Institut für Visualisierung und Interaktive Systeme
Universität Stuttgart
Universitätsstraße 38
70569 Stuttgart
Germany

Diplomarbeit Nr. 3320

Erweiterung der Themeriver Visualisierung um dynamische Relationen

Qi Hu

Studiengang:	Informatik
Prüfer:	Prof. Dr. Daniel Weiskopf
Betreuer:	Dr. rer. nat. Dipl.-Inf. Michael Burch
begonnen am:	23.04.2012
beendet am:	23.10.2012
CR-Klassifikation:	I.3.6, H.5.2

Zusammenfassung

Die sogenannte Themeriver Visualisierung bietet die Möglichkeit, mehrere zeitabhängige quantitative Werte visuell miteinander zu vergleichen und wachsende, fallende oder konstante Phasen in einem oder mehreren quantitativen Werten schnell zu erkennen. Allerdings geht bei der Standarddarstellung typischerweise die Information verloren, welche Variablen miteinander in Bezug stehen und zu welchem Ausmaß. Dies bedeutet, dass ein dynamischer gerichteter und gewichteter Graph zusätzlich in die Standarddarstellung integriert werden muss, um auch Einsichten über das relationale Verhalten unter den Werten zu bekommen.

In den letzten Jahren wurden leistungsfähige Methoden entwickelt, die große Datenmengen visualisieren können. Auch für die Darstellung zeitlicher Verläufe sind verschiedene Techniken bekannt. Die Verbindung dieser Techniken mit einer Zusatzdarstellung für relationale Daten ist jedoch bisher noch offener Forschungsgegenstand. In der vorliegenden Diplomarbeit wird eine geeignete interaktive Erweiterung der Themeriver Visualisierung entwickelt. Dabei wird das TimeArcTrees Verfahren [GBD09] in dem Visualisierungswerkzeug verwendet, um interessante Einsichten in die dynamischen Abhängigkeiten der sich zeitlich verändernden quantitativen Werte zu bekommen. Interaktive Features unterstützen den Analytiker solcher Daten hierbei durch die Manipulation der visuellen Darstellung in allen dargestellten Dimensionen, um weitere Einsichten in die Daten zu erlangen, die mit einem statischen Diagramm nicht gewonnen werden können.

Abstract

The so-called Themeriver visualization offers the possibility to compare several time-dependent quantitative values to each other visually and to recognize increasing, decreasing or constant phases in one or more quantitative values quickly. However, the information is typically lost by the standard representation, which variables are correlated and to what extent. This means that a dynamic directed and weighted graph can be additionally integrated into the standard representation in order to get insights about the relational behavior among the values.

In recent years, powerful methods have been developed to visualize large data sets. Also for the representation of temporal processes, various techniques are known. The combination of these techniques with an additional visualization of relational data is an open and challenging research topic. In this thesis, an appropriate extension of the interactive Themeriver visualization will be developed. The TimeArcTrees method [GBD09] is used in the visualization tool to get interesting insights into the dynamic relations between variables of time-varying quantitative values. Interactive features support the analyst of such data in this case by the manipulation of the visual representation in all shown dimensions to obtain further insights in the data that cannot be gained by just inspecting the static diagram.

Inhaltsverzeichnis

1. Einleitung

Heutzutage werden Visualisierungsmethoden nahezu unverzichtbar im Bereich der Informationstechnologie wie E-Commerce, Data Warehousing, soziale- oder Computer-Netzwerke eingesetzt. Beispielsweise wird ein virtueller Automobil-Crashversuch in Form interaktiver 3D-Diagramme dargestellt oder sich die Wettervorhersage im Fernsehen mit Grafiken betrachtet. Die heutige Gesellschaft definiert sich im Zeitalter der Informationsflut. Mit der Entwicklung von Netzwerk-Informationstechnologien entsteht eine Vielzahl von Informationen in einem sozialen Netzwerk. Zur Auswertung von sozialen Netzwerkdaten kann man deshalb auch die Vorteile der Visualisierung nutzen. Soziale Netzwerkdaten haben nicht nur einen relationalen Charakter („Mit welchem anderen Wort ist ein Wort aufgetreten?“), sondern enthalt auch eine zeitliche Komponente („Wann ist ein Wort mit einem anderen gemeinsam aufgetreten?“).

Die Analyse sozialer Netzwerke hat sich in den letzten Jahren zu einem Forschungsfeld von zunehmendem Interesse entwickelt [Jan06]. „Es geht darum, Einsichten in die Handlungsweisen der einzelnen Akteure einer definierten Personengruppe (Netzwerk) zu gewinnen. Durch eine Untersuchung dieser Netzwerke werden Erkenntnisse über die Ursachen und Wirkungsweisen von Verhaltensmustern gestellt. Beispielsweise könnte man analysieren, unter welchen Bedingungen Jugendliche mit dem Rauchen anfangen. Um strukturelle Merkmale von sozialen Netzwerken zu veranschaulichen, finden Visualisierungen, vor allem in graphischer Form, ihren Einsatz und dienen den jeweiligen Wissenschaftlern als aussagekräftiges Hilfsmittel bei der Exploration und Auswertung der Daten. Untersucht wird, wie die einzelnen Akteure miteinander verknüpft sind, welchen Einfluss sie auf ihr Umfeld nehmen und in wie weit sie durch ihr Umfeld beeinflusst werden.“ (vgl. [Son08])

Dennoch ist die Visualisierung dynamischer, sozialer Netzwerke momentan eher als Nische zu sehen, auf die sich erst in den letzten Jahren vermehrt Forschungsaktivitäten richten. Somit stehen noch viele Fragen offen, die genauer erforscht werden müssen. Hier werden die sozialen Netzwerkdaten in einer Themeriver Visualisierung dargestellt. Themen sind dabei z.B. einzelne Wörter. Ein Vorteil dieser Darstellungsvariante ist eine hohe Übersichtlichkeit und rasche Informationsübertragung auf den Anwender. Allerdings ist sie ungeeignet, um Einsichten, insbesondere wie in unserem Fall erwünscht, in die dynamischen Abhängigkeiten der sich zeitlich verändernden quantitativen Werte zu bekommen.

In der vorliegenden Diplomarbeit wird eine geeignete interaktive Erweiterung der Themeriver Visualisierung entwickelt, wobei der Benutzer die Darstellung entsprechend seines Interesses mit Hilfe interaktiver Features anpassen kann. Schwerpunkte der Interaktivität sind die Einstellbarkeit der Darstellungsanzahl der am häufigsten aufgetretenen Wörter und die Einstellbarkeit des Darstellungsintervalles der gerichteten Relationsbögen (curved links) anhand der dynamischen Abhängigkeiten.

1.1 Motivation

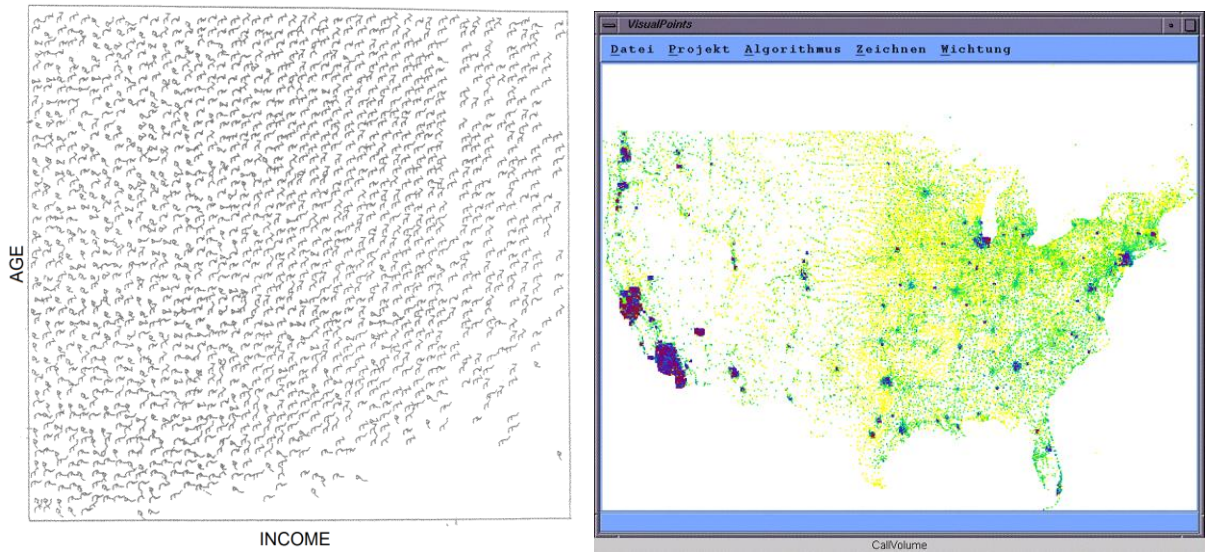
Moderne Datenbanksysteme speichern heutzutage immense Datenmengen. Forscher an der Universität California in Berkeley haben eine Studie [LV03] veröffentlicht, die besagt, dass im Jahr 2002 ca. 5 Exabyte (= 5 Million Terabyte) an Informationen produziert werden - ein

großer Teil davon in digitaler Form (Print, Film, magnetische und optische Speichermedien). Das heißt aber, dass die Menge der in der gesamten menschlichen Entwicklung zuvor generierten Daten in den nächsten drei Jahren verzweifacht werden. Die Daten werden oft automatisch mit Hilfe von Sensoren und Überwachungssystemen aufgezeichnet. So werden beispielsweise alltägliche Vorgänge des menschlichen Lebens, wie etwa das Bezahlen mit EC-Karte oder die Benutzung des Internets, durch Computer aufgezeichnet. Dabei entstehen gewöhnlich hochdimensionale Datensätze bei der Speicherung aller vorhandenen Parameter. Durch Visualisierung wird versucht, die Gewinnung von wertvollen Informationen, die einen Wettbewerbsvorteil bieten können, aus solchen großen Datensätzen zu vereinfachen. Heutige Datenbankmanagementsysteme stehen aber noch nicht für die Darstellung dieser riesigen Datenmengen zur Verfügung. Werden die Daten Beispielsweise in textueller Form dargestellt, reichen höchstens ein paar hundert Zeilen auf dem Bildschirm offenbar nicht aus. Bei Millionen von Datensätzen entspricht dies aber nur einem „Staubkorn im Universum“. Hierzu erfolgt eine Abbildung der Daten auf visuelle Objekte, so dass sie möglichst einfach und intuitiv vom menschlichen visuellen System erfassbar sind.

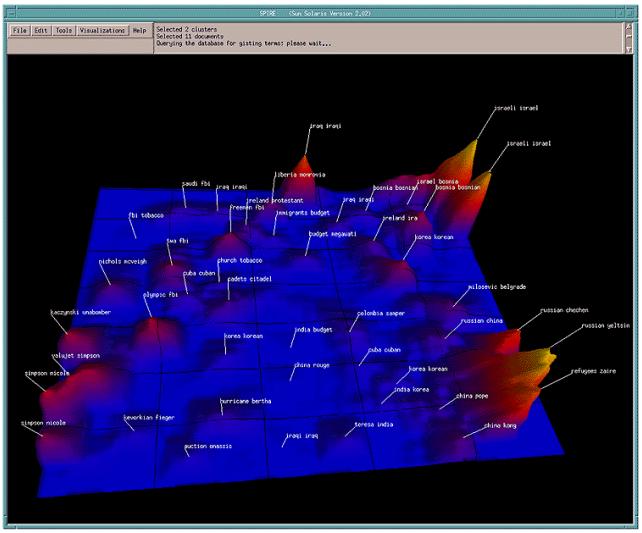
Nach [Kei02] gilt „Für ein effektives Data Mining ist es wichtig, den Menschen in den Datenexplorationsprozess mit einzubinden, um die Fähigkeiten des Menschen - Flexibilität, Kreativität und das Allgemeinverständnis - mit den enormen Speicherkapazitäten und Rechenleistungen moderner Computersysteme zu kombinieren.“ Die Grundidee der visuellen Datenexploration (vgl. [Kei02]) ist die geeignete Darstellung der Daten in visueller Form, die es dem Menschen erlauben, einen Einblick in die Struktur der Daten zu bekommen, Schlussfolgerungen aus den Daten zu ziehen, sowie direkt mit den Daten zu interagieren und Hypothesen über die Daten aufzustellen. Dabei kann mit stark inhomogenen und verrauschten Daten gearbeitet werden und die Datenexploration kann auch durch Laien durchgeführt werden (siehe Abbildung 1.1).

„Visuelle Data Mining-Verfahren haben in den letzten Jahren einen hohen Stellenwert innerhalb des Forschungsbereiches Data Mining erhalten. Ihr Einsatz ist immer dann sinnvoll, wenn wenig über die Daten bekannt ist und die Explorationsziele nicht genau spezifiziert sind. Dadurch, dass der Mensch direkt am Explorationsprozess beteiligt ist, können die Explorationsziele bei Bedarf verändert und angepasst werden. Die visuelle Datenexploration kann als ein Prozess zur Generierung von Hypothesen aufgefasst werden. Sie ermöglicht dem Menschen ein tieferes Verständnis für die Daten, wodurch er neue Hypothesen über die Daten aufstellen kann. Die Hypothesen können dann wiederum mit Hilfe visueller Datenexplorationsverfahren untersucht und verifiziert werden. Die Verifikation kann jedoch auch mit Hilfe von Techniken aus dem Bereich der Statistik und der künstlichen Intelligenz durchgeführt werden.“[Kei02]

Zusammenfassend kann man feststellen, dass die visuelle Datenexploration in vielen Fällen eine einfachere Untersuchung der Daten erlaubt und oft auch bessere Ergebnisse erzielt, insbesondere wenn die herkömmlichen automatischen Algorithmen nur unzureichende Ergebnisse liefern. Die visuelle Datenexploration bietet darüberhinaus ein besseres Verständnis des Datenexplorationsprozesses sowie der erzielten Ergebnisse. Visuelle Datenexplorationstechniken werden deshalb in vielen Anwendungsbereichen eingesetzt und in Verbindung mit automatischen Algorithmen sind sie ein unentbehrliches Verfahren zur Exploration wichtiger Informationen aus großen Datenbanken.



