports befinden, anzuzeigen, wurde ein zweites Tool geschrieben, das keine Minimaps verwendet, ansonsten aber die gleichen Funktionalitäten und Interaktionsmöglichkeiten anbietet. Verbindungen zu Knoten außerhalb des Viewports lassen sich dabei nur mit Hilfe von Interaktion wie Zooming und Panning ermitteln. Die erwähnte Sidebar wurde für die meisten Aufgaben deaktiviert, weil die Exploration des Graphen auf der Karte im Vordergrund der Studie steht.

Für die Durchführung der User-Tests - ein User-Test hat im Schnitt ca. 45 Minuten gedauert - wurde jedem Probanden ein Fragebogen vorgelegt, den er beantworten sollte. Der Fragebogen besteht aus 22 Fragen und Aufgaben, die mit Hilfe der oben erwähnten Tools (einmal mit und einmal ohne Minimaps), die künstlich generierte Netzwerke anzeigen, beantwortet werden sollen. Die Tools werden jedem Nutzer zu Beginn der Studie kurz vorgestellt.

Der erste Teil des Fragebogens besteht aus Aufgaben, die im Tool, ohne Verwendung von Minimaps, gelöst werden sollen. Für jede Aufgabe wird sowohl die Zeit als auch die Fehlerrate gemessen. Die gemessenen Daten sind in Kapitel 4.2 notiert. Bei den Aufgaben, die so konzipiert sind, dass sie in der Schwierigkeit steigen, geht es darum, Relationen eines Knotens zu identifizieren.

Nach jeder Aufgabe wird der User nach ihrer Schwierigkeit auf einer ordinalen Skala

- Sehr Leicht
- Leicht
- Mittel

- Schwer
- Sehr Schwer

gefragt und ihm wird die Möglichkeit geboten, Kommentare dazu abzugeben.

Fragen zur Bewertung gewisser Funktionalitäten kann der User mit Werten der ordinalen Skala

- Sehr Gut
- Gut
- Mittel
- Schlecht
- Sehr Schlecht

beantworten.

Ein Ausschnitt des Fragebogens ist in Abbildung 41 gezeigt.

**3) *Explorieren Sie bitte den Graphen und untersuchen dabei die Verbindungen, die Barcelona zu anderen Städten hat***

---

**4) *Beurteilen Sie, wie schwer es Ihnen gefallen ist.***

---

Sehr Leicht	Leicht	Mittel	Schwer	Sehr Schwer
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kommentar:

Abbildung 41: Auszug aus dem Fragebogen für die Nutzerstudie

Für die erste Aufgabe (**Aufgabe 3** im Fragebogen), bei der alle direkten Verbindungen eines Knotens identifiziert werden sollen, sind alle Knoten, die mit  $p_{curr} = \text{Barcelona}$  verbunden sind, im Viewport sichtbar (siehe Abbildung 42).

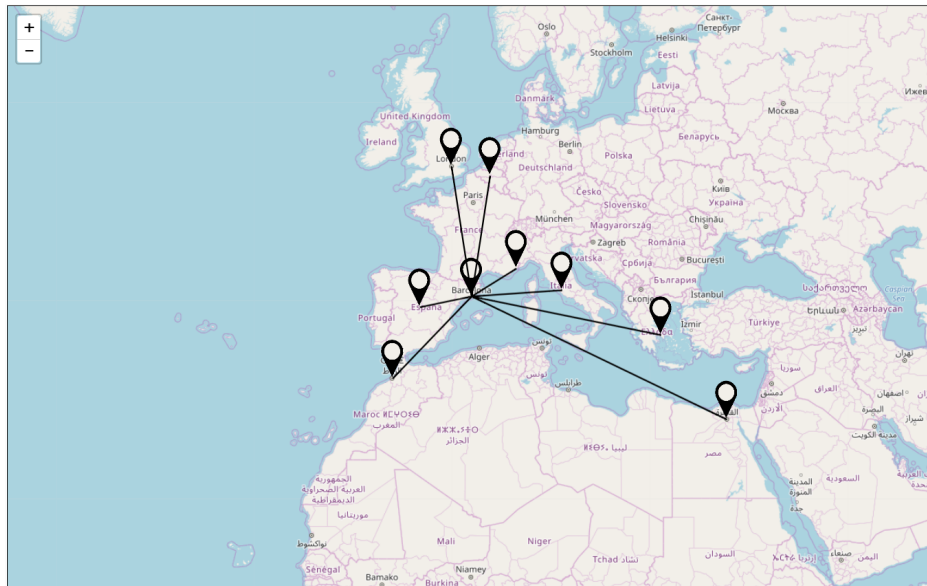
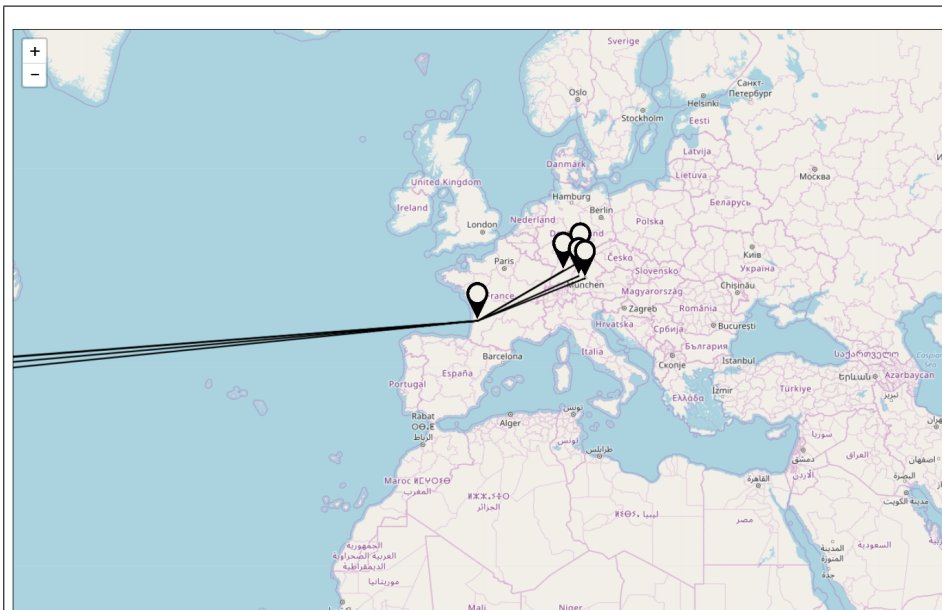


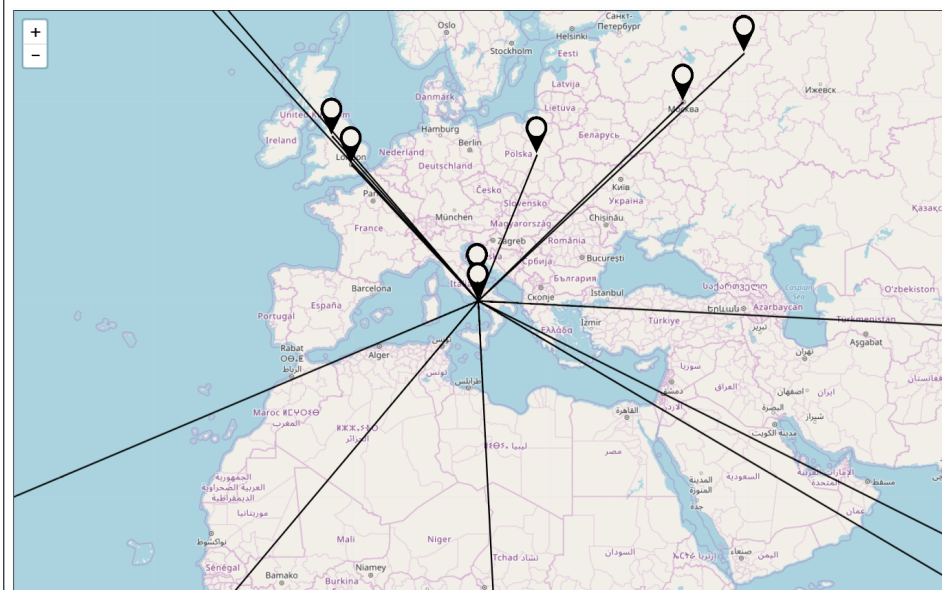
Abbildung 42: Simpler Fall, bei dem alle Knoten im Viewport enthalten sind.