(a) (b)



(c)

Abbildung 1.1: (a) Eine Strichmännchenvisualisierung stellt Zensusdaten der USA dar. Hierbei sind die Strichmännchen auf der horizontalen Achse nach den Einnahmen und auf der vertikalen Achse nach dem Lebensalter eingeteilt. (b) X-Y Plot zeigt Telefondaten in den USA. Sie ist beispielsweise geeignet für die überlappungsfreie Visualisierung Geographie-basierter Daten. (c) Das Bild zeigt eine große Anzahl von Dokumentkollektionen als Künstliche Landschaft. Hierbei repräsentieren Berge in der Form einer 3D-Version des Themerivers häufig auftretende Themengebiete. Quelle: Keim, 2002, S. 33-36 [Kei02]

Auf die Darstellung zeitlicher Bezüge sind Themeriver spezialisiert. Sie bieten ein breites Spektrum von Techniken zur Präsentation zeitabhängiger quantitativer Daten. Die Darstellung von dynamischen Relationen auf Graphen ist bisher jedoch nur wenig untersucht worden.

Dies gilt insbesondere für die gleichzeitige Darstellung mehrerer Parameter. Mögliche Kombinationen von Zeit- und Relationsdarstellung werden im Folgenden untersucht.

1.2 Aufgabenstellung

Die Aufgabe in der vorliegenden Diplomarbeit besteht darin, eine geeignete interaktive Erweiterung der Themeriver Visualisierung zu entwickeln, die

- Graphdaten in einem speziellen Format einlesen kann,
- die angezeigten Themeriver interaktiv manipulierbar in der Variablen- und Zeitdimension macht,
- den dynamischen Graph als eindimensionales Knoten-Kanten-Diagramm in Form von Curved Links (Arcs) visualisieren kann,
- einen Algorithmus verwendet, der die Kantenlängen verkürzt,
- Filterfunktionen für die dynamischen Graphen bereitstellt.

Das Visualisierungswerkzeug soll mithilfe mehrerer interaktiver Funktionen dem Benutzer die Möglichkeit bieten, schnelle und interessante Einsichten in die dynamischen Abhängigkeiten der sich zeitlich verändernden quantitativen Werte zu liefern, die auf einer gegebenen Datei gespeichert sind. Eine der wichtigsten Funktionen hierbei ist die Darstellung des ausgewählten interessanten Themenflusses. Die aktuell eingegebene Anzahl der häufigsten aufgetretenen Flüsse bestimmt die auf dem Graph darzustellenden Flüsse. Eine weitere wichtige Funktion soll das Darstellungsintervall der gerichteten Relationsbögen anhand der dynamischen Abhängigkeiten sein.

Die Implementierung des Visualisierungswerkzeuges wird in der Programmiersprache JAVA erfolgen unter Benutzung der Graphikbibliotheken AWT und Swing.

1.3 Aufbau der Diplomarbeit

Im letzten Abschnitt des Einführungskapitels soll nun der Aufbau der Arbeit vorgestellt werden. Die Arbeit gliedert sich in folgender Weise:

Kapitel 2 - Grundlagen der menschlichen Wahrnehmung: Im Kapitel sollen die Grundlagen für diese Arbeit benannt und erklärt werden. Dabei werden Grundkenntnisse der menschlichen Wahrnehmung dargestellt, weil zur Visualisierung der Themeriver eine geeignete Farbauswahl nötig ist.

Kapitel 3 - Verwandte Arbeiten: Ein Überblick über verwandte Themen, die Überschneidungen mit dem Themeriver Konzept dieser Arbeit aufweisen, wird in diesem Kapitel gegeben. Auch dynamische Graphen und Interaktionstechniken werden beschrieben.

Kapitel 4 -Visualisierungstechnik: In diesem Kapitel wird eine konzeptionelle Vorgehensweise definiert. Es wird dabei nicht beschrieben, was gemacht werden soll, sondern es geht darum, wie vorgegangen werden muss, um die Aufgabe zu lösen. Zunächst wird eine Einführung in das Thema Themeriver gegeben, wobei die Geschichte, die Anwendungsbereiche und

die Einschränkungen des Themerivers kurz vorgestellt werden. Desweiteren werden die relevanten Algorithmen, interaktive Features, Skalierbarkeit und Performanz des Themeriver-Systems genau beschrieben.

Kapitel 5 - Implementierung: Dieses Kapitel befasst sich mit der Implementierung eines prototypischen Werkzeugs zur Datenvisualisierung. Einige ausgewählte Implementierungsdetails werden darin genauer behandelt.

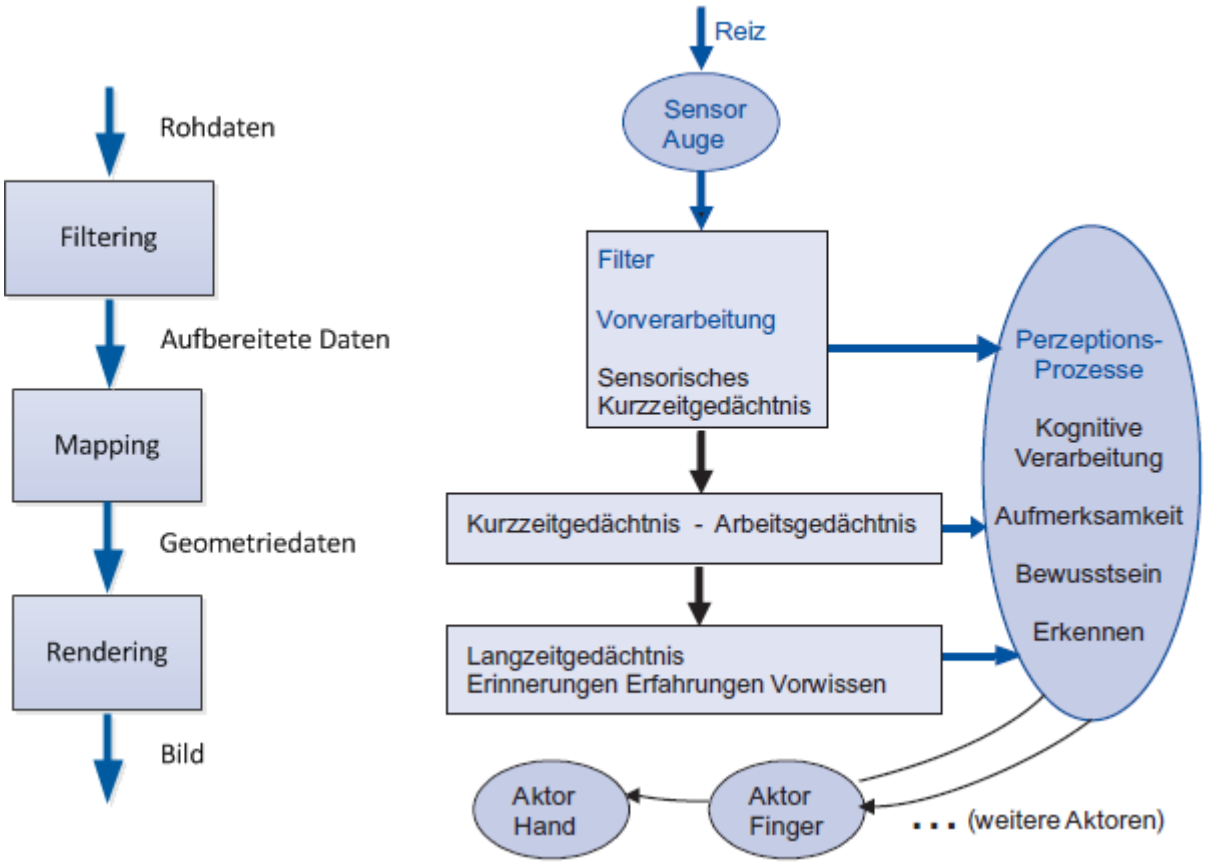
Kapitel 6 - Fallstudie: In diesem Kapitel wird eine Fallstudie vorgestellt und Anwendungsfall des Werkzeuges mithilfe eines Datensatzes präsentiert. Dabei werden die wichtigsten Funktionen benutzt, um aussagekräftige Visualisierungen zu erzeugen.

Kapitel 7 - Zusammenfassung und Ausblick: Abschließend werden die entwickelten Bausteine kritisch reflektiert und die im Rahmen dieser Arbeit gewonnenen Erkenntnisse zusammengefasst. Weiterhin wird ein Ausblick auf weitere potenzielle Ansätze zur Optimierung der entworfenen Prototypen geliefert.

2. Grundlagen der menschlichen Wahrnehmung

Die Visualisierungspipeline (siehe Abbildung 2.1(a)) spezifiziert den Prozessvorgang, mittels derer Daten über Geometrie in Bilder umgewandelt werden. Sie besteht aus einem dreistufigen Prozess, Filtern und Bereinigen von Daten, Abbilden der Daten auf Geometrien und Rendern dieser Objekte. „Die Qualität einer Visualisierung definiert sich durch den Grad, in dem die bildliche Darstellung das kommunikative Ziel der Präsentation erreicht. Sie lässt sich als Verhältnis von der vom Betrachter in einem Zeitraum wahrgenommenen Information zu der im gleichen Zeitraum zu vermittelnden Information beschreiben.“ [SM00]

Es muss zunächst einmal die menschliche Wahrnehmung verstanden werden, um bestmögliche Visualisierungen erstellen zu können. Hier wird die Wahrnehmung von Menschen erklärt, die vom Visualisierungskonzept vorausgesetzt wird. Im Wesentlichen erfolgt die Wahrnehmung von äußeren Reizen über die fünf Sinne (Sehen, Hören, Riechen, Schmecken, Tasten) des Menschen. Nach [Dah06] gilt, dass die Sinneseindrücke, in Computergrößen umgerechnet, mit einem Datendurchsatz von ca. 8 MBit/s verarbeitet werden - dabei liefert circa 80% das Auge. Da sich auch eine Visualisierung, wie der Name schon sagt, mit der visuellen Darstellung beschäftigt, wird im Folgenden lediglich die Funktionsweise des menschlichen Auges beschrieben.



(a) (b)

Abbildung 2.1: (a) Visualisierungspipeline (b) Einfaches Blockschaltbild der menschlichen Informationsverarbeitung. Quelle [Dah06], verändert

Abbildung 2.1 (b) zeigt den Prozessvorgang der Wahrnehmung über den Sehsinn. Primär ist das Auge als Sensor an der Wahrnehmung beteiligt und nimmt die visuellen Reize von der Umgebung auf. Anschließend werden die Reize als Nervenimpulse in das sensorische Kurzzeitgedächtnis weitergeleitet, von dort werden die Informationen ausgelesen und zuerst an das Kurzzeitgedächtnis, dann schließlich an das Langzeitgedächtnis überführt. In jedem Schritt wird die eingehende Information mit den vorliegenden gespeicherten Informationen abgeglichen. Durch diesen Vorgang des Abgleichens wird die noch nicht gefasste Information abgespeichert. Dann wird das Gesehene erst wahrgenommen. Dies bedeutet, konkrete Objekte werden vom Gehirn mit Hilfe solcher verarbeiteten Informationen und gesammelten Erfahrungen erkannt. Beispielsweise kann ein Objekt unkorrekt im Bild repräsentiert sein und, wenn möglich, wird der fehlerhafte Teil wird durch die im Gedächtnis gespeicherten Erfahrungen korrigiert. Sofern das Gesehene eine Handlung in Anspruch nimmt, so folgt der Handlungsbefehl zu einem Aktor, z.B. einer Hand bzw. einem Finger.

2.1 Aufbau des Auges

In der Abbildung 2.2 stellt eine vereinfachte Version den schematischen Aufbau des Auges dar. Der Sehvorgang beginnt mit dem Eintreffen eines Lichtstrahls auf der Linse. Diese bricht das Licht so, dass ein komplettes Bild auf die Netzhaut abgebildet wird. Lichtempfindliche Rezeptoren, Stäbchen und Zapfen, unterstützen uns dabei, dieses Bild wahrzunehmen und über den Sehnerv ins Gehirn zu leiten. Stäbchen reagieren dabei sehr fein auf Graustufen und bilden insgesamt 95% aller Rezeptoren auf der Netzhaut. Die restlichen 5% befinden sich als sogenannte Zapfen in der Netzhaut. Diese dienen als Verantwortlicher des Farbsehens, so dass Helligkeitsabstufungen auf den drei Farbkanälen Rot, Grün und Blau interpretiert werden können. Die Fovea, welche im Herz der Netzhaut vorhanden ist und dem Menschen die Möglichkeit zum scharfen Sehen bietet, besteht aus dem Großteil der Zapfen.

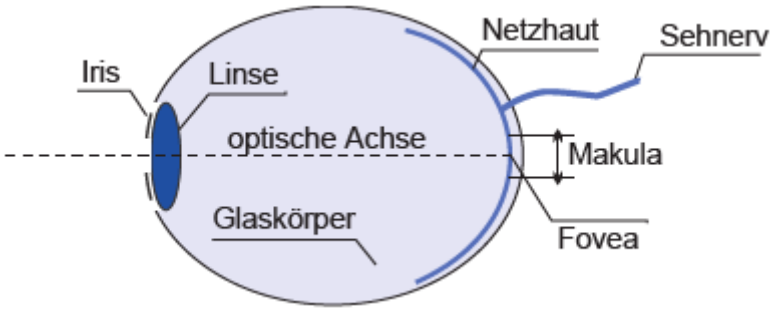


Abbildung 2.2: Ein Querschnitt des Auges. Quelle [Dah06]

2.2 Wahrnehmung durch das Auge

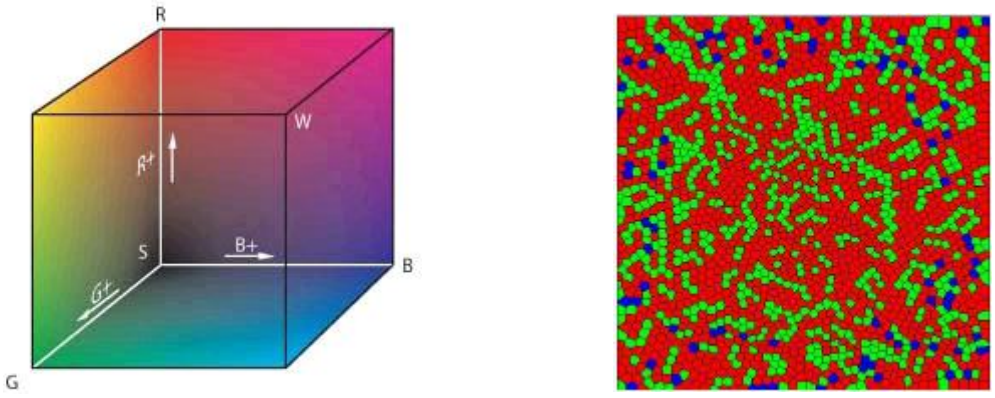
Die drei Fähigkeiten der Hell-Dunkel-Sehens, Farbsehens und Scharfsehens spielen eine entscheidende Rolle, um ein gesehenes Bild im Gedächtnis darzustellen.

Hell-Dunkel-Sehen Wie bereits zur Kenntnis genommen, ermöglichen die Stäbchen das Wahrnehmen der Helligkeitsabstufungen. Nach [Dah06] gilt, dass diese nur ca. 200 bis 250 Graustufen unterscheiden können. Darüberhinaus kann dieser Kontrastumfang durch die Regulierung der Irisöffnung (Pupille) weiter ergänzt werden. Die Weite der Iris verändert sich, dadurch wird die relative Lichtmenge angepasst. Unter Adaption versteht man die Fähigkeit des Auges, sich an unterschiedliche Helligkeitsstufen anzupassen. Dies reguliert die Licht-

menge, die auf die Netzhaut trifft. Aus diesem Grund gilt der Kontrastumfang zu jeder Helligkeitsanpassung der Iris öffnung und summiert sich dadurch auf über 100.000 unterscheidbare Graustufen.

Farbsehen Die Farbwahrnehmung wird durch die Zapfen geregelt. Im menschlichen Auge existieren drei verschiedene Zapfenarten: jeweils eine für rotes, blaues und grünes Licht. Die Information, die also jeder Zapfen bietet, ist die Höhe der Intensität der roten, blauen, bzw. grünen Farbe. Durch Addieren der Intensitäten von drei nah beieinander liegenden unterschiedlichen Zapfen können auch Farben dargestellt werden, die nicht zu diesen drei Grundfarben gehören. Der Raum, der durch diese Farben aufgespannt wird, wird im Allgemeinen als RGB-Farbraum bezeichnet (siehe auch Abbildung 2.3 (a)). Da sämtliche Anzeigegeräte, wie Monitor, Fernseher und Smartphone üblicherweise auf RGB arbeiten, wird in der graphischen Datenverarbeitung oftmals das additive RGB-System verwendet.

Die Zapfen kommen verstärkt in der Fovea vor. Eine ungefähre Verteilung der einzelnen Zapfentypen stellt die Abbildung 2.3 (b) dar. Dabei ist auffällig, dass blaue Rezeptoren sehr rar besetzt sind, sie bilden nur ca. neun Prozent der Gesamtheit aller Zapfen. Die Rezeptoren für rotes und grünes Licht bilden den Rest, wobei Rotzapfen doppelt so häufig vorkommen wie Grünzapfen (vgl. Abbildung 2.3 (b)). Aber auch außerhalb der Fovea existieren Zapfen, obgleich ihre Auflösung dort sehr viel geringer ist. Ein Objekt wird im sogenannten peripheren Blickfeld nur dann als farbig wahrgenommen, wenn seine Farbe bereits bekannt ist oder es groß genug ist, um durch die geringe Auflösung der Zapfen abgetastet zu werden.



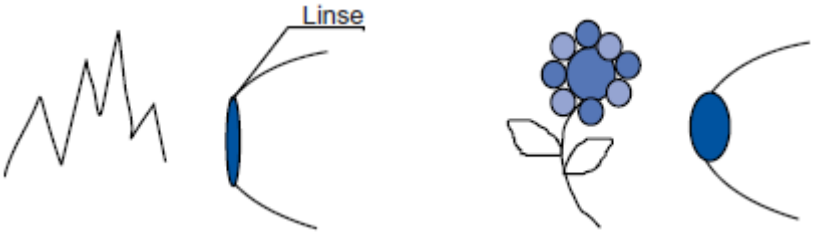
(a) (b)

Abbildung 2.3: (a) Der RGB-Farbraum in Würfeldarstellung. Dabei bilden die Grundfarben rot, grün und blau die Basisvektoren, so dass Mischfarben nur Linearkombinationen dieser sind. Quelle: http://de.wikipedia.org/wiki/RGB-Farbraum. (b) Zapfenverteilung in der Fovea. Die Farben beschreiben die Rezeptorart des jeweiligen Zapfens. Quelle [Geg12]

10% der Männer und 1% der Frauen weisen eine Farbschwäche auf. Farbwahrnehmung ist durch das Fehlen der Zapfen nur eingeschränkt relevant im alltäglichen menschlichen Leben, da z.B. Farbenblinde das Manko oft jahrelang selbst nicht bemerken. In Unserem Fall zeigt die Themeriver Visualisierung verschiedene Variablen als deutlich gefärbte Datenströme, die

sich über die Zeit hinweg ändern. Aus diesem Grund muss das Farbschema so ausgewählt werden, dass es die Farbwahrnehmung bei der Themeriver Visualisierung relativ wenig einschränkt.

Scharfes Sehen Wie bereits erwähnt, liegt das Kernstück des scharfen Sehens, die sogenannte Fovea (siehe auch Abbildung 2.2), in der Mitte des Blickfeldes. Der Mensch verhält sich also immer bei der Fokussierung eines Objekts so, dass das Auge genau dorthin bewegt wird, damit der Bereich des Interesses auf den Bereich der Fovea abgebildet wird.



(a) (b)

Abbildung 2.4: Akkomodation der Linse für nahes Objekt in (a) und entferntes Objekt in (b). Quelle [Dah06]

Das Auge bricht das eintretende Licht genauso mit Hilfe einer Linse, dass ein scharfes Bild auf der Netzhaut entstehen kann. Wie in Abbildung 2.4 (a) und (b) nachvollzierbar, wird durch das Zusammenziehen des Muskels eine kürzere Brennweite erzielt und nahe Gegenstände können scharf abgebildet werden. Um entfernte Gegenstände scharf zu sehen, entspannen sich die Muskeln und die Linse wird flach gezogen.

Das bedeutet, dass unser Themeriver Graph so zu gestalten ist (eventuell mit Hilfe einer Zoom Funktion), damit das Bild mit möglichst wenig Verlust an Informationen in üblicher Größe in ca. 30 cm Abstand vom Auge ohne große Anstrengung betrachtet wird.

2.3 Einflussfaktoren der Wahrnehmung

Bisher wurden die grundsätzlichen Prinzipien der visuellen Wahrnehmung erläutert. Darauf aufbauend soll folgendes Kapitel über Phänomene des Sehens, die unter bestimmten Voraussetzungen auftreten, aufklären.

(a) (b)

Abbildung 2.5: (a) Beispiel für Simultankontrast bei Grauflächen. Quelle [Dah06]. (b) Beispiel für Simultankontrast bei Farbflächen. Quelle [Dah06]

Laterale Hemmung Laterale Hemmungen bezeichnen nichts Anderes als die „seitliche Verknüpfung von Sehzellen in der Netzhaut“ [Dah06]. Durch diese Eigenschaft können Farben und Kontrastwerte durch ihre Umgebung verfälscht werden. Ein Beispiel dafür ist der soge-

nannte Simultankontrast. Dabei erscheinen Graufl ächen in heller Umgebung dunkler als in dunkler Umgebung. In Abbildung 2.5 (a) tritt der Simultankontrast bei den inneren kleinen Rechtecken auf. Obwohl beide Rechtecke im gleichen Grauton dargestellt sind, erscheint das Linke dunkler als das Rechte. Bei Farbfl ächen wird durch die Umgebungsfarbe nicht nur die Helligkeit beeinflusst, sondern auch der Farbton. In Abbildung 2.5 (b) wird deutlich, dass obwohl beide blaue Rechtecke exakt dieselbe Farbe haben, das Linke in seiner Umgebung matt erscheint, w ährend das Rechte eher gl änzend wirkt. Dieses Problem tritt auch in dieser Arbeit auf, wenn n ämlich Curved Links als gewichtet Kanten über mehrere Fl üsse des Themerivers gezeichnet werden m üssen. Dies kann zu einer Misinterpretation der d ünn gezeichneten linienbasierten Kantendarstellung f ühren.

Ein weiteres Ph änomen, das durch laterale Hemmungen zu begr ünden ist, wird durch das sogenannte Hermann-Gitter (siehe Abbildung 2.6) deutlich. Hierbei erscheinen die Kreuzungen der hellen Streifen als dunkle Punkte, da sie von allen vier Seiten gehemmt werden.

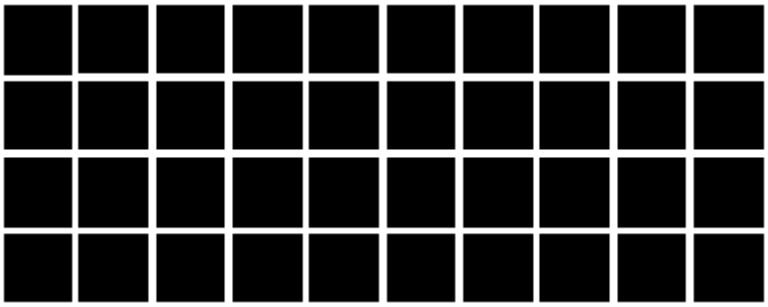


Abbildung 2.6: Hermann-Gitter. Quelle [Dah06]

Subjektive Farben Die Farbe in Bezug auf die Farbwahrnehmung meint in der Realit ät das Eintreffen von Licht einer bestimmten Wellenl änge auf der Netzhaut. Abbildung 2.7 zeigt die Empfindlichkeitsbereiche der unterschiedlichen Rezeptortypen (Zapfen). Demnach hat ein reines Rot eine Wellenl änge von circa 585 nm, was aber nicht unbedingt genau das gleiche Rot sein muss, was ein bestimmter Mensch als reines Rot empfindet, da die „Benennung und

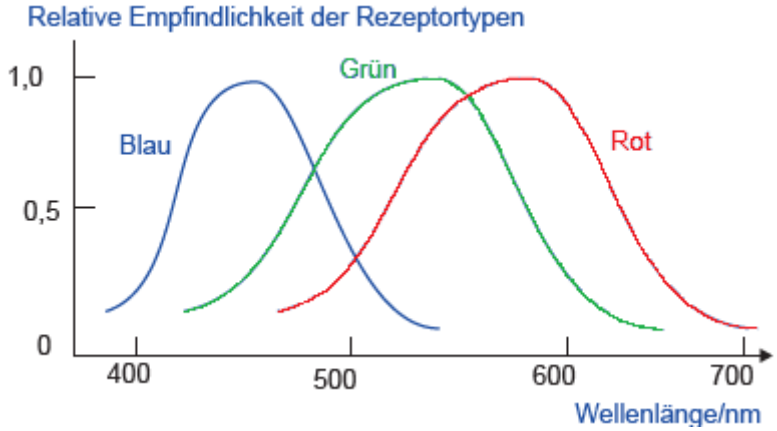


Abbildung 2.7: Empfindlichkeitsbereiche der Rezeptortypen auf der Netzhaut. Quelle [Dah06]

Beurteilung von Farben eine vollkommen subjektive Angelegenheit“ [Dah06] ist. Wieder sei hierbei auf das Blockschaltbild Abbildung 2.1 zur Wahrnehmung von visuellen Reizen ver-

wiesen. Zum Erkennen einer Farbe wird die gesehene Farbfläche mit den Erfahrungen und dem Wissen aus dem Gehirn beurteilt und somit das Rot letztendlich als Selbiges erkannt. Dadurch ist die Farbname-Farbwert Zuordnung nicht eindeutig, sondern variiert zwischen einzelnen Individuen. Dieses Phänomen kann eine besondere Rolle spielen, wenn Farbe eine besondere Wirkung erzielen soll (zum Beispiel Rot als Symbolfarbe für Liebe, Feuer, Gefahr).

Tiefeneindruck Der Computerbildschirm an sich hat lediglich die Möglichkeit, Objekte zweidimensional darzustellen. Um dennoch einen dreidimensionalen Eindruck eines Objekts oder Raumes zu kreieren, existieren folgende Kriterien, die einen Tiefeneindruck entstehen lassen.

Verdeckt ein Objekt ein anderes, so erscheint es dem Betrachter, als würde das verdeckte Objekt auch räumlich hinter dem Verdeckenden liegen (vgl. Abbildung 2.8 (a)). Weiterhin gilt,

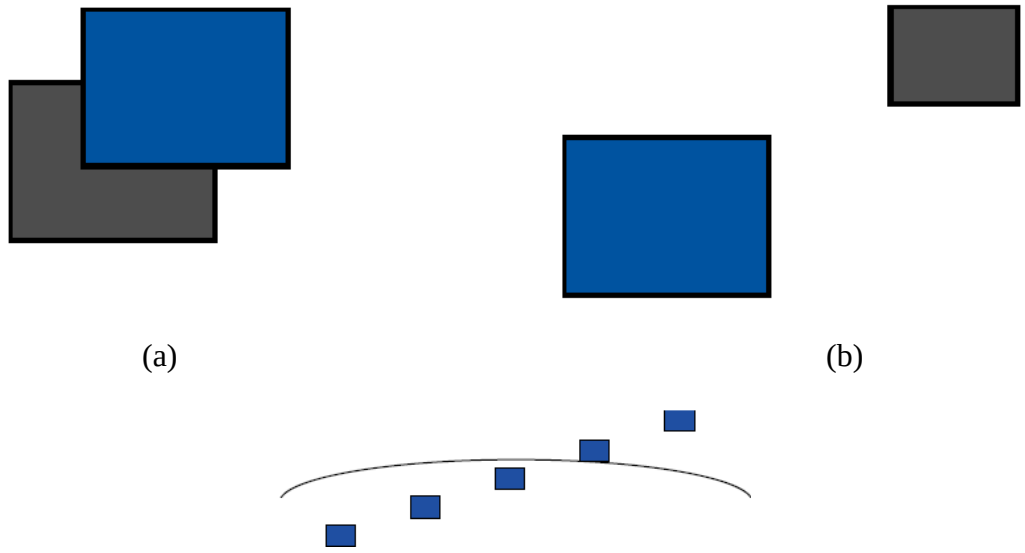


Abbildung 2.8: (a) Objekte im Vordergrund verdecken diejenigen im Hintergrund. Quelle [Dah06]. (b) Objekte im Vordergrund sind größer als Objekte im Hintergrund. Quelle [Dah06]. (c) Objekte im Vordergrund erscheinen niedriger als Objekte im Hintergrund. Die gekrümmte Linie stellt eine Art Horizont dar und bildet somit ein Bezugsobjekt. Quelle [Dah06]

dass Objekte im Vordergrund generell größer dargestellt sind, als Objekte im Hintergrund (vgl. Abbildung 2.8 (b)). Außerdem lässt sich über die Höhe der Objekte eine entsprechende Aussage treffen. Nahe Objekte sind niedriger dargestellt als entfernte Objekte. Dabei muss jedoch die Höhe der Objekte durch eine Bezugslinie oder Ähnliches verdeutlicht sein (vgl. Abbildung 2.8 (c)). Durch perspektivische Verzerrung und Beleuchtungs- und Schatteneffekte lassen sich, wie in Abbildung 2.9 ersichtlich, bereits sehr realistische Darstellungen erzielen. Abschließend sei an dieser Stelle noch die Tiefenschärfe als weiteres Kriterium, das einen Tiefeneindruck entstehen lässt, genannt. Sie stellt das fokussierte Objekt und alle in dieser Entfernungsebene liegenden Objekte scharf und nähere oder fernere Objekte unscharf dar.