Weder aufwändige Zoom- noch Panoperationen sind notwendig, um die Verbindungen zu Barcelona zu identifizieren. Erkennt der Nutzer eine Stadt nicht, kann er den zugehörigen Städtenamen einblenden lassen, in dem er die Maus über den entsprechenden Marker bewegt.

Für die nächsten beiden Aufgaben des Fragebogens (**Aufgabe 5** und **Aufgabe 7**) wird immer mehr Interaktion benötigt, um direkte Verbindungen zu identifizieren. Abbildung 43 a) und b) enthält Abbildungen der Karten-ausschnitte direkt nach Auswahl der Knoten der entsprechenden Aufgabe.



(a) Abbildung nach Auswahl des Knotens aus Aufgabe 5



(b) Abbildung nach Auswahl des Knotens aus Aufgabe 7

Abbildung 43: Abbildungen zu Aufgaben 5 und 7 des Fragebogens

Fürs Explorieren der Verbindungen des zu untersuchenden Knotens aus

Aufgabe 5 muss der Nutzer entweder rauszoomen oder nach links navigieren, um die Knoten zu sehen, die an der Ost- und Westküste Amerikas platziert sind.

Für Aufgabe 7 ist im Gegensatz zu Aufgabe 5 wiederholte Interaktion vonseiten des Nutzers notwendig, um alle Knoten zu identifizieren. Die Knoten liegen in allen Himmelsrichtungen verstreut und befinden sich außerhalb des Viewports. Wiederholtes Zooming oder Panning in die unterschiedlichen Richtungen, in die die Knoten liegen, erschweren die Aufgabe.

Während der erste Teil des Fragebogens darauf abzielt, vermutete Schwierigkeiten beim Explorieren von Graphen mit regionalen und transkontinentalen Relationen genauer zu untersuchen und gegebenenfalls zu bestätigen, geht es im zweiten Teil des Fragebogens darum herauszufinden, ob und wie gut der Ansatz mit Minimaps funktioniert. Für den zweiten Teil wird deshalb das Tool mit Minimaps verwendet.

Bei der ersten Aufgabe des zweiten Teils des Fragebogens (**Aufgabe 9**) geht es darum, schrittweise alle direkten Verbindungen eines Knotens, und für jeden dieser Knoten wiederum erneut alle seine Relationen, zu identifizieren; dies entspricht einer Tiefensuche mit Tiefe 2. Nachdem der Nutzer den zu untersuchenden Knoten, in diesem Fall ist  $p_{curr} = \text{Rabat}$ , ausgewählt hat, wählt er einen Knoten, mit dem  $p_{curr}$  in Relation steht aus, untersucht dessen Verbindungen und kehrt daraufhin zum ursprünglichen Knoten (Rabat) zurück, um den nächsten Knoten auf die gleiche Art und Weise zu untersuchen.

Da die Knoten sowohl regional als auch transkontinental verteilt sind, soll der Nutzer den Umgang mit Minimaps testen und zugleich das Zusammenspiel von Knoten, die in Minimaps dargestellt werden, und Knoten, die sich im Viewport befinden und deshalb nicht durch Minimaps repräsentiert werden, erfahren. Der Nutzer arbeitet dabei mit geclusterten Minimaps, kann beobachten wie Minimaps verschwinden, wenn ihre Orte auf der Hauptkarte sichtbar werden und neue Minimaps für Orte hinzukommen, die den View-

port durch Interaktion verlassen und nutzt dabei den roten Marker, um den zuletzt besuchten Ort leichter zu erkennen. Die restlichen Fragen des zweiten Teils des Fragebogens und ihre Ergebnisse werden in Kapitel 4.2 genauer erläutert. Dabei werden auch Fragen zur Form und Positionierung der Minimaps gestellt.

Beim dritten und letzten Teil des Fragebogens geht es um einen direkten Vergleich zwischen der Variante mit und ohne Minimaps. Bei **Aufgabe 19** soll der Nutzer direkte Relationen für  $p_{curr} = \text{Vorkuta}$ , bei der Variante ohne Minimaps, und direkte Relationen für  $p_{curr} = \text{Tokio}$ , bei der Variante mit Minimaps, ermitteln. Beide Knoten haben 22 ein- und ausgehende Verbindungen, die sowohl regional als auch transkontinental verteilt liegen.

## 4.2 Ergebnisse

Im Folgenden werden die wichtigsten Ergebnisse der Nutzerstudie aufgelistet, die nach Nutzergruppe A und Nutzergruppe B unterteilt sind.

### 4.2.1 Komplexität bei Graphen mit regionalen und transkontinentalen Relationen

Der erste Teil des Fragebogens überprüft, ob und wie stark Knoten, die sowohl regionale als auch transnationale Relationen haben, die Exploration des Graphen erschweren.

Für die **Aufgaben 3, 5 und 7**, die bereits im vorherigen Kapitel vorgestellt wurden, sind die Ergebnisse der nachfolgenden Tabelle und Abbildung 44 zu entnehmen.

Die Vermutung, dass die Komplexität der Aufgabe steigt, wenn sowohl regionale als auch transkontinentale Relationen existieren, wird durch den ersten Teil der Nutzerstudie unterstützt. Sowohl bei den Experten aus Nutzergruppe A als auch bei den Probanden aus Nutzergruppe B ist sowohl eine

Erhöhung der Dauer als auch der Fehlerquote beim Lösen der Aufgaben bemerkbar.

Table 1: Zeitmessungen für das Lösen der Aufgabe 3, 5 und 7

Aufgabe	Nutzergruppe	Ø Zeit in s	Anzahl Fehler
3	A	28.8	0
3	B	34.6	0
5	A	32.5	0
5	B	40.3	0
7	A	50.9	1
7	B	58.1	2