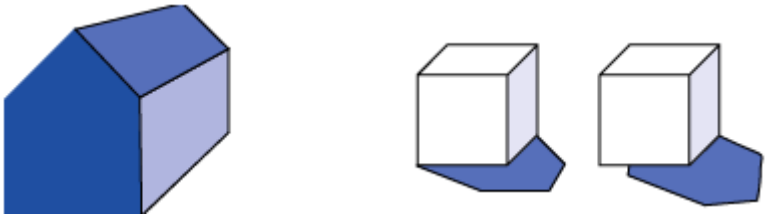


Abbildung 2.9: Realistische Abbildung durch perspektivische Verzerrung und Schattenwurf. Quelle [Dah06]

3. Verwandte Arbeiten

Auf dem Gebiet der Visualisierung von Graphen existieren bereits sehr viele Arbeiten. In diesem Kapitel werden Arbeiten, die sich mit ähnlichen Problemstellungen wie die Beobachtung und Vorhersage von Trends mit der Themeriver-Visualisierung beschäftigen, vorgestellt. Es ist verständlich, dass an dieser Stelle nur der Teil vorgestellt wird, der eng mit der TimeArcTrees-Technik verwandt ist.

3.1 Visualisierungen zeitabhängiger Daten

Visualisierungen von mehreren Zeitreihen wurden bereits vor vielen Jahren entwickelt. Wissenschaftler haben längst erkannt, dass Zeitreihendarstellungen trotz ihrer Einfachheit viele subtile Trade-Offs beinhalten. Graphen, die Zeitreihen mit gestapelten Schichten zeigen, reichen mindestens bis Playfair's Arbeit [Pla86] zurück. Erst vor kurzem sind jedoch Versionen entwickelt worden, die auch für größere Anzahlen von Zeitreihen skalieren.

Mit den zunehmenden Ansprüchen des Benutzers, komplexere Einsichten in die Daten zu bekommen, begannen Forscher zu erkunden, welche Maße in sozialen Medien verwendet werden können, um etwa auch aufkommende Trends in den Zeitserien zu erkennen und zu beobachten.

Im *Themeriver System* von Havre [HHNW02] wird die Visualisierung von gestapelten Graphen (stacked graphs) erstmals vorgestellt. In diesem System (vgl. Abbildung 4.2) repräsentieren die Schichten die Häufigkeit des Auftretens bestimmter Begriffe oder „Themes“ in großen Mengen von Textdokumenten über die Zeit. Die Zeitdimension wird auf die x-Achse abgebildet, wobei mehrfach auftretende zeitabhängige Daten auf die Breite der einzelnen Flusssegmente abgebildet werden. Zeit ist die fokussierte Dimension in zeitorientierten Daten und deshalb wird ausreichender Bildschirmplatz für die Darstellung der Zeitdimension in einer solchen Visualisierung benötigt (vgl. [AMSH11]). Unter den Innovationen in Themeriver gehören eine neuartige Technik zur Erzeugung einer glatten Interpolation aus diskreten Daten, und eine Layout-Methode, bei dem Schichten gestapelt wurden, nicht flach startend ab der x-Achse, sondern in einer symmetrischen Form mit der x-Achse lokalisiert in der Mitte. Hierbei werden die Schichten eines Graphen im Gegensatz zu konventionellen gestapelten Graphen nicht auf eine flache Grundlinie sondern auf eine anhand aller Schichten berechnete und entsprechend gekrümmte Grundlinie übereinandergelegt. Diese Art von Graph bietet dem Benutzer allerdings wenige interaktive Features und ist weniger gut geeignet für die Darstellung von semantischen Informationen. Allerdings überzeugt sie durch ihr ästhetisches Auftreten.

In [Wat05] wurde ein interaktives Werkzeug für geschichtete Graphen vorgestellt, die *NameVoyager*, die eine schnelle Erkundung von mehr als 6.000 Datensätzen auf einmal erlauben. Die Layout-Methode der *NameVoyager* war nicht neu. Hier wurde ein Standardlayout mit einigen Level-of-Detail Berechnungen verwendet. Ein Follow-up-Design des *NameVoyagers*, in [WK06] beschrieben, zeigte hierarchische Zeitreihen. Hierbei wurde die Interaktivität und Farbe verwendet, um Zeitreihen, die in Kategorien und Unterkategorien angeordnet darzustellen. Im *Many Eyes System* [VWVKM07] wurde diese Technik weitgehend im Web bereitgestellt.

Als eine kleine Abwandlung der Themeriver Methode hatte Lee Byron [BW08] eine neue Art von gestapelten Graphen für ein Projekt der New York Times im Februar 2008 entwickelt, die sehr viel positives Feedback erhielt. Ein inspirierender *StreamGraph* ist ein Graph, bei dem mehrere Datenschichten übereinandergelegt sind (vgl. Abbildung 3.1). Dies sieht nicht nur dynamisch, visuell ansprechende aus, sondern macht auch Informationen erkennbar, die anhand eines konventionellen Graphen schwer nachvollziehbar sind. Zusätzlich sind viele Interaktionsmöglichkeiten dem Benutzer geboten, den Graph zu manipulieren. Beispielsweise kann man mit Hilfe der Scrollbalken das Bild nach links bzw. nach rechts verschieben, damit interessante Regionen detailliert betrachtet werden können. Allerdings sind die interessanten Einsichten in die dynamischen Abhängigkeiten der sich zeitlich verändernden quantitativen Werte nicht möglich.

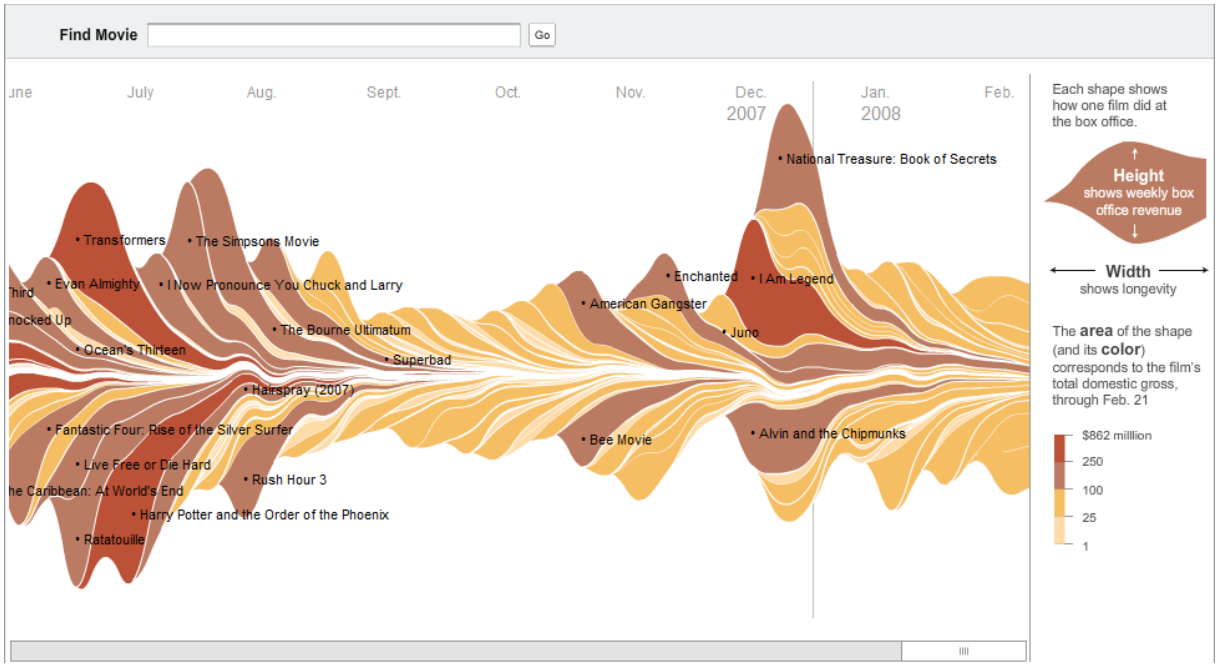


Abbildung 3.1: Abgebildet auf dieser wirklich innovativen Grafik (eine ansprechende Form des klassischen gestapelten Flächendiagramms) sind die Einspielergebnisse der erfolgreichsten Filme von 1986 bis 2008 in den USA. Die jeweilige Breite der Formen zeigt an, wie viel ein Film zu einem bestimmten Zeitpunkt eingespielt hat. Über die Zeit hinweg lässt sich also nachvollziehen, wie sich die Einspielergebnisse über die Zeit verändert haben - wie schnell der Film beim Publikum angekommen ist und wie lange er das Publikum begeistern konnte. Zugleich sieht man deutlich, dass es im Jahresverlauf zwei heiße Phasen des Kinobesuchs gibt: Sommer und Winter. Darüberhinaus ist die Grafik auch noch interaktiv: Falls man die einzelnen Formen mit dem Mauszeiger berührt, werden einige Detailinformationen angezeigt und man kann sich in der Regel zu der NYT-Filmkritik durchklicken. Das ist elegant und zeigt, wie man neue Technologien einsetzen kann, um die Data-Ink-Ratio zu verbessern.

Fukuhara [Fuk05], zum Beispiel präsentiert ein System *Kanshin*, das eine tägliche Trendgrafik von Weblog-Artikel mit einem beliebigen bestimmten Schlüsselwort generiert. Glance et al. [GHT04] stellen ein Tool namens *BlogPulse* vor, dass es ermöglicht, Trends in Weblogs

zu beobachten. Sie zeigen eine Korrelation zwischen „realem Welt Blog und temporalen Daten“ wie Temperatur- und News-Artikel. Hotho [HJSS06] präsentiert einen Ansatz namens *FolkRank* für die Entdeckung von Thema-spezifischen Trends innerhalb Folksonomies, durch Anpassung des PageRank-Algorithmus.

Eine weitere Verwandte Arbeit namens *TimeMines* [SJ00] ist ein automatisiertes System, das einen Überblick über die Zeitlinien von Themen in Textnachrichten erzeugt.

In den nächsten Abschnitten werden Arbeiten vorgestellt, die direkt mit der TimeArcTrees-Technik verwandt sind. Zuerst werden Techniken zur Visualisierung von statischen Graphen vorgestellt, dann die Arbeiten zur Darstellung von Folgen von Graphen, d.h. dynamischen Graphen.

3.2 Visualisierung statischer Graphen

Laut einer Studie [Kel06] sind graphische (Knoten-Kanten-Diagramm) und matrixbasierte Visualisierungen für typische Visualisierungsaufgaben grundlegend geeignet. Beide Visualisierungsmethoden können sowohl die Strukturelemente als auch die Relationen zwischen den Elementen im ausreichenden Maße darstellen. Die graphische Methode kommt besonders bei Strukturen mit einer kleinen Menge von Elementen (weniger Kantenkreuzungen) zum Einsatz, während das Visual Clutter Problem [RM05] mit matrixbasierten Methoden leichter reduziert werden kann.

Ein Ansatz, der für Hierarchien flächenfüllende Visualisierungen verwendet und für die Relationen zwischen den Hierarchieelementen Kanten darüber zeichnet, ist der *ZTree* [BUC00]. Hierbei werden die Elemente der Hierarchie als Fenster dargestellt. Falls nun Relationen zwischen zwei Fenstern oder deren untergeordneten Elementen bestehen, wird dies durch eine Kante angezeigt. Der Benutzer hat durch Anklicken der Fenster die Möglichkeit, tiefer in die Hierarchie hineinzuschauen. So hat er stets alle Relationen zwischen den aufgeklappten Fenstern im Blick. Die Visualisierung erfolgt nach einem speziellen Layout-Algorithmus, der dem Benutzter beim Navigieren durch große Hierarchien helfen soll.

Einen ähnlichen Ansatz findet man in *Overlaying Graph Links on TreeMaps* [FWDAP03]. Hier wird die Hierarchie als traditionelle *TreeMap* dargestellt und die Kanten verlaufen jeweils zwischen den Mittelpunkten der Flächen. Die Kanten werden als Bezier-Kurven gezeichnet und deuten durch ihre Krümmung die Laufrichtung an. So können auch gerichtete Graphen dargestellt werden. Durch die Anordnung der Elemente als *TreeMap* kann man auch bei großen Datensätzen die Teilhierarchien erkennen, zwischen denen viele Relationen bestehen. Zwischen diesen Bereichen der *TreeMap* verlaufen besonders viele Kurven. Zusätzlich hat der Benutzer die Wahl, sich entweder alle Linien anzeigen zu lassen oder nur diejenigen von bzw. zu einem Element der TreeMap. Diese Funktion hilft dem Benutzter, wenn er sich nur für bestimmte Knoten und deren Beziehungen interessiert. In der Arbeit „Trees in a Treemap“ von Burch und Diehl [BD06] werden zusätzlich Präfixbäume als Knoten-Kanten Diagramme in eine bestehende Treemap gezeichnet.