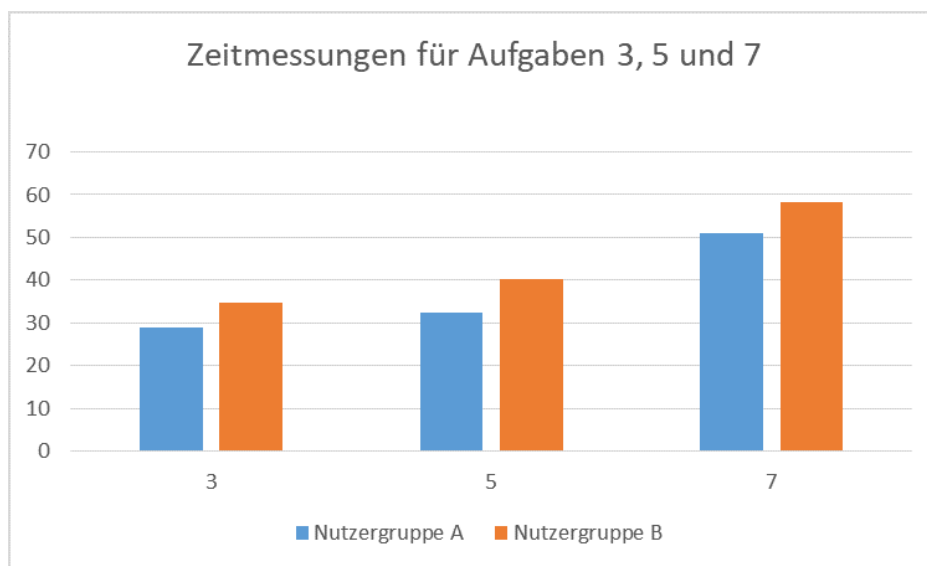


Abbildung 44: Zeitmessungen für Aufgaben 3, 5 und 7

Auffällig ist, dass bei transkontinentalen Relationen in nur eine Richtung, wie es bei Aufgabe 5 der Fall ist, nur leichte Erhöhungen in der Lösungsdauer der Aufgabe gemessen wurden. Bei transkontinentalen Relationen in unterschiedliche Richtungen steigt nicht nur die benötigte Zeit, sondern auch die



Fehlerquote. Während bei Aufgabe 3 und 5 kein Proband der beiden Nutzergruppen einen Fehler gemacht hat, haben bei Aufgabe 7, bei der transkontinentale Relationen in unterschiedliche Richtungen untersucht werden mussten, drei Probanden (einer aus Nutzergruppe A und zwei aus Nutzergruppe B) mindestens einen Knoten bei ihrer Aufzählung übersehen.

Die steigende Schwierigkeit, die sich beim Lösen der Aufgaben beobachten lässt, wird von den Nutzern wahrgenommen, wie in der folgenden Tabelle und in Abbildung 45 für Nutzergruppe A und Abbildung 46 für Nutzergruppe B, die die Bewertung der Schwierigkeit nach Empfinden der Nutzer zeigen, wiedergegeben ist.

Table 2: Schwierigkeitseinstufungen für Aufgabe 3, 5 und 7

<b>Aufgabe</b>	<b>Nutzergr.</b>	<b>Sehr Leicht</b>	<b>Leicht</b>	<b>Mittel</b>	<b>Schwer</b>	<b>Sehr Schwer</b>
3	A	10	1	0	0	0
3	B	8	4	0	0	0
5	A	6	5	0	0	0
5	B	6	3	3	0	0
7	A	0	9	2	0	0
7	B	0	5	6	1	0

Während Aufgabe 3 noch von 18 der insgesamt 23 Teilnehmer der Studie als *Sehr Leicht* empfunden wird (bei Aufgabe 5 sind es noch 12), empfindet Aufgabe 7 kein einziger Teilnehmer mehr als *Sehr Leicht*. Begründet wurde die erhöhte Schwierigkeit vor allem mit der vermehrten Interaktion (Zooming, Panning und Hovering über Knoten) vonseiten der Nutzer.

Zu erwähnen ist außerdem, dass die Expertengruppe, wie bereits zu erwarten war, insgesamt weniger Zeit zum Lösen der Aufgaben benötigt und weniger Fehler gemacht hat.

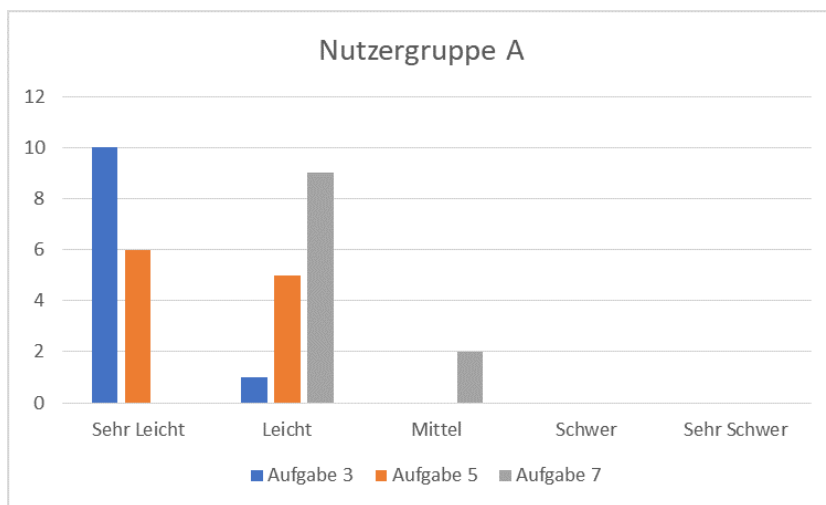


Abbildung 45: Schwierigkeitseinstufungen von Nutzergruppe A für Aufgabe 3, 5 und 7

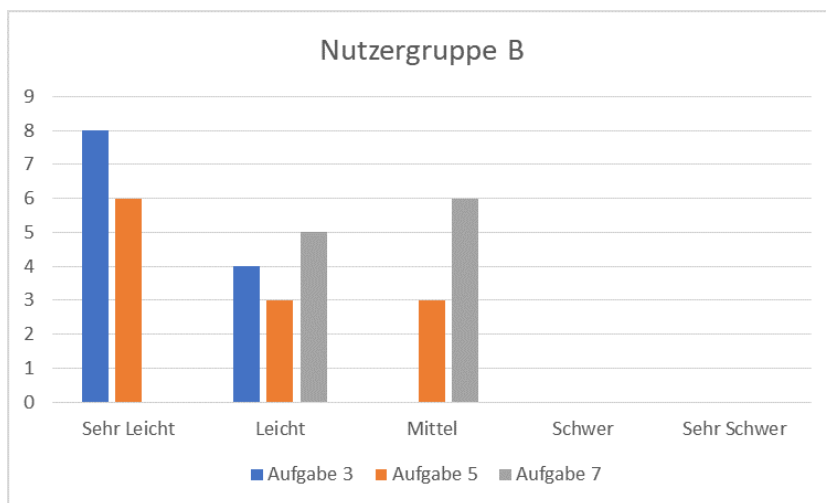


Abbildung 46: Schwierigkeitseinstufungen von Nutzergruppe B für Aufgabe 3, 5 und 7

#### 4.2.2 Nutzer-Feedback zum Tool

Beim zweiten Teil der Nutzerstudie, der mit dem Tool, das Minimaps verwendet, durchgeführt wurde, sollten die Probanden erst anhand einer etwas aufwändigeren Aufgabe, bei der ein Teil des zu untersuchenden Graphen schrittweise exploriert werden soll, mit dem implementierten Konzept vertraut werden, bevor detaillierte Fragen zum Konzept und seiner Umsetzung gestellt werden.

Bei **Aufgabe 9** des Fragebogens, die bereits im vorherigen Kapitel erläutert wurde, geht es darum für den Ort Rabat alle direkten Verbindungen und wiederum deren direkten Verbindungen zu identifizieren.

Die Ergebnisse sind in den folgenden Tabellen notiert:

Table 3: Zeitmessungen fürs Lösen der Aufgabe 9

Aufgabe	Nutzergr.	Ø Zeit in s	Anzahl Fehler
9	A	118.1	1
9	B	140.8	1

Table 4: Schwierigkeitseinstufungen für Aufgabe 9

Aufgabe	Nutzergr.	Sehr Leicht	Leicht	Mittel	Schwer	Sehr Schwer
9	A	2	7	2	0	0
9	B	1	4	6	1	0

Nutzergruppe A hat im Durchschnitt zum Lösen der Aufgabe 118.1 Sekunden benötigt und war damit deutlich schneller als Nutzergruppe B, die im Schnitt 140.8 Sekunden für die gleiche Aufgabe benötigt hat. In beiden Nutzergruppen hat jeweils nur eine Person einen Fehler gemacht - beide Personen haben einen Ort nicht genannt - was vermutlich auf die mit der Dauer des Tests nachlassende Konzentration zurückzuführen ist. Die erhöhte Dauer im Vergleich zu den vorherigen Aufgaben lässt sich auf die erhöhte Anzahl

Knoten, die zu untersuchen war, zurückführen. Ein besserer Vergleich wird im dritten Teil der Ergebnisse der Nutzerstudie vorgestellt.

Die Bewertungen bezüglich der Schwierigkeit der Aufgabe variiert von *Sehr Leicht* bis *Schwer*. Viele der Probanden, die vor allem Bewertungen im Bereich mittel bis schwer abgegeben haben, haben dies mit der erhöhten Dauer, die für das Lösen der Aufgabe benötigt war, begründet.

Den implementierten Ansatz, der Minimaps zum Anzeigen weit entfernter Orte verwendet, wurde in

**Frage 11:** *Wie bewerten Sie den Einsatz von Minimaps, um entfernte Orte anzuzeigen?* (mit den Antwortmöglichkeiten: *Sehr Schlecht*, *Schlecht*, *Mittel*, *Gut* und *Sehr Gut*)

erfragt und wurde durchwegs positiv beantwortet.

Aus Nutzergruppe B haben alle Probanden den Einsatz von Minimaps als *Sehr Gut* bewertet. Die Expertengruppe hat, mit Ausnahme von drei Personen, die mit *Gut* gestimmt haben, den Einsatz von Minimaps ebenfalls als *Sehr Gut* bewertet.

Positiv vermerkt wurde vor allem, dass weniger Interaktion benötigt wird, um den Graphen zu explorieren. Der Einsatz der Minimaps wurde außerdem von vielen als natürlich und selbsterklärend empfunden. Verbesserungspotential haben die drei Probanden aus der Expertengruppe, die mit *Gut* gestimmt haben, bei der Positionierung der Minimaps gesehen, die in der nächsten Frage genauer untersucht wurde.

**Frage 12:** *Wie bewerten Sie die Platzierung der Minimaps?* (erneut von *Sehr Schlecht* bis *Sehr Gut*).

Die Ergebnisse stehen in der folgenden Tabelle:

Table 5: Bewertung für die Platzierung der Minimaps

Frage	Nutzergr.	Sehr Gut	Gut	Mittel	Schlecht	Sehr Schlecht
12	A	8	2	1	0	0
12	B	10	2	0	0	0

Insgesamt vergaben 18 der 23 Teilnehmer die Bestnote *Sehr Gut* für die Positionierung der Minimaps. Jeweils zwei Personen aus Nutzergruppe A und Nutzergruppe B bewerteten sie mit *Gut* und eine Person stimmte mit *Mittel*. Während die Personen, die mit *Sehr Gut* gestimmt haben, fanden, dass die Minimaps genau dort platziert sind, wo sie es erwartet haben, bemerkten die restlichen vier Probanden, dass in manchen Fällen Minimaps nicht genau entlang der Richtung platziert sind, in die die Orte, die die Minimaps repräsentieren, liegen.

Dieses Verhalten wurde vor allem dann beobachtet, wenn sich eine Minimap in der Nähe eines Markers befindet, der nicht durch eine Minimap repräsentiert wird. Die Abstoßungskraft  $F_{r_{node}}$ , die Minimaps von anderen Knoten im aktuellen Viewport abstößt, ist für diese Verschiebung, die von 4 Probanden als Fehlplatzierung empfunden wurde, verantwortlich. Eine Veränderung der Faktoren, die zur Regulierung der Kräfte verwendet wird, insbesondere von  $k_{map}$ , welcher mit  $F_{r_{node}}$  verrechnet wird, könnte dieses Problem beheben.

Dieses Problem wurde von den gleichen Probanden erneut bei der Frage zum Verschwinden und Erscheinen von Minimaps bei Interaktion erwähnt.

**Frage 16:** *Wie bewerten Sie, dass Minimaps verschwinden, wenn ihre Orte auf der Hauptkarte sichtbar sind?*

Table 6: Ergebnisse zu Frage 16

Frage	Nutzergr.	Sehr Gut	Gut	Mittel	Schlecht	Sehr Schlecht
12	A	9	2	0	0	0
12	B	11	1	0	0	0

Obwohl das Problem, dass Minimaps in manchen Fällen nicht optimal platziert werden, bei dieser Frage erneut erwähnt wurde, fällt die Bewertung etwas besser aus, als bei der Frage zur Positionierung der Minimaps.

Zwei der Probanden, die bei Frage 12 noch mit *Mittel* und *Gut* gestimmt haben, verbesserten ihre Bewertung von *Mittel* auf *Gut* und von *Gut* zu *Sehr Gut*, weil sie, genauso wie die restlichen Personen, die mit *Sehr Gut* gestimmt haben, sowohl das Erscheinen als auch das Verschwinden der Minimaps als natürlich und richtig empfanden. Die Positionierung der Minimaps wurde bereits in Frage 12 beantwortet und deshalb von einigen der Probanden bei der Beantwortung dieser Frage nicht miteinbezogen.

Zur Form und der Darstellung der Minimaps im Viewport wurden folgende zwei Fragen mit den darunter stehenden Bildern gestellt.

Frage 13: Welche Form würden Sie bei Minimaps bevorzugen?

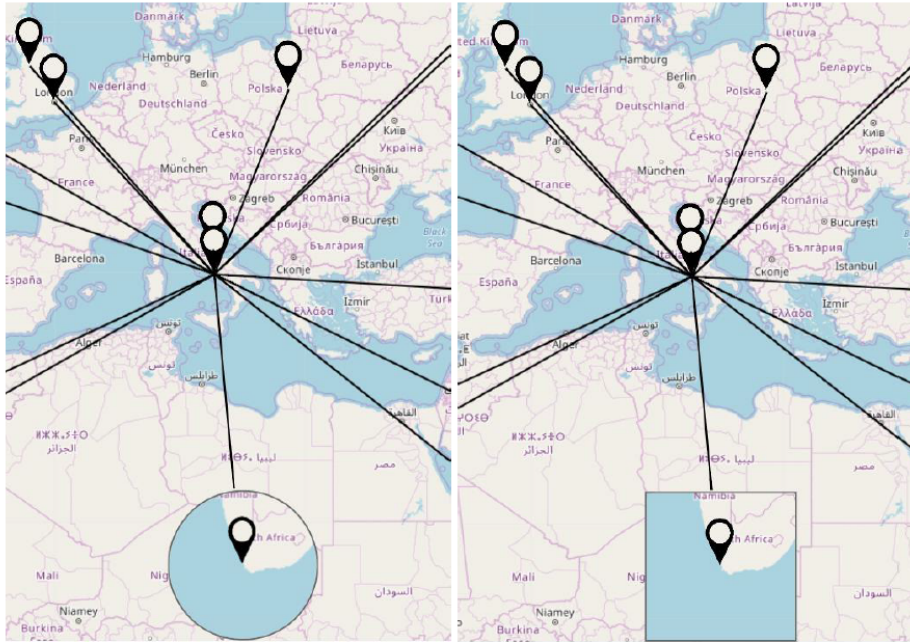


Abbildung 47: Links: Runde Form. Rechts: Quadratische Form

Frage 20: Welche der beiden Varianten bevorzugen Sie? (Mit oder ohne Schatten)