Ähnlich zum TimeArcTrees-Ansatz (siehe Kapitel 4.3) werden in *ArcTrees* [NSC05] alle Elemente auf einer Geraden angeordnet und Beziehungen als Kreisbögen um die Gerade herum gezeichnet. Allerdings sind nur ungerichtete Graphen darstellbar und alle Kanten schwin-

den sich auf einer Seite. Die Hierarchie wird wie bei der traditionellen *TreeMap* mithilfe ineinander verschachtelter Rechtecke dargestellt. Eine Aggregation über die Hierarchie ist möglich, indem man entweder Kanten oder Rechtecke anklickt. Der Grad der Aggregation einer Kante wird durch Transparenz angedeutet und die Breite der Kante repräsentiert das Gesamtgewicht.

3.3 Dynamische Graphvisualisierung

Dynamische Graphen verändern ihre Struktur im Laufe der Zeit. Es ist sehr schwierig für konventionelle Graphvisualisierungstechniken, gleichzeitig auch für dynamische Graphen geeignet zu sein. In diesem Abschnitt diskutieren wir einige verwandte Arbeiten, die dieses Problem umgehen. Wenn verwandte Werkzeuge vorgestellt werden, können *TimeRadarTrees* [BD08] und *Timeline Trees* [BBD08] nicht ausgelassen werden, da sie die Idee für die Entwicklung von TimeArcTrees geliefert haben. Alle drei Werkzeuge beschäftigen sich mit Folgen von Graphen oder Transaktionen, deren Knoten einer Informationshierarchie unterliegen. Allerdings werden die Daten unterschiedlich visualisiert.

Das TimeRadarTrees-Werkzeug (siehe Abbildung 3.2 (a)) verwendet ein radiales, kreisförmiges Baumlayout zur Darstellung der Hierarchie und Kreissektoren repräsentieren die zeitlichen Änderungen des Graphen. Die Beziehungen zwischen den Elementen der Hierarchie werden nicht explizit als Kanten dargestellt. Alle eingehenden Kanten eines Knotens werden aufsummiert und als gefärbter Kreissektor dargestellt. Um herauszufinden, ob zwei Elemente in Beziehung stehen, werden rund um den gesamten Kreis an jedem Sektor kleine äußere Kreise (Thumbnails) gezeichnet, die nur die ausgehenden Kanten des Knotens als Kreissektoren zeigen. Der Zeitverlauf wird im Kreis von innen nach außen dargestellt.

Das Werkzeug Timeline Trees (siehe Abbildung 3.2 (b)) visualisiert Folgen von Transaktionen zwischen Elementen einer Informationshierarchie. Die Hierarchie wird wie beim TimeArcTrees-Werkzeug auf der linken Seite des Bildschirms als Baum von links nach rechts gezeichnet. Alle aktuellen Blätter besitzen rechts der Hierarchie eine Zeitleiste, welche die Abfolge der Transaktionen visualisiert.

In der Zeitleiste werden die Transaktionen als farbige Rechtecke dargestellt. Die Farben und Größen der Rechtecke zeigen, wie stark ein Element an der Transaktion beteiligt ist.

Zusätzlich werden zu jedem Element Miniaturbilder angezeigt, welche die Zeitleiste so darstellen, dass nur Transaktionen, an denen das Element beteiligt ist, farbig gezeichnet werden.

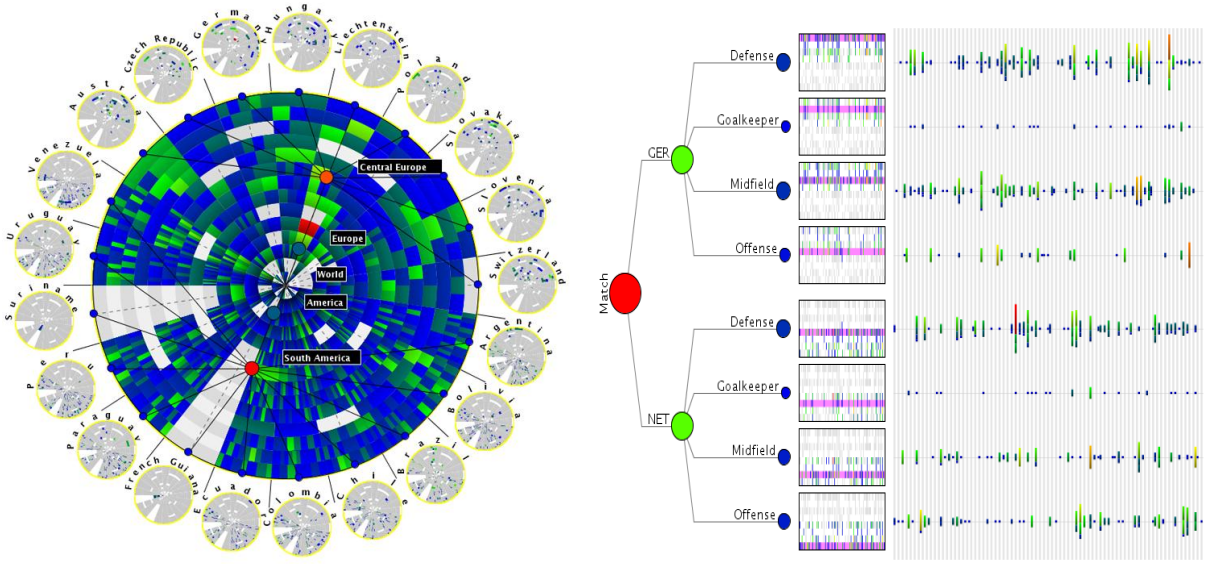
Der Benutzer kann die Evolution der Transaktionen und die Rollen ihrer Mitglieddelemente analysieren, und anschließend erkennen, wann und wie stark die Elemente der Hierarchie verknüpft sind.

Das Werkzeug TimeArcTrees (siehe Abbildung 3.3 (a)) verwendet einen statischen Ansatz zur Visualisierung dynamischer Compound-Digraphen. Sie zeigen eine Folge von Knoten-Kanten-Diagrammen mit horizontaler Knotenausrichtung in einer einzigen Ansicht, wodurch ihr direkter Vergleich unterstützt wird.

Die Knoten der einzelnen Graphen der Folge werden auf einer Vertikalen von oben nach unten gezeichnet. Die Reihenfolge wird von der Informationshierarchie bestimmt, die auf der

linken Seite der Ansicht als Knoten-Kanten-Diagramm abgebildet wird. Die Kanten in den Graphen der Folge verlaufen links bzw. rechts der Vertikalen. Aufwärts laufende Kanten befinden sich links und abwärts laufende rechts der Geraden. Diese Anordnung soll beim visuellen Verfolgen von Kanten helfen und die Graphen möglichst strukturiert darstellen.

Eine weitere Verwandte Arbeit *Parallel Edge Splatting* [BVBDV11] zur Repräsentation eines dynamischen Graphen verwendet eine neuartige Visualisierungstechnik auf Knoten-Kanten-Diagrammbasis (siehe Abbildung 3.3 (b)). Das Diagramm wird nebeneinander von links nach rechts als Reihenfolge von schmalen Streifen gezeichnet, die orthogonal zur horizontalen Zeitachse angeordnet sind. Graphknoten richten sich so aus, dass sie auf einer vertikalen Achse hierarchisch angeordnet sind. Jede Achse repräsentiert einen Zeitpunkt, wobei gerichtete Kanten zwischen zwei benachbarten parallelen Achsen in Relation stehende Knoten darstellen. Das Visual Clutter Problem aufgrund einer großen Menge an Kantenkreuzungen im Layout eines Knoten-Kanten-Diagramms wird hier mit Hilfe des *Edge Splatting* Ansatzes (Kanten werden in einem Dichtefeld repräsentiert) reduziert. Das Dichtefeld wird durch die Farbkodierung anhand des Kantengewichtes so dargestellt, dass die Kantenverlauf in ganz unübersichtlichen Bereich noch erkennbar wird.



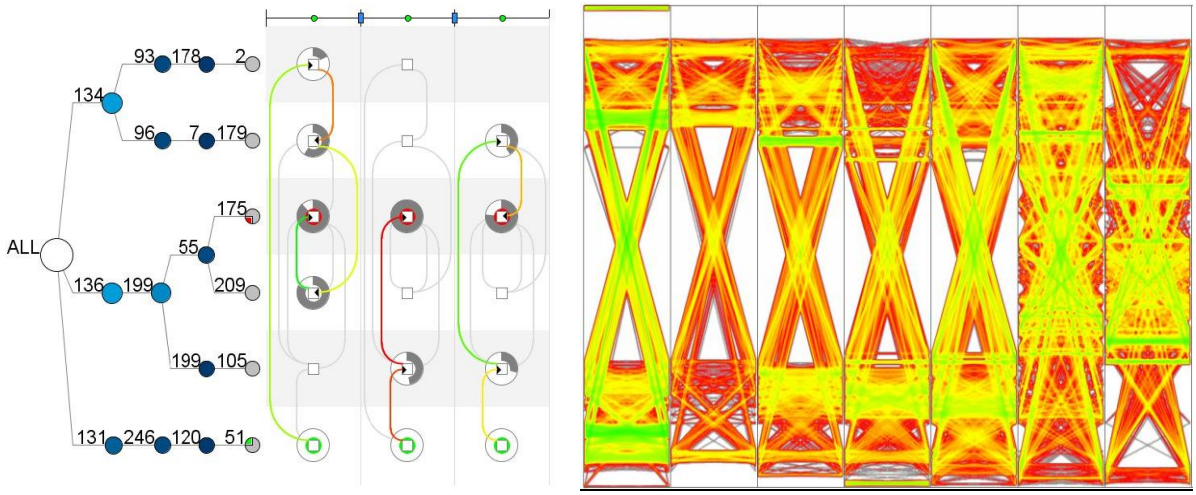
(a) (b)

Abbildung 3.2: (a) TimeRadarTrees-Visualisierung: Vergleich der Fußballspielergebnisse zwischen den nationalen Fußballmannschaften in Mitteleuropa und Südamerika der 14 Jahre von 1992 bis 2005 [BD08]. (b) Timeline Trees Visualisierung der Ballkontakte in einem Fußballspiels [BBD08].

3.4 Interaktion

Die wachsende Flut der Informationen aus unterschiedlichsten Quellen, wie z.B. soziale Netzwerke, Medizindatenbanken usw. stellen immer härtere Anforderungen an Techniken zur Interaktion mit solchen Daten. Die Interaktion zwischen Analytiker und Daten spielt dabei eine sehr wichtige Rolle bei der Informationsvisualisierung. Sie unterstützt den Analytiker auch bei der Analyse großer, zeitbasierter Datensätze. In diesem Abschnitt wird die Interakti-

onsfähigkeit der in Kapitel 3.3 vorgestellten Visualisierungstechnik in Bezug auf die vorliegende Arbeit verglichen.



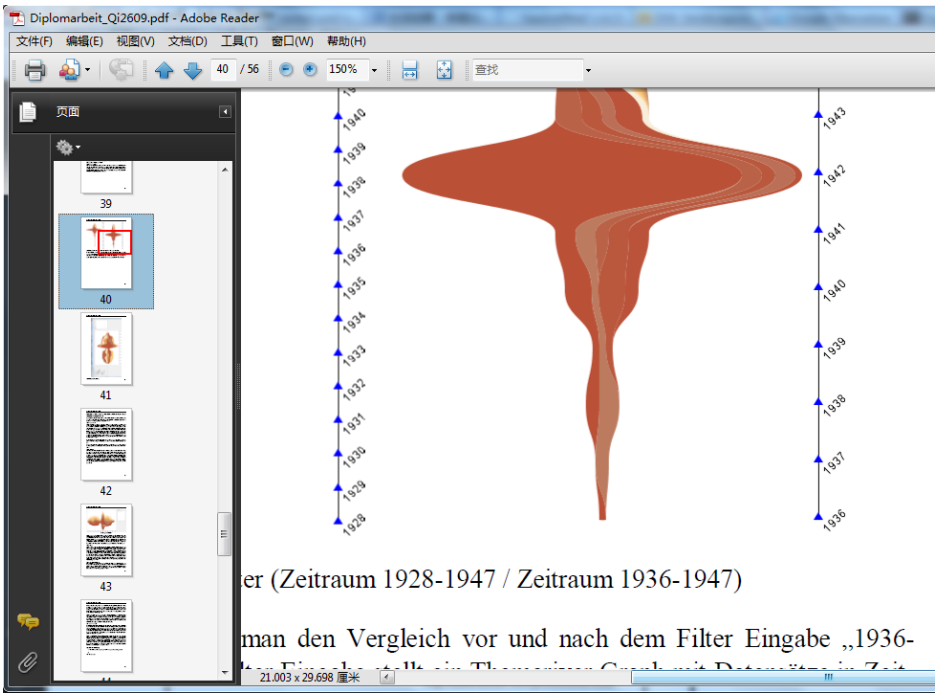
(a) (b)

Abbildung 3.3: (a) Knoten-Kanten-Diagramm eines gerichteten Graphen in einer gewichteten TimeArcTrees-Darstellung. Die gewichteten Kanten werden durch farbige Bögen dargestellt [GBD09]. (b) Parallel Edge Splatting Visualisierung [BVBDV11].

Laut [Shn96] unterstützt das Visual Information Seeking Mantra die Analyse der Daten. Dieses besteht in folgenden drei Schritten: „Overview first, zoom and filter, then details-on-demand“. Der Benutzer soll sich zunächst eine Übersicht über den ganzen Datensatz beibringen, um interessante Einsichten zu bekommen (Overview). Durch Zoom- und Auswahltechniken können die genaueren Untersuchungen im Bereich seines Interesses ausführlicher dargestellt werden (Zoom and filter). Schließlich kann der Benutzer bei Bedarf in dem ausgewählten Betrachtungsbereich die Ausschnitte der Daten abfragen (Details-on-demand), damit zusätzliche Informationen herausgefunden werden und z.B. in textueller Form anzeigen werden können. Allgegenwärtige Interaktionstechnik wie etwa „Brushing and Linking“ (siehe Abbildung 3.4), „Focus + Context“ (siehe Abbildung 3.5) und Aggregation sind auch in der Graphvisualisierung anwendbar.

Das Werkzeug TimeRadarTrees beinhaltet viele interaktive Funktionen zur Erforschung der Datensätze. Es ist auch möglich, Teilbäume der Informationshierarchie zu einem Knoten zusammenzufassen oder einzelne Graphen der Folge herauszufiltern. Der große Vorteil von TimeRadarTrees ist, dass die Visualisierung nicht durch viele Kanten und damit auch durch viele Kantenkreuzungen überladen wird. Die einzigen Kanten, die gezeichnet werden, sind die der Hierarchie, welche kreuzungsfrei angeordnet werden können.

Timeline Trees hilft Benutzern mit mehreren Interaktionsmöglichkeiten, Trends in den Daten zu verstehen und zusätzlich die Relationen auf den verschiedenen Abstraktionsebenen. Beispielsweise lassen sich einzelne Teilbäume der Hierarchie beliebig auf- und zuklappen.



er (Zeitraum 1928-1947 / Zeitraum 1936-1947)

man den Vergleich vor und nach dem Filter Eingabe „1936-

Abbildung 3.4: „Brushing and Linking“ am Beispiel des Acrobat Reader-Tools. Brushing (Einfärbung): Der Hintergrund der ausgewählten Seite ist mit blau gefärbt, zusätzlich roter aktueller Ansichtsbereich. Linking (Verknüpfung): zusätzliche werden die Auswahl betreffende Informationen angezeigt.

TimeArcTrees ermöglicht dem Benutzer ebenfalls, mit vielen interaktiven Features dynamische Graphen zu manipulieren. Der Benutzer kann das Gewicht der Kanten je nach Datensatz und Anwendung einstellen. Optionen zur Veränderung der Farbskala werden auch angeboten. Ein Filter existiert ebenfalls im Werkzeug, damit der Wertebereich der Gewichte eingeschränkt werden kann. Es ist mit der eingebauten Zoomfunktion auch möglich, die Ballungsstellen des Graphen genauer zu betrachten. Der Zoomausschnitt wird in doppelter Größe dargestellt. Ein horizontaler Streifen auf der Höhe des Mauszeigers wird über die gesamte Breite des Fensters gestreckt gezeichnet, damit auch während der Benutzung die Zuordnung der Repräsentanten zu den korrespondierenden Hierarchieknoten zu erkennen ist. Der Ausschnitt in dem Intervall unter dem Mauszeiger wird besonders genau dargestellt, indem auch in die Breite gestreckt wird.

Die Visualisierungsmethode Parallel Edge Splatting unterstützt die Datenvisualisierung ebenfalls durch viele Interaktionstechniken: Brushing und lingking, Aggregation, Filterung und Zooming, um nur einige zu nennen.

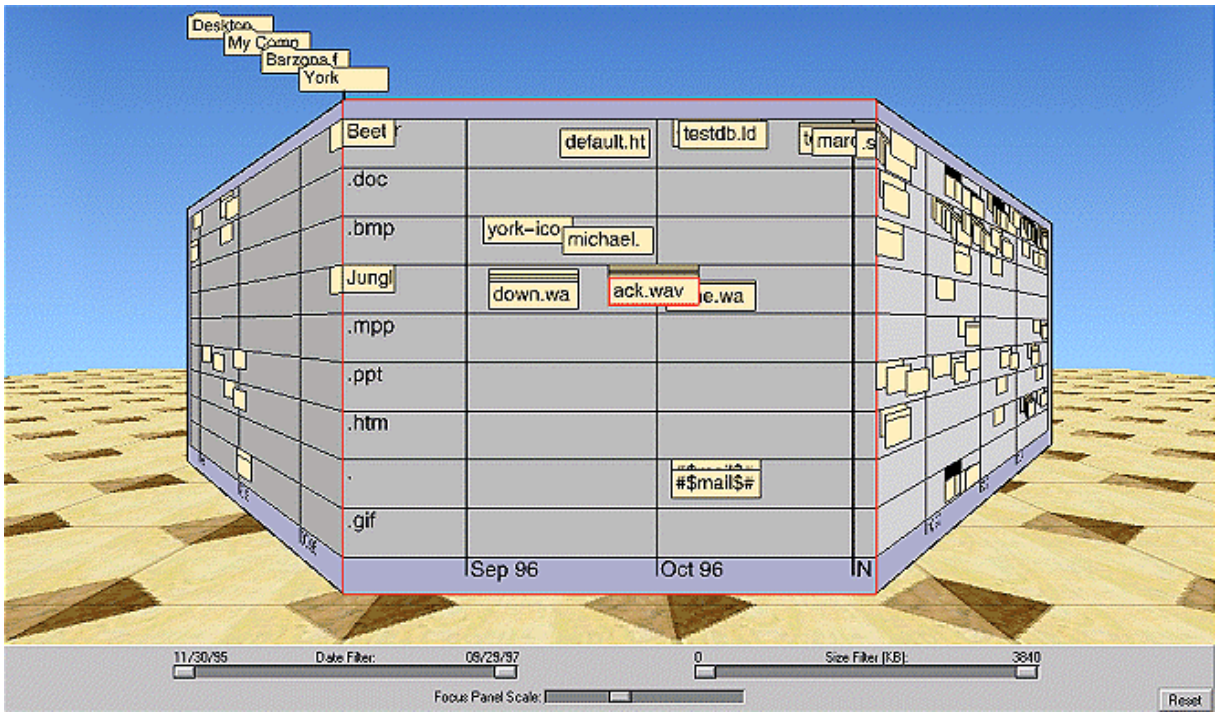


Abbildung 3.5: Perspective Wall Repräsentation mit Interaktionstechnik „Focus + Context“. Die zeitbasierte Daten werden auf eine 3D perspektivische Wand abgebildet, wobei die Zeit als horizontale x-Achse dargestellt wird. Die Bereiche neben dem vom Benutzer ausgewählten Fokus werden nach hinten weggeklappt und wirken perspektivisch verzerrt. Durch die Bestätigung vom Benutzer per Mauszeiger verschiebt sich das Objekt in den Vordergrund, während die verbleibenden Informationen entsprechend vorrücken. Quelle vgl. [AMSH11]

4. Visualisierungstechnik

Im letzten Kapitel wurden einige verschiedene Visualisierungstechniken vorgestellt. Allerdings zeigen sie mit unterschiedlichen Eigenschaften dementsprechend verschiedene Vor- und Nachteile. Gerade aufgrund ihrer Unterschiede lohnt es sich, die verschiedenen Techniken integrierend einsetzen zu können. Aus diesen Überlegungen resultiert die Idee, ein Visualisierungs- und Analysewerkzeug prototypisch zu entwickeln, welches in der Lage ist, die Themeriver Visualisierung interaktiv zu erweitern. Vor der eigentlichen Implementierung eines Visualisierungswerkzeuges ist es notwendig, sich Gedanken über den grundsätzlichen Aufbau und die einzelnen Komponenten zu machen. Im Folgenden werde ich auf eben diese Fragen eingehen und dabei die Grundlagen für die in Kapitel 5 beschriebene Implementierung schaffen.

Das Hauptziel der Visualisierung ist die Abbildung von Daten auf geometrische Primitive und deren Attribute (zum Beispiel Form, Farbe usw.). Dies soll nach [SM00] so geschehen, dass die Darstellung

- expressiv,
- effektiv und
- angemessen

ist. Von einer expressiven Darstellung spricht man, wenn die in den Daten enthaltenen Informationen - und nur diese - angezeigt werden. Werden die Eigenschaften des menschlichen visuellen Systems optimal angesprochen, so dass ein Bild schnell und intuitiv interpretierbar ist, so gilt das Effektivitätskriterium als erfüllt. Angemessen ist eine Visualisierung, wenn Nutzen und Aufwand bei der Erstellung des Bildes in einem ausgewogenen Verhältnis stehen.

Folgt man Daassi [DFN02] und Aigner [Aig06], so weisen zeitorientierte Informationen stets datenspezifische und zeitliche Aspekte auf, die bei der Visualisierung der Daten berücksichtigt werden müssen (siehe Abbildung 4.1). Für die Erzeugung von Bildern aus Daten sind mehrere Schritte notwendig, die nacheinander durchlaufen werden. Die so bezeichnete Visualisierungs-Pipeline besteht aus den Prozessen Filtering, Mapping und Rendering. Das heißt, dass zunächst die Daten gefiltert werden (zum Beispiel, um Nullwerte zu eliminieren oder die Datenmenge einzuschränken), danach auf geometrische Primitive und deren Attribute abgebildet werden und anschließend durch den Rendering-Schritt in Bilddaten überführt werden. Hierbei hat das Mapping den weitaus größten Einfluss auf die Expressivität, Effektivität und Angemessenheit der Darstellung.

Visualisierungstechniken können einfach einen Überblick über die Daten erzeugen und erlauben es einem Benutzer, interessante Teilmengen innerhalb der Visualisierung schnell zu erkennen. Während des Fokussierens auf interessante Teilmengen ist es wichtig, einen Überblick über die Daten beizubehalten, was zum Beispiel durch eine interaktive Verzerrung der visuellen Überblicksdarstellung bezüglich des Fokus' erfolgen kann. Für die weitere Exploration interessanter Teilmengen benötigt der Datenanalytiker eine Möglichkeit, die Daten genauer zu betrachten, um Details nachzuvollziehen.

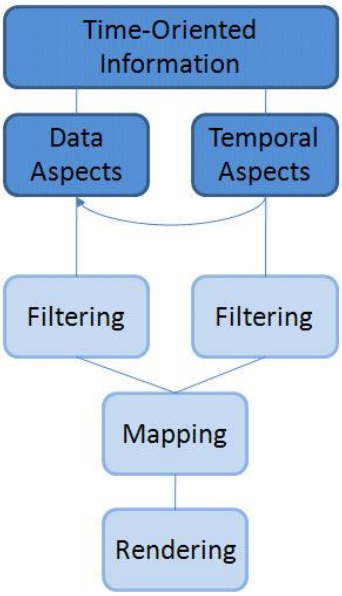


Abbildung 4.1: Visualisierungspipeline für zeitbezogene Informationen nach Aigner [Aig06] und Daassi [DFN02]

Für die Visualisierung von Daten existiert eine Vielzahl von Visualisierungstechniken. Neben den weitverbreiteten Standard 2D/3D Techniken, wie zum Beispiel x-y (bzw. x-y-z) Diagrammen, Balkendiagrammen, Liniendiagrammen usw., stehen heute eine Reihe weiterentwickelter Visualisierungstechniken zur Verfügung. Diese Techniken können für die visuelle Datenexploration hilfreich sein, jedoch sind sie im Allgemeinen beschränkt auf relativ kleine und niedrigdimensionale Datenmengen. In den vergangenen Jahren wurden eine Vielzahl neuartiger Techniken für hochdimensionale Datenmengen ohne interne 2D- oder 3D-Semantik entwickelt. Im Folgenden werden die Eigenschaften des vom Autor entwickelten Werkzeugs vorgestellt.

4.1 Datenmodell

Hier wird eine Einführung in das Datenmodell gegeben, mit dem das Werkzeug arbeitet und aus dem das Eingabeformat abgeleitet wird. Für die Graphdaten benötigt man ein Datenmodell, das alle darzustellenden Eigenschaften beinhaltet. Die Menge der Knoten V des Graphen G repräsentiert die darzustellenden Elemente der Visualisierung und die Menge der Kanten E die Relationen zwischen den Elementen.

$$G = (V, E)$$

Da es sich im Digraphen um gerichtete Kanten handelt, also E ⊆ V × V, spielt die Reihenfolge der Elemente beim Modellieren der Relation eine wichtige Rolle. Häufig besitzen die Beziehungen zwischen den Elementen ein Gewicht, welches zugeordnet zur Kante durch einen numerischen Wert angegeben werden kann. Weiterhin werden die Knoten der einzelnen Graphen der Folge jeweils mit entsprechender Häufigkeit zugeordnet. Die Menge der Knoten V wird als eine Folge V von n Knoten modelliert:

$$\mathrm{V} = \{v_1, v_2, \ldots, v_n\}$$

Aus allen bisher erwähnten Eigenschaften kann bereits ein einzelner Graph modelliert werden. Die wichtigste Eigenschaft des Datenmodells besteht nun darin, Folgen von Graphen zu beschreiben. Es wird hierfür jedem Graphen ein Zeitstempel t mit Funktion f zugeordnet. So kann der Verlauf der Beziehungen zwischen den Elementen innerhalb einer Zeitspanne modelliert werden, um daraus Informationen herzuleiten:

$$f: V \times \mathbb{N} \rightarrow \mathbb{R}_0^+$$

$$V = Knotenmenge$$

$$\mathbb{N} = Zeitschritte$$

$$\mathbb{R}_0^+ = Gewicht\ zweier\ Kanten$$

Im Folgenden werden zuerst die Eingangsdaten im Format des Werkzeugs, zweitens die internen Datenstrukturen und schließlich die verschiedenen möglichen Ausgaben des Werkzeugs beschrieben.

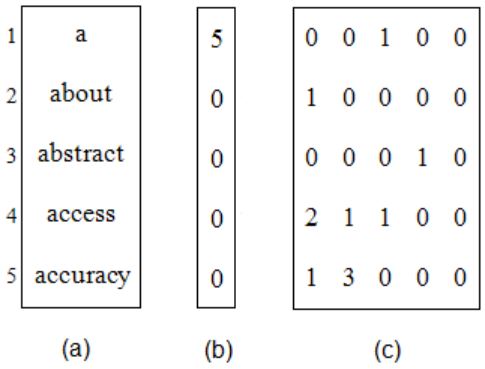
Basierend auf diesem Datenmodell kann die Form der Eingabedateien erzeugt werden. Das Werkzeug akzeptiert ein Dateiformat, das wir als Dateninformationen mit der Dateiendung .txt speichern. Dieses Format liegt in Multi-Spalten Form vor, das leicht mit Tabellen oder anderen Datenbank-Output-Mechanismen erstellt werden kann. Wir machen hier Gebrauch von diesem Format, weil es weitverbreitet ist.

Die Eingangsdaten bestehen aus mehreren Informationen, d.h. Wörter, Zeitstempel, Häufigkeiten und Relationen. Daher werden drei Arten von Dateien zum Einsatz kommen. Die erste Eingabedatei enthält die Namensinformationen und hat folgende Form: Sie besteht aus einer einzigen Spalte und jede Zeile beschreibt den Namen eines Themas. Die zweite enthält die Häufigkeitsdaten und hat die gleiche Form wie die erste: Jede Zeile beschreibt hier die Häufigkeit, in der das Thema auftritt. Die letzte enthält die Relationsdaten und hat folgende Form: Sie besteht aus einer quadratischen Matrix, deren Element die Relationsinformationen zwischen verschiedenen Themen beschreibt. Die Spalten sind durch Tabs getrennt (unsichtbar). Die Dateien aus den letzten zwei Gruppen beinhalten die Zeitinformationen, damit die Häufigkeits- und Relationsdaten mit den Zeitstempeln verknüpft werden können. Die Anzahl der Knoten, Kanten und Graphen ist im Datenmodell unbeschränkt. So können theoretisch auch sehr große Datenmengen beschrieben werden.