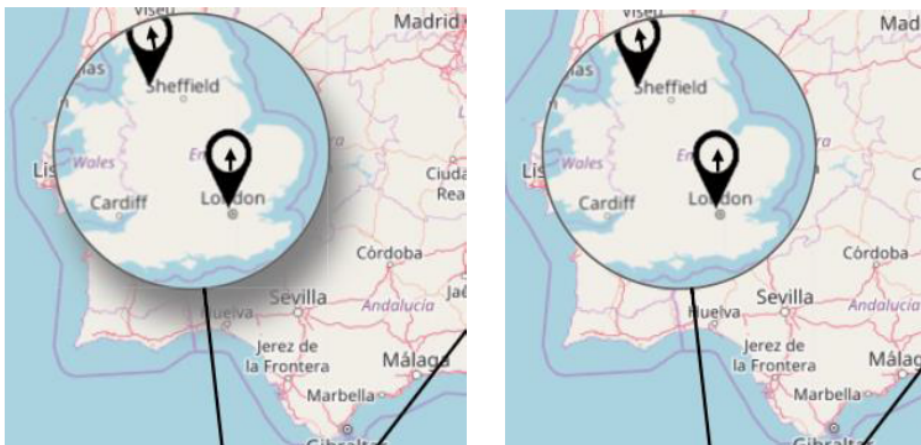


Abbildung 48: Links: Mit Schatten. Rechts: Ohne Schatte

Insgesamt bevorzugten 19 der 23 Probanden kreisförmige Minimaps gegenüber quadratischen. Aus Nutzergruppe A stimmte lediglich eine Person für die quadratische Form, konnte ihre Entscheidung allerdings nicht begründen. Viele der Teilnehmer, die für die kreisförmige Minimap gestimmt haben, erwähnten, dass sie an eine Lupe erinnert und weniger Platz als die quadratische beansprucht. Details interessieren vor allem im Zentrum der Minimap, weshalb Informationen an den Ecken der quadratischen Minimap als weniger wichtig und somit unnötig empfunden wurden.

Bis auf eine Person aus Nutzergruppe A bevorzugten alle Personen (aus Nutzergruppe A und B) die Darstellung mit Schatten bei Minimaps. Positiv wurde erwähnt, dass der Schatten hilft, Minimaps von der Hauptkarte abzugrenzen und somit leichter erkannt werden können. Als Nachteil wurde der zusätzliche Platz, den der Schatten einnimmt erwähnt, der von der Person, die für die Variante ohne Schatten gestimmt hat, ausschlaggebend für ihre Entscheidung war.

Zu den Markern wurden bei der Nutzerstudie zwei Fragen gestellt, wobei die eine auf die Verwendung von roten Markern, um den zuletzt besuchten Ort anzuzeigen, bezogen war und die andere auf die Pfeile, die innerhalb der Marker gezeichnet werden, abzielt.

**Frage 14:** *Wie bewerten Sie den Einsatz von roten Markern, um den zuletzt besuchten Ort anzuzeigen?*

Die Frage zum Einsatz von roten Markern ist die einzige, die von allen Teilnehmern als *Sehr Gut* bewertet wurde. Positiv vermerkt wurde vor allem, dass der Nutzer sich nicht merken muss, welchen Ort er zuletzt besucht hat. Stattdessen kann er seine Konzentration der Exploration des aktuellen Knoten widmen. Drei Probanden aus der Expertengruppe haben weiter den Wunsch geäußert alle Knoten farblich zu markieren, die schon mal besucht wurden und mehr Farben oder Graustufen einzusetzen, um die Reihenfolge besuchter Knoten zu erfassen.

**Frage 18:** *Welche der beiden Varianten bevorzugen Sie, um Pfeilrichtun-*



gen anzuzeigen?

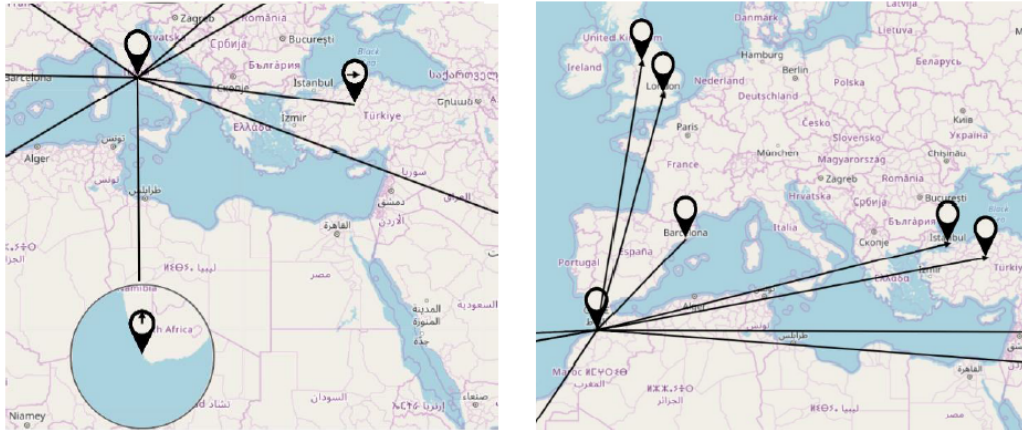


Abbildung 49: Links: Pfeil innerhalb des Markers. Rechts: Pfeil zu Beginn oder Ende der Verbindungslinie zum Knoten.

Wie in der folgenden Tabelle zu sehen ist, kam es bei dieser Frage zu keinem klaren Ergebnis.

Frage	Nutzergr.	Pfeil im Marker	Pfeil bei Verbindung
18	A	6	5
18	B	5	7

Der erste Impuls hat viele Probanden für die rechte Variante, die die Pfeile bei den Verbindungen anzeigt, stimmen lassen. Insgesamt haben 12 Personen für diese Variante gestimmt, wovon 5 aus Nutzergruppe A und 7 aus Nutzergruppe B stammen. Für die linke Variante, die die Pfeilrichtung innerhalb der Marker zeigt, haben die 11 verbliebenen Personen gestimmt.

Ausschlaggebender Grund für die Entscheidung für die rechte Variante war, dass sie diese Form der Darstellung der Pfeilrichtung gewohnt sind und sie deshalb als natürlich empfinden. Teilnehmer, die länger über die Frage nachgedacht haben, haben in den meisten Fällen für die linke Variante gestimmt und sie damit begründet, dass sie nicht wüssten, wie die rechte Variante für Marker innerhalb von Minimaps funktionieren würde. Außerdem

erwähnten sie, dass bei der rechten Variante viele in  $p_{curr}$  eingehende Kanten zu Visual Clutter führen, was das Erkennen der Pfeilrichtung erschwert. Kanten direkt mit Markern, die innerhalb einer Minimaps liegen, zu verbinden, würden viele der Personen, die für die linke Variante gestimmt haben, als störend empfinden, weil die Verbindungen wichtige Informationen, die in der Minimaps gezeigt sind, überdecken könnten. Bei geclusterten Minimaps, die mehrere Marker enthalten, würde dieses Problem besonders stark auffallen.

Ein Proband aus Nutzergruppe B empfand die Verbindungen zu Markern oder Minimaps als komplett überflüssig und schlug vor, die schwarzen Verbindungslinien wegzulassen. Stattdessen sollen nur die Pfeile innerhalb der Marker verwendet werden.

Ein weiterer Proband, diesmal aus Nutzergruppe A, schlug eine Mischform vor, bei der für Marker, die sich nicht in Minimaps befinden, die rechte Variante und für Marker, die sich in Minimaps befinden, die linke Variante verwendet werden soll. Dieser hatte deshalb Schwierigkeiten für eine der beiden Varianten zu stimmen, entschloss sich dann aber dennoch für die rechte, weil er fand, dass Nutzer eine gewisse Zeit bräuchten, um sich an die linke Variante zu gewöhnen.

Ein eindeutiges Ergebnis konnte bei den letzten beiden Fragen des zweiten Teils der Nutzerstudie erzielt werden.

**Frage 16:** *Wie bewerten Sie, dass mehrere Orte in einer Minimaps angezeigt werden?*

Bei dieser Frage stimmten alle 12 Personen aus Nutzergruppe B und 9 Personen aus Nutzergruppe A mit *Sehr Gut*. Die übrigen zwei Personen aus Nutzergruppe A stimmten mit *Gut*. Alle Probanden erwähnten, dass sie es als sinnvoll und natürlich empfinden, mehrere Orte, die nah beieinander sind, in einer Minimaps anzuzeigen. Die Wahl, welche Orte in einer Map geclustert werden, empfanden alle Personen, bis auf die beiden Probanden, die mit *Gut* gestimmt haben, als richtig und hatten nichts auszusetzen. Die beiden Personen, die mit *Gut* gestimmt hatten, begründeten ihre Entscheidung damit, dass

Minimaps, die mehrere Orte enthalten beim Pannen in Richtung der Minimaps nicht aufgelöst wurden, bis alle Orte der Minimaps auf der Hauptkarte sichtbar waren. Sie hätten erwartet, dass sobald ein Ort einer geclusterten Minimaps auf der Hauptkarte erscheint, dieser Ort nicht mehr in der Minimaps repräsentiert wird, sondern stattdessen auf der Hauptkarte gezeichnet wird.

**Frage 17:** *Wie bewerten Sie die Detailansicht links?*

Für diese Frage wurde die Sidebar links neben dem Viewport eingeblendet und ihre Funktionen erklärt - siehe Kapitel 3.6. Sowohl ihr Design als auch ihre Funktionalitäten und die Informationen, die sie enthält, wurde von den meisten Personen positiv bewertet. Alle Probanden aus Nutzergruppe B bewerteten die Sidebar als *Sehr Gut*. Neun Personen aus Nutzergruppe A schlossen sich dieser Meinung an. Vier Personen bewerteten die Sidebar mit *Gut* und begründeten ihre Entscheidung damit, dass sie einige der Informationen, wie zum Beispiel die Anzahl ein- und ausgehender Kanten, direkt in der Hauptkarte, beispielsweise als Glyph, bevorzugen würden. Die Nützlichkeit der Sidebar wurde wiederum von allen Personen als hoch eingestuft, wobei die Sidebar je nach Aufgabe, die es zu lösen gilt, mal mehr und mal weniger hilft.

### 4.2.3 Direkter Vergleich der beiden Ansätze

Beim dritten und letzten Teil der Nutzerstudie sollten die Probanden zwei ähnlich komplexe Graphen untersuchen - einmal mit der Variante mit Minimaps und einmal ohne. Ziel war dabei ein möglichst guter Vergleich der beiden Varianten, um die Hypothese, dass die Variante mit Minimaps besser geeignet ist als gängige Verfahren, um Graphen zu explorieren, zu bestätigen oder zu widerlegen.

In **Aufgabe 19**, die bereits im vorherigen Kapitel erwähnt wurde, soll der Nutzer beim Tool ohne Minimaps alle direkten Verbindungen von Vorkuta und beim Tool mit Minimaps alle direkten Verbindungen von Tokio aus ermitteln. Beide Szenarien bestehen aus 22 Knoten, die sowohl regional aus

auch transkontinental verteilt sind.

Die Ergebnisse, erneut nach Nutzergruppe A und B unterteilt, sind in folgender Tabelle beschrieben.

<b>Aufgabe</b>	<b>Nutzergr.</b>	<b>Ø Zeit in s</b>	<b>Fehler</b>
19 (mit Minimaps)	A	35.5	1
19 (ohne Minimaps)	A	57.3	3
19 (mit Minimaps)	B	41.2	0
19 (ohne Minimaps)	B	65.2	3

Wie bereits in vorherigen Aufgaben haben die Probanden der Expertengruppe für beide Varianten etwas weniger Zeit als die Probanden aus Nutzergruppe B benötigt. Beide Nutzergruppen waren bei der Variante, die Minimaps verwendet, etwas schneller als bei der Variante ohne Minimaps. Die Expertengruppe hat für das Lösen der Aufgabe unter Verwendung von Minimaps im Schnitt 35.5 Sekunden und ohne Verwendung von Minimaps 57.3 Sekunden benötigt. Im Vergleich dazu haben die Probanden aus Nutzergruppe B 41.2 Sekunden mit und 65.2 Sekunden ohne Minimaps benötigt.

Besonders auffällig waren die gemachten Fehler bei dieser Aufgabe. Während beim Untersuchen der Verbindungen, die Tokio hat (mit Minimaps), die Expertengruppe einen Fehler gemacht, kam es bei Nutzergruppe B zu keinem einzigen Fehler. Der wahrscheinlichste Grund für den begangenen Fehler ist vermutlich die nachlassende Konzentration des Probanden, bei dem die Nutzerstudie etwas mehr als eine Stunde gedauert hat.

Für die ähnliche Aufgabe ohne Verwendung von Minimaps haben in beiden Nutzergruppen jeweils drei Personen einen Fehler gemacht und eine Stadt ausgelassen.

Die Testergebnisse zeigen deutlich, dass die Variante mit Minimaps der ohne in bestimmten Situationen überlegen ist. Um zu verstehen, wie der in dieser Arbeit konzipierte und implementierte Ansatz im Vergleich mit bereits veröffentlichten Ansätzen bei unterschiedlichen Aufgabenstellungen

abschneidet, müssen weitere Studien durchgeführt werden, bei denen dieser Ansatz mit weiteren Ansätzen - wie im Kapitel für verwandte Arbeiten aufgelistet - verglichen wird. Außerdem sollten unterschiedliche Aufgaben auch auf Realwelt-Daten anstelle von künstlich generierten Daten getestet werden, obwohl die Ergebnisse dieser Studie ein vergleichbares Ergebnis vermuten lassen.

## 5 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Konzept erarbeitet, das Nutzern die Exploration von Daten mit geographischem Bezug erleichtern soll. Um es zu testen wurde ein Softwareprototyp mit Hilfe der Open-Source JavaScript-Bibliotheken Leaflet (lea) und d3js (d3j) implementiert, der für die Nutzerstudie verwendet wurde. Dazu wurden im ersten Schritt bestehende Lösungen auf diesem Gebiet vorgestellt und verglichen. Die Stärken und Schwächen verwandter Arbeiten sind bei der Konzeptionierung des in dieser Arbeit entwickelten Ansatzes miteingeflossen.

Die Grundidee des Ansatzes beruht auf der Verwendung von Minimaps, die Orte, die außerhalb des Viewports liegen, repräsentieren. Für die Positionierung der Minimaps wurden bekannte Graphenlayout-Algorithmen angepasst und erweitert. Das zugrundeliegende Kräftemodell für die Positionsbestimmung der Minimaps berücksichtigt nicht nur Relationen zwischen Knoten, sondern auch die ursprüngliche Platzierung der Orte, die durch Minimaps repräsentiert werden.