Ein Beispiel für die Eingabedateien sieht folgendermaßen in der Liste 4.1 aus.

4.2 Themeriver

Die im Bereich der Informationsvisualisierung vorkommenden Daten besitzen in der Regel eine große Anzahl an Variablen. Jeder Datensatz entspricht dabei einer Beobachtung, wie zum Beispiel einer Messung bei einem physikalischen Experiment oder einer Transaktion in einem E-Commerce System, und besitzt eine feste Anzahl an Attributen. Im Bereich der Informationsvisualisierung spricht man in der Regel von Dimensionen.



Liste 4.1: In (a), (b) und (c) werden Dateien mit den Namen „dblp.xml.words“, „dblp.xml.words.1936“, „dblp.xml.words.relation.1936“ dargestellt. Da die Dateien mit den entsprechenden Namen versehen sind, liest das Werkzeug die Daten aus den Dateien und speichert sie entsprechend ins Modell.

Eindimensionale Daten besitzen in der Regel ein kontinuierliches Attribut, das eine vollständige Ordnung auf den Daten definiert. Das bedeutet jedoch nicht, dass sie nur ein einziges Attribut besitzen. So könnte z.B. eine Liste mit Städten durch zahlreiche Attribute wie Anzahl der Krankenhäuser, Defizite im kommunalen Haushalt oder Einwohnerzahl beschrieben werden. Ist diese Liste aber sequentiell alphabetisch geordnet, ist nur das Attribut „Anfangsbuchstabe“ von Interesse. „Frankfurt“ käme z.B. vor „Stuttgart“, auch wenn die Einwohnerzahl von Frankfurt die von Stuttgart übersteigt. Beispiele für eindimensionale Datensätze sind Textdokumente, Programmquellcodes und alphabetische Namenslisten. Häufige Nutzeraufgaben sind zum Beispiel, die Anzahl der Objekte herauszufinden oder Objekte mit speziellen Attributen (wie alle Zeilen eines Dokuments, die Überschriften darstellen, alle Leute aus einer Liste, die älterer als 67 Jahre sind). Als Spezialfall von eindimensionalen Daten werden dynamische Daten angesehen. Es handelt sich dabei um zeitabhängige Daten, bei denen jedem Zeitpunkt mehrere Datenwerte zugeordnet werden können. Ein Beispiel wäre die zeitliche Entwicklung von Zeitungsmeldungen (vgl. Themeriver Visualisierung [HHNW02] in Abbildung 4.2).

Die Hauptaufgabe bei der Visualisierung von Dokumenten in Textform besteht darin, Informationen aus einem Dokument oder einer Dokumentgruppe zu gewinnen, ohne die eigentlichen Texte lesen zu müssen. Das kann z.B. bedeuten, dass verschiedene Themenschwerpunkte veranschaulicht werden oder dass Zusammenhänge zwischen dem Auftreten von bestimmten Begriffen herausgearbeitet werden.

Mit den thematischen Schwerpunkten in Ansammlungen von Dokumenten befasst sich die Technik Themeriver. Der Themeriver ist eine Visualisierungstechnik für Dokumente, die Veränderungen des thematischen Schwerpunkts innerhalb einer großen Menge an Dokumenten visualisiert und helfen soll, Trends, Muster, unerwartetes Auftreten bzw. Nichtauftreten von Themen oder bestimmten Daten zu identifizieren. Die Themenschwerpunkte werden farblich voneinander abgegrenzt und bilden die visuellen Flüsse der Datenwerte entlang der Zeitachse, wobei die Breite einer Schicht proportional zur Bedeutung des Themas ist. Ein wichtiges Merkmal des Themerivers ist der von links nach rechts fließende kontinuierliche Fluss, der dadurch entsteht, dass zwischen den einzelnen untersuchten Zeitpunkten interpoliert wird.

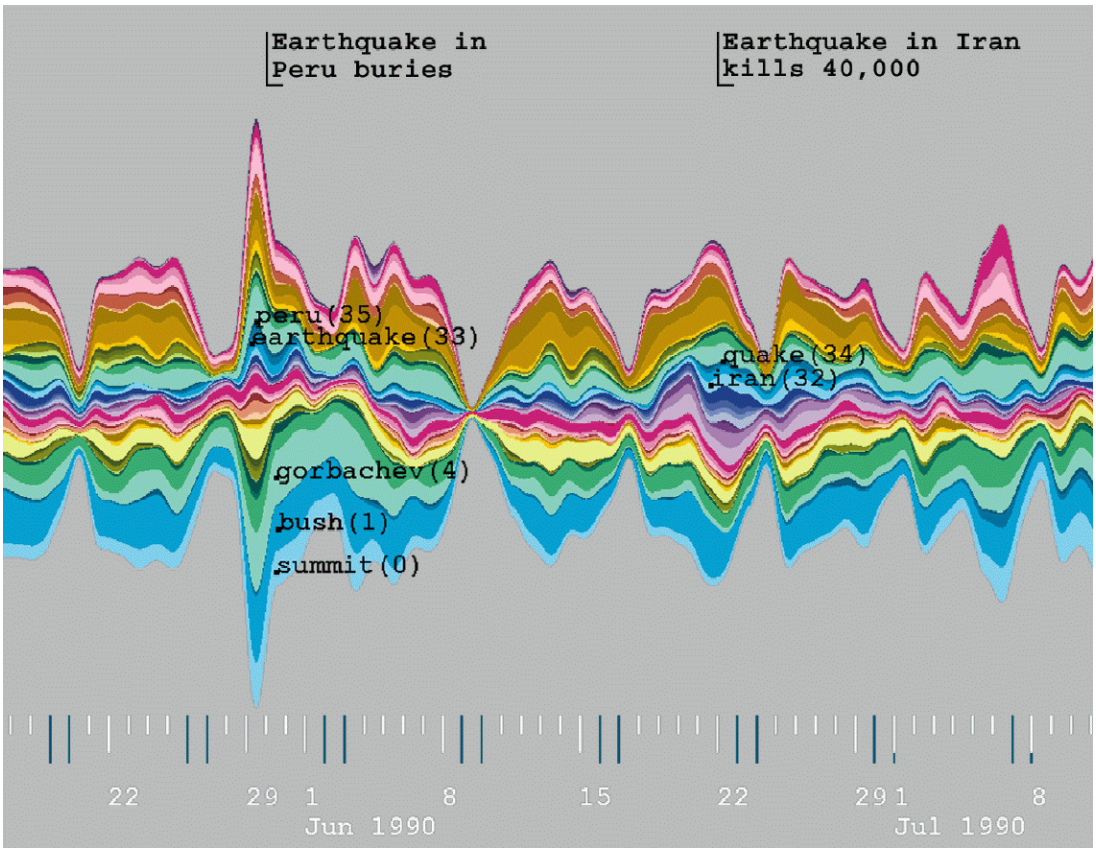


Abbildung 4.2: Die Themeriver Visualisierungstechnik stellt die zeitlichen thematischen Veränderungen dar, die durch sich verändernde Breiten der einzelnen Flusssegmente visualisiert werden. Quelle [HHNW02]

Dies kann z.B. linear oder durch kubische Splines erfolgen. Je schmaler die Farbschicht an einer bestimmten Stelle ist, desto weniger vorherrschend war das Thema zu dem entsprechenden Zeitpunkt in den untersuchten Dokumenten.

In unserem Beispiel wird ein Archiv von Associate Press Nachrichtenmeldungen von Juni bis Juli 1990 visualisiert. Die wichtigsten Ereignisse dieses Zeitraumes, wie das Erdbeben in Peru und das Gipfeltreffen zwischen Bush und Gorbatschow, können dabei gut identifiziert werden. Der Themeriver stellt im Prinzip eine Visualisierung mehrerer quantitativer Werte über eine weitere Dimension (ursprünglich die Zeit) dar. Allgemeiner lässt sich sagen, dass die einzelnen Flüsse untereinander Werte vergleichbarer Domänen repräsentieren müssen, damit die Vergleichsmöglichkeit gegeben bleibt. Der Anwender hat die Möglichkeit, das gewünschte Design hinsichtlich einer Einbeziehung von zusätzlichen Informationen, wie zum Beispiel besonders geschichtsträchtigen Ereignissen, als Textvermerke zu beeinflussen. Auch Interaktion ist beim Themeriver bis zu einem gewissen Grad gewährleistet. Bezüglich der Zeit ist Zooming möglich, so dass man entweder einen größeren zeitlichen Kontext oder eine detailliertere Darstellung betrachten kann.

Die übersichtliche und intuitive Darstellung, die in der Themeriver Technik verwendet wird, macht sie gut geeignet für Benutzer, die kein Spezialwissen besitzen. Der Fluss ist leicht zu überblicken und die Anzahl der Dimensionen ist beschränkt auf die Zeit und die Häufigkeit des Auftretens eines Themas. Der Zusammenhang zwischen Schichtdicke und Wichtigkeit

des Themas entspricht auch unserem intuitiven Verständnis und ist somit nicht schwer zu verstehen. Diese relativ einfache und leicht verständliche Darstellung ermöglicht auch Laien einen einfachen Zugang.

Allerdings gehen bei der Themeriver Standarddarstellung typischerweise die relationalen Informationen zwischen Objekten verloren. Aus diesem Grund muss ein dynamischer gerichteter und gewichteter Graph zusätzlich in die Standarddarstellung integriert werden.

4.3 Dynamische Relationen

Oft ändern sich die Relationen von Zeitpunkt zu Zeitpunkt. Die Graphen sind dynamisch und können durch eine Folge von einzelnen Graphen dargestellt werden. Jede Änderung der Struktur des Graphen kann durch einen separaten Graphen in der Folge modelliert werden. Zusätzlich unterliegen die Objekte oft einer hierarchischen Ordnung. Diese Ordnung wird Informationshierarchie genannt und kann gemeinsam mit den zusätzlichen Relationen zwischen den hierarchischen Objekten als Compound Graph dargestellt werden. Informationshierarchien sind in vielen Anwendungsbereichen zu finden. Beispiele sind hierarchische Strukturen eines Betriebes, geographische Einteilungen der Welt und Dateisysteme mit Ordnern und Dateien. Die Evolution der Relationen zwischen Blattknoten einer Informationshierarchie kann mithilfe einer Folge von Compound (Di-)Graphen modelliert werden. Die Stärke der Relation kann mit gewichteten Kanten ausgedrückt werden.

Ein einfaches Beispiel für einen Compound Digraph ist der Graph in Abbildung 4.3. Hierbei dient die Unterteilung der Domain als Hierarchie H und der Digraph G wird mittels Kanten dargestellt.

Die Beschreibung von Beziehungen und Abhängigkeiten innerhalb einer Menge von Objekten mittels Graphen ist ein weit verbreitetes Mittel in der Informatik und allen anderen Wissenschaften. Das Graphzeichnen ist eine eigene Disziplin der Informatik und beschäftigt sich u.a. mit dem Problem, einen Graphen dem Benutzer möglichst effizient zu präsentieren. Dabei ist zwischen statischen und dynamischen Graphen zu unterscheiden. Im statischen Fall wird nur ein Graph dargestellt, wogegen im dynamischen Fall eine Folge von Graphen gezeichnet wird. Für kleine einzelne Graphen genügt es, sich mit dem statischen Graphzeichnen zu beschäftigen. In unserem Fall werden jedoch dynamische Graphen vorausgesetzt. Laut [BBD09] werden die folgenden Kriterien für dynamische Graphen gewünscht:

- Maximierung der Anzeigestabilität zwischen den Zeitpunkten (Dynamic Stability)
- Verminderung der kognitiven Last bei der Zeit-Dynamik Analyse
- Minimierung der zeitlichen Aliase

Hierbei wird eine statische Anzeige verwendet, um visuell die zeitlichen Veränderungen von Graphelementen zu repräsentieren. In der Regel sind Benutzer in der Lage, ihre eigenen größeren Änderungen in den Daten zu erkennen und sich daran zu erinnern. Die statische Sicht wird für eine detailliertere Analyse der Datenänderungen bevorzugt, weil Vergleiche über mehrere Zeitschritte einfacher sind als etwa bei einer Animation [VKSKVFF11]. Statische

Ansichten, die auch die zeitliche Dimension der Daten beinhalten, sind allerdings komplexer zu designen.

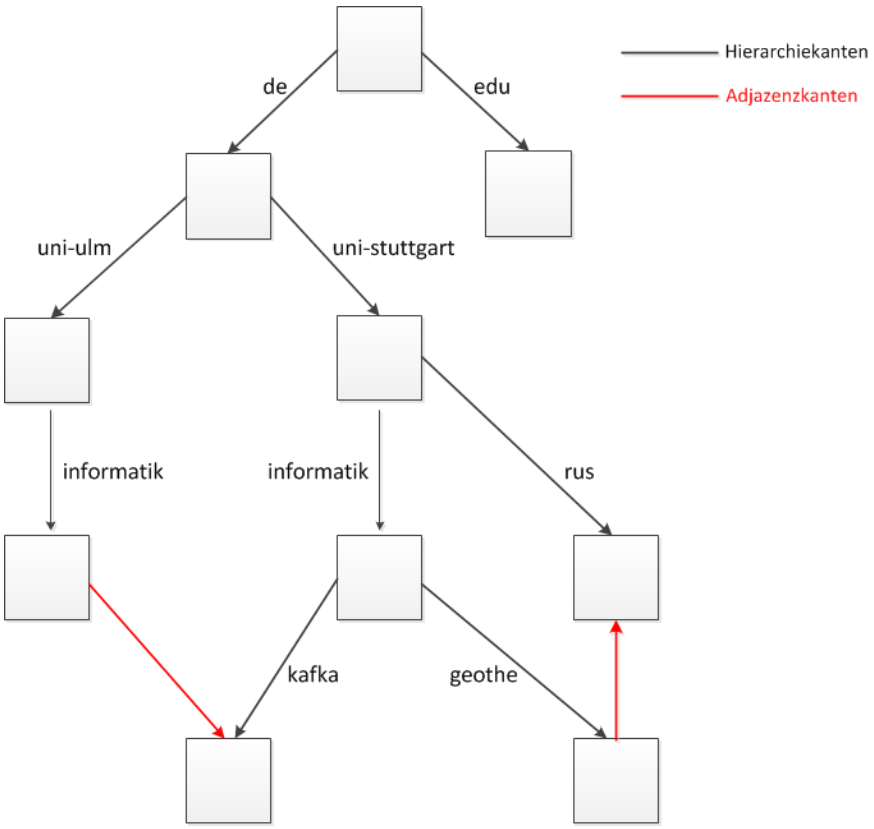


Abbildung 4.3: Oft unterliegen die Knoten V eines Digraphen einer zusätzlichen Hierarchie. Dies kann durch einen Compound Digraph modelliert werden. Sei V eine Menge von Knoten und E_1 sowie E_2 Mengen von Kanten. Sei weiter H = (V, E_1) ein Baum, also ein spezieller Graph, mit dem sich Hierarchien modellieren lassen und ein Graph G = (I, E_2), wobei I ⊂ V. Der Baum definiert somit eine Hierarchie auf der Menge I der Knoten des Graphen G.