Eine darauffolgende Kollisionsbehebung wird angewandt, um bestehende Überlagerungen unter Minimaps und zwischen ihnen und Markern zu lösen. Zum Schluss werden die Minimaps entlang ihrer Breiten- und Längengrade sortiert, um zu verhindern, dass Minimaps, die Orte weiter im Süden repräsentieren, weiter unten gezeichnet werden als welche, die Orte weiter im Norden repräsentieren (genauso für Ost und West beziehungsweise links und rechts).

Um Visual Clutter zu reduzieren, wurden zwei Konzepte zum Clustern von Minimaps vorgestellt. Beim einen werden Minimaps nach ihrem politischen und beim anderen nach ihrem geographischen Kontext geclustert.

Für die Bestimmung der Form der Minimaps und für die Visualisierung der Kanten, um die Richtung der Relationen anzuzeigen, wurden verwandte Arbeiten auf diesem Gebiet untersucht. Um die lupenähnliche Funktion der Minimaps zu unterstreichen, wurde eine Kreisform für Minimaps gewählt, die

weitere Vorteile mit sich bringt. Verbindungen werden durch Kanten zu Minimapen, beziehungsweise zu Markern auf der Hauptkarte, gezeichnet, wobei die Verbindungsrichtung innerhalb der Marker durch kleine Pfeile dargestellt wird.

Neben gewöhnlichem Zooming und Panning wurden viele weitere Interaktionen, beispielweise fürs Auswählen eines Knotens oder zum Anzeigen des Labels eines Knotens, implementiert. Eine zusätzliche Sidebar, die links von der Hauptkarte angedockt ist, liefert zusätzliche Informationen über den ausgewählten Teilgraphen und weitere Interaktionsmöglichkeiten für die Exploration des gesamten Graphen.

Das Konzept wurde mit Hilfe einer Nutzerstudie, an der insgesamt 23 Personen teilgenommen haben, evaluiert. Dazu haben die Probanden das implementierte Tool, das Minimapen zur Verwendung weit entfernter Orte verwendet, mit einer simpleren Variante verglichen, die keine Minimapen verwendet.

Der in dieser Arbeit erst konzipierte, dann implementierte und zum Schluss evaluierte Ansatz zum Explorieren von Netzwerken von Daten mit geographischem Bezug ermöglicht dem Nutzer einen direkten Überblick über einen Knoten und seinen Relationen, sobald dieser ausgewählt wurde. Die benötigte Interaktion, um sowohl regionale als auch transkontinentale Verbindungen zu untersuchen, ist geringer als bei simplen Ansätzen, wie beispielsweise bei Übersicht + Details. Diverse Interaktionsmöglichkeiten stehen dem Nutzer dennoch zur Verfügung, um einen Knoten zu selektieren und weitere Details zu erhalten.

## 5.1 Ausblick

Trotz der größtenteils sehr positiven Rückmeldungen aus der Nutzerstudie gibt es einige Themen, die in Zukunft genauer untersucht, verbessert und gegebenenfalls implementiert werden sollten, welche im Folgenden aufgelistet sind.

## Dynamische Anpassung des Kräftemodells

Das zugrunde liegende Kräftemodell für die Berechnung der Position der Minimaps verwendet konstante Koeffizienten  $k_{map}$ ,  $k_{node}$ ,  $k_a$  und  $k_{orig}$  für die Kräfte  $F_{r_{map}}$ ,  $F_{r_{node}}$ ,  $F_a$  und  $F_{a_{orig}}$ . Die feste Wahl der Koeffizienten führt dazu, dass die Platzierung der Minimaps in manchen Fällen sehr gut und in anderen dafür weniger gut ausfällt. Eine dynamische Anpassung der Koeffizienten, beispielsweise anhand einer Bewertungsfunktion für die Platzierung der Minimaps, könnte das Problem beheben, das in der Nutzerstudie von einigen Probanden erwähnt wurde, dass Minimaps manchmal nicht in Richtung ihrer Orte, die sie repräsentieren, platziert sind.

## Form von Minimaps

In dieser Arbeit wurden für Minimaps lediglich Kreise oder Quadrate als mögliche Formen in Betracht gezogen und untersucht. Weitere Formen, wie Rechtecke oder sogar Polygone, werden in ähnlichen Arbeiten, zum Beispiel Rechtecke bei Brodkorb et al. (BKA<sup>+</sup>16), verwendet. Es wäre durchaus denkbar, dass unterschiedliche Formen in verschiedenen Fällen hilfreich wären, um Informationen über Orte, die außerhalb des Viewports liegen, besser zu visualisieren.

## Verbesserung der Kollisionsbehebung

Für die Kollisionsberechnung wird in dieser Arbeit davon ausgegangen, dass Minimaps eine runde Form haben, was die Kollisionsberechnung gegenüber Minimaps mit unterschiedlichen Formen erleichtert. In der Arbeit von Lin et al. (LG98) werden unterschiedliche Verfahren vorgestellt, die für die effiziente Ermittlung von Kollisionen unterschiedlicher geometrischer Körper verwendet werden können.

Bei der implementierten Kollisionsbehebung werden Minimaps, je nachdem wo sie sich befinden, entlang der Kanten des Viewports verschoben. Eine Verschiebung in Richtung der eigentlichen Orte, die durch die zu verschiebenden Minimaps repräsentiert werden, anstelle einer fixen Richtung, würde die Platzierung der Minimaps weiter verbessern.



## Zoomlevel von Minimaps

Bei Minimaps, die mehrere Orte enthalten, wird das Zoomlevel so gewählt, dass es möglichst groß ist, aber dennoch alle Orte, die durch die Minimap repräsentiert werden sollen, enthält. Für Minimaps, die nur einen Ort enthalten, stellt sich die Wahl der optimalen Zoomstufe, als durchaus schwierig heraus. Harrison et al. (HD08) betrachten für einen anderen Use-case, bei dem sie die optimale Zoomstufe ermitteln möchten, nicht nur den Inhalt, der dargestellt werden soll, sondern auch äußere Faktoren, wie beispielsweise die Entfernung des Nutzers zum Bildschirm, mit Hilfe einer Kamera. Die Wahl des Zoomlevels bei Karten stellt sich bereits ohne Einbeziehung äußerer Faktoren als schwierige Aufgaben heraus. Je nach Ort und Inhalt ist ein anderes Zoomlevel notwendig, um ausreichend viele Details und Informationen wiederzugeben. In Abbildung 37 ist für die Minimap in der oberen linken Ecke so gut wie nichts, mit Ausnahme des Markers, erkennbar. Das gleiche Zoomlevel reicht für die anderen Minimaps im selben Bild, die nur einen Ort repräsentieren, aus, um zu erkennen, welcher Ort repräsentiert wird. Städt- und Ländergrenzen genauso wie Flüsse oder Meere helfen bei der Identifizierung eines Ortes. Mit adaptiven Zooming, das beispielsweise in der Arbeit von Cecconi et al. (CG02) genauer untersucht wird, lassen sich Zoomlevel so wählen, dass die entsprechenden Kartenausschnitte aussagekräftig genug sind, um den repräsentierten Ort zu identifizieren. Ein so gewähltes Zoomlevel für Minimaps, die nur einen Knoten repräsentieren, würde weiter zur Verbesserung des implementierten Konzepts führen. Eine zusätzliche visuelle Auskunft über die Zoomstufe der Minimaps und der Hauptkarte, könnte für den Nutzer eine weitere wichtige Information sein, die in der Zukunft untersucht werden sollte.

## Labels

Labels, die Auskünfte über die Städtenamen geben, werden in der implementierten Version eingeblendet, wenn der Nutzer sich mit der Maus über den entsprechenden Markern befindet. Sobald der Nutzer die Maus vom Marker entfernt, verschwindet auch das zugehörige Label. Dass der Nutzer erst mit der Maus über den Marker navigieren muss, ist eine zusätzliche Interaktion, die nicht benötigt ist, wenn alle Labels zu jeder Zeit sichtbar sind. Diese nehmen allerdings viel Platz in Anspruch und ihre Positionierung ist nicht trivial, wenn sie keine wichtigen Informationen überdecken sollen. Mit der Positionierung von Labels beschäftigen sich Forscher bereits seit vielen Jahren. Neyer et al (Ney01) untersuchen dabei die Platzierung von Labels auf Weltkarten, was für zukünftige Arbeiten auf diesem Gebiet durchaus relevant sein kann.

## Kantenwahl

Im Kapitel für Evaluation wurden bereits zwei unterschiedliche Möglichkeiten zur Darstellung von Kantenrichtungen bei Graphen verglichen. Dabei gab es die größten Meinungsunterschiede. Während ungefähr die Hälfte der Probanden den gewöhnlichen Ansatz, bei dem Knoten über Kanten mit Pfeilen zu Beginn oder Ende der Kanten, verbunden werden, bevorzugen, empfand die andere Hälfte die Darstellung der Pfeile innerhalb von Markern als besser. In der Arbeit von Holten et al. (HVW09) werden weitere Möglichkeiten, wie beispielsweise Tapered Edges, vorgestellt, die genauer untersucht werden sollten. Kanten komplett wegzulassen und dafür nur Pfeile innerhalb der Marker zu benutzen, scheint aber ebenfalls ein möglicher Weg zu sein, um Verbindungsrichtungen zu visualisieren. Zuletzt sollte außerdem überprüft werden, wie sich weitere Informationen, wie beispielsweise die Distanz von  $p_{curr}$  zu anderen Knoten, visualisieren lässt. Unterschiedliche Farben oder Breiten der Kanten wären mögliche Ansätze hierfür.

## References

- [BKA<sup>+</sup>16] BRODKORB, Felix ; KUIJPER, Arjan ; ANDRIENKO, Genady ; ANDRIENKO, Natalia ; VON LANDESBERGER, Tatiana: Overview with details for exploring geo-located graphs on maps. In: *Information Visualization* 15 (2016), Nr. 3, S. 214–237
- [BR03] BAUDISCH, Patrick ; ROSENHOLTZ, Ruth: Halo: a technique for visualizing off-screen objects. In: *Proceedings of the SIGCHI conference on Human factors in computing systems* ACM, 2003, S. 481–488
- [CG02] CECCONI, Alesandro ; GALANDA, Martin: Adaptive zooming in web cartography. In: *Computer Graphics Forum* Bd. 21 Wiley Online Library, 2002, S. 787–799
- [CKB09] COCKBURN, Andy ; KARLSON, Amy ; BEDERSON, Benjamin B.: A review of overview+ detail, zooming, and focus+ context interfaces. In: *ACM Computing Surveys (CSUR)* 41 (2009), Nr. 1, S. 2
- [d3j] *d3js*. <https://d3js.org/>,
- [DBETT94] DI BATTISTA, Giuseppe ; EADES, Peter ; TAMASSIA, Roberto ; TOLLIS, Ioannis G.: Algorithms for drawing graphs: an annotated bibliography. In: *Computational Geometry* 4 (1994), Nr. 5, S. 235–282
- [FD13] FRISCH, Mathias ; DACHSELT, Raimund: Visualizing offscreen elements of node-link diagrams. In: *Information Visualization* 12 (2013), Nr. 2, S. 133–162
- [FR91] FRUCHTERMAN, Thomas M. ; REINGOLD, Edward M.: Graph drawing by force-directed placement. In: *Software: Practice and experience* 21 (1991), Nr. 11, S. 1129–1164

- [Fur86] FURNAS, George W.: *Generalized fisheye views*. ACM, 1986
- [GBGI08] GUSTAFSON, Sean ; BAUDISCH, Patrick ; GUTWIN, Carl ; IRANI, Pourang: Wedge: clutter-free visualization of off-screen locations. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* ACM, 2008, S. 787–796
- [goo] *Google Maps*. <https://www.google.com/maps>,
- [HD08] HARRISON, Chris ; DEY, Anind K.: Lean and zoom: proximity-aware user interface and content magnification. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* ACM, 2008, S. 507–510
- [HVW09] HOLTEN, Danny ; VAN WIJK, Jarke J.: Force-Directed Edge Bundling for Graph Visualization. In: *Computer graphics forum* Bd. 28 Wiley Online Library, 2009, S. 983–990
- [JKK15] JÄCKLE, Dominik ; KWON, Bum C. ; KEIM, Daniel A.: Off-Screen Visualization Perspectives: Tasks and Challenges. In: *Symposium on Visualization in Data Science (VDS) at IEEE VIS 2015*, 2015
- [KK<sup>+</sup>89] KAMADA, Tomihisa ; KAWAI, Satoru u. a.: An algorithm for drawing general undirected graphs. In: *Information processing letters* 31 (1989), Nr. 1, S. 7–15
- [lea] *Leaflet*. <http://leafletjs.com/>,
- [LG98] LIN, Ming ; GOTTSCHALK, Stefan: Collision detection between geometric models: A survey. In: *Proc. of IMA conference on mathematics of surfaces* Bd. 1, 1998, S. 602–608
- [lol] *League of Legends*. [https://play.euw.leagueoflegends.com/de\\_DE](https://play.euw.leagueoflegends.com/de_DE),

- [LSC08] LUBOSCHIK, Martin ; SCHUMANN, Heidrun ; CORDS, Hilko: Particle-based labeling: Fast point-feature labeling without obscuring other visual features. In: *IEEE transactions on visualization and computer graphics* 14 (2008), Nr. 6, S. 1237–1244
- [NBM<sup>+</sup>06] NEKRASOVSKI, Dmitry ; BODNAR, Adam ; MCGRENERE, Joanna ; GUIMBRETIERE, François ; MUNZNER, Tamara: An evaluation of pan & zoom and rubber sheet navigation with and without an overview. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems* ACM, 2006, S. 11–20
- [NBW14] NETZEL, Rudolf ; BURCH, Michel ; WEISKOPF, Daniel: Comparative eye tracking study on node-link visualizations of trajectories. In: *IEEE transactions on visualization and computer graphics* 20 (2014), Nr. 12, S. 2221–2230
- [Ney01] NEYER, Gabriele: Map labeling with application to graph drawing. In: *Drawing graphs*. Springer, 2001, S. 247–273
- [SB92] SARKAR, Manojit ; BROWN, Marc H.: Graphical fisheye views of graphs. In: *Proceedings of the SIGCHI conference on Human factors in computing systems* ACM, 1992, S. 83–91
- [TAVHS06] TOMINSKI, Christian ; ABELLO, James ; VAN HAM, Frank ; SCHUMANN, Heidrun: Fisheye tree views and lenses for graph visualization. In: *Information Visualization, 2006. IV 2006. Tenth International Conference on IEEE*, 2006, S. 17–24
- [vis] *Microsoft Visio*. <https://products.office.com/visio>,
- [YKR06] YOUNIS, Ossama ; KRUNZ, Marwan ; RAMASUBRAMANIAN, Srinivasan: Node clustering in wireless sensor networks: Recent developments and deployment challenges. In: *IEEE network* 20 (2006), Nr. 3, S. 20–25

## Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.<sup>1</sup>

---

(Ort, Datum, Unterschrift)

---

<sup>1</sup>I hereby declare that the work presented in this thesis is entirely my own. I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.