Um Graphen bildlich darzustellen, wurden bereits viele Algorithmen entwickelt. Viele dieser Algorithmen können aber nur statische Graphen zeichnen und nicht Folgen von Graphen. Nur wenige sind in der Lage, mit wechselnden Relationen umzugehen. Abhängigkeiten und Relationen zwischen verschiedenen Objekten verändern sich in der Regel über die Zeit. Um interessante Einsichten in die dynamischen Abhängigkeiten der sich zeitlich verändernden quantitativen Werte zu bekommen, wird hierfür das im Kapitel 3.3 vorgestellte TimeArcTrees Verfahren [GBD09] im Visualisierungswerkzeug zum Einsatz kommen. TimeArcTrees (siehe Abbildung 3.3 (a)) verwenden einen statischen Ansatz zur Visualisierung dynamischer Compound-Digraphen. Sie zeigen eine Folge von Knoten-Kanten-Diagrammen in horizontaler Knotenausrichtung in einer einzigen Ansicht, wodurch ihr direkter Vergleich unterstützt wird (Mental Map Preservation, Dynamic Stability).

Knoten-Kanten-Diagramme sind für die intuitive Graphvisualisierung geeignet. Jedoch kommt es aufgrund einer hohen Anzahl an Knoten bzw. der Komplexität des Graphen immer wieder zu Visual Clutter. Insbesondere bei dichten Graphen oder einer ungünstigen Position der Knoten überlappen sich die Kanten und lassen sich dadurch nicht mehr voneinander un-

terscheiden. Es führt dann zu einer unübersichtlichen Visualisierung. Bei ungerichteten und ungewichteten Graphen ist das Problem schon spürbar. Falls zusätzliche Informationen wie Kantenrichtung und Kantengewichte hinzukommen, nimmt der Visual Clutter deutlich zu. Die Kanten zwischen den Knoten müssen so gezeichnet werden, dass die Darstellung den Graph nicht unlesbar macht. Hierfür soll ein vernünftiges Darstellungsintervall der gerichteten Relationsbögen anhand der dynamischen Abhängigkeiten ausgewählt werden (siehe Kapitel 5.3.1).

4.4 Algorithmen für den Entwurf des Themeriver Graphen

Desweiteren ist es nicht das Ziel dieser Arbeit, einen oder mehrere neue Visualisierungsalgorithmen zu entwickeln, sondern vielmehr bestehende Visualisierungswerkzeuge anzupassen und in ein Rahmenwerk zu integrieren. In dieser neuen Technik wird der Themeriver mit den TimeArcTrees, also einem Knoten-Kanten-Diagramm kombiniert. Es gibt dazu drei wesentliche Bestandteile, die unsere Technik bestimmen. Die Form der Gesamtsilhouette ist der erste Bestandteil. Diese Form ist kritisch, da es die Begrenzungslinien und Krümmung der einzelnen Schicht bestimmt. Der zweite wichtige Parameter ist die Auswahl der Farben. Es erlaubt dem Betrachter, verschiedene Schichten zu unterscheiden und potenziell zusätzliche Datendimensionen zu vermitteln. Schließlich ist die Anordnung der Schichten kritisch, die ausgewählt werden können, um verschiedenen ästhetischen Kriterien zu entsprechen. In diesem Abschnitt beschreiben wir Algorithmen, die jede dieser drei Bestandteile in Bezug auf die Aspekte bei der Gestaltung der Lesbarkeit und Ästhetik betrachtet.

4.4.1 Geometrie

Die Geometrie eines Themerivers besteht aus einer Reihe Schichten, die der Zeitserie entsprechen. Mit dem „Macro/Micro“ Prinzip gibt es keinen Raum zwischen den Schichten, so dass die Dicke der gesamten Stapel die Summe der einzelnen Zeitreihen reflektiert. Angesichts dieser Einschränkung wird die gesamte Geometrie der Themeriver Darstellung durch zwei Faktoren bestimmt: Die Form der Grundlinie, d.h. die Unterkante der untersten Schicht, und die Reihenfolge der Schichten. In diesem Kapitel diskutieren wir die Wirkung der Grundlinie auf die gesamte Geometrie des Themerivers und im nächsten Kapitel wird die Auswahl der Farben diskutiert.

Um die Geometrie genau zu beschreiben, verwenden wir die folgende Schreibweise. Wir modellieren unsere Zeitreihen als eine Menge von n reellwertigen nichtnegativen Funktionen, f_1, … f_n. Im Folgenden nehmen wir der Einfachheit halber an, dass diese differenzierbar und auf dem Intervall [0,1] definiert sind. Man könnte auch überlegen, dass Funktionen Funktionswerte zu den vielen diskreten Zeitpunkten annehmen. Aber die Notation ist umständlicher und es ist in jedem Fall einfacher, den diskreten Fall in den differenzierbaren Fall durch Interpolation zu verschieben.

Wir beziehen uns auf die Grundlinienfunktion, die die Unterkante des gestapelten Graphen als g_0 definiert. Die Oberkante der entsprechenden Schicht für die i-te Zeitreihe f_i ist daher durch die Funktion g_i gegeben:

$$g_i = g_0 + \sum_{j=1}^{i} f_j$$

Ein solches Scenario wird für diese Definitionen für n = 2 in der Abbildung 4.4 dargestellt.

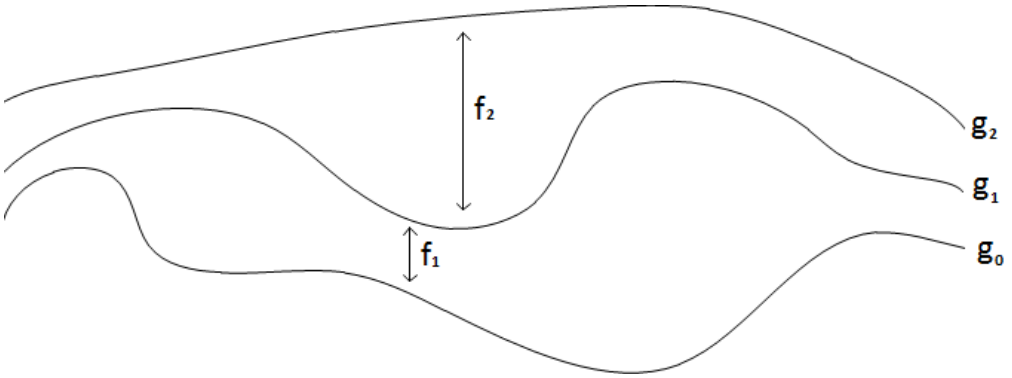


Abbildung 4.4: Eine visuelle Beschreibung von gestapelten Graphfunktionen f_i und g_i für n = 2 wie in diesem Abschnitt verwendet

Es gibt eine Vielzahl von Möglichkeiten, um die Grundlinienfunktion g_0 zu definieren. Die einfachste davon ist der konventionelle gestapelte Graph (siehe Abbildung 4.5), deren Schichten auf eine ebene Grundlinie aufgetragen werden:

$$g_0 = 0$$

Alle nachfolgenden Schichten sind relativ zu den darunterliegenden Schichten gezeichnet. In dieser und der folgenden Abbildung verwenden wir zufällig zugeordnete Farben, um Schichten zu unterscheiden. Der Vorteil dieses Layouts ist, die Summe der Werte einfach zu erkennen, während die einzelnen Schichten nicht so leicht ersichtlich sind.

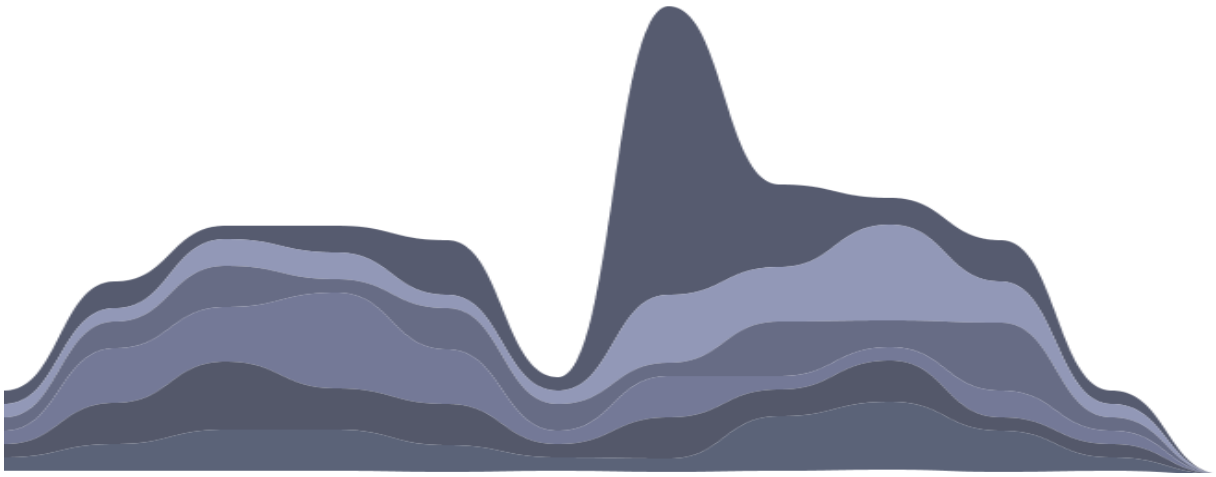


Abbildung 4.5: Ein konventioneller gestapelter Graph mit der Grundlinie $g_0 = 0$.

Ein einfach alternatives Layout wurde von Havre et al. in dem Themeriver-System [HHNW02] vorgeschlagen. Sie verwenden ein Layout symmetrisch zur x-Achse. Mathematisch kann dies wie folgt ausgedrückt werden:

$$g_0 + g_n = 0$$

oder aus der Definition von g_n,

$$2g_0 + \sum_{i=1}^{n} f_i = 0\,,$$

welche die Themeriver Lösung (siehe Abbildung 4.6) für g_0 ergibt:

$$g_0 = -\frac{1}{2}\sum_{i=1}^{n} f_i$$

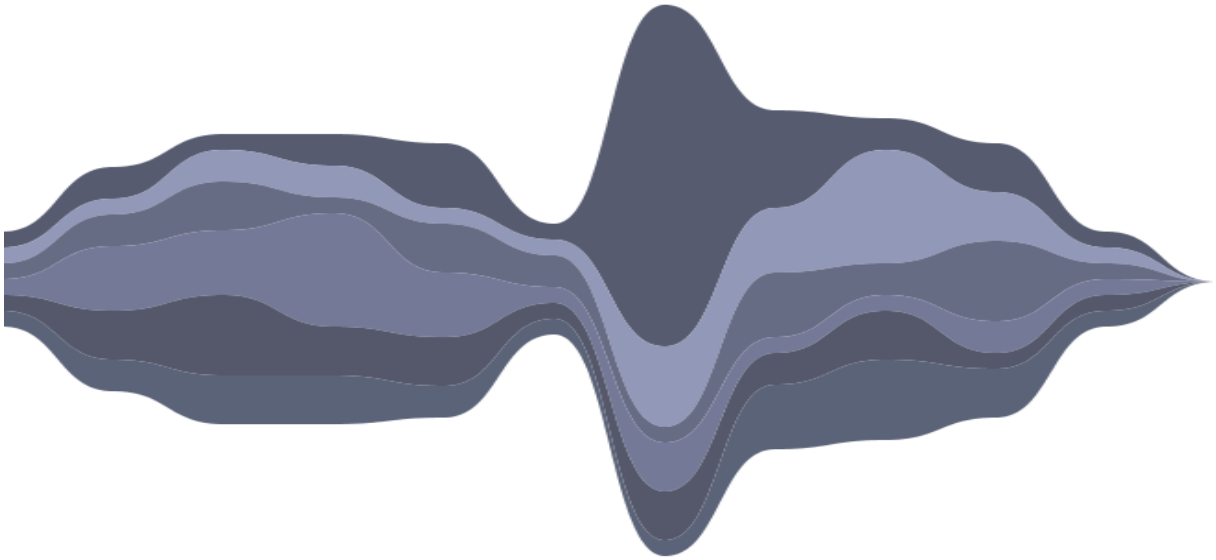


Abbildung 4.6: Die gleichen Daten mit dem Themeriver Layout Algorithmus

Dies ist eine ziemlich einfache Definition. Allerdings kann man aus dieser Formel einige Verallgemeinerungen gewinnen. Neben einer bestimmten ästhetischen Qualität, lässt dieses Layout einige wichtige Quantität zu minimieren. Vor allem an jedem Punkt schmiegt sich die Silhouette so nah wie möglich an der x-Achse an und hält zusätzlich die Steigungen der Oberkante und der Unterkante so gering wie möglich (im Sinne von Gesamtsummen der Plätze). Für eine Menge von reellen Zahlen $\{a_1,..., a_n\}$ minimiert x die Funktionswerte:

$$\sum_{i=1}^{n}(x + a_i)^2, \text{ wobei } x = -\frac{1}{n}\sum_{i=1}^{n} a_i$$

Aus dieser Tatsache folgt, dass der Wert von g_0, welcher das symmetrische Themeriver Layout ergibt, die Summe der Quadrate der Oberkante und der Unterkante der Silhouette ist (an jedem Punkt im Intervall [0,1]):

$$silhoutte(g_0) = {g_0}^2 + {g_n}^2, \text{da } {g_0}^2 + {g_n}^2 = {g_0}^2 + (g_0 + \sum_{i=1}^{n} f_i)^2$$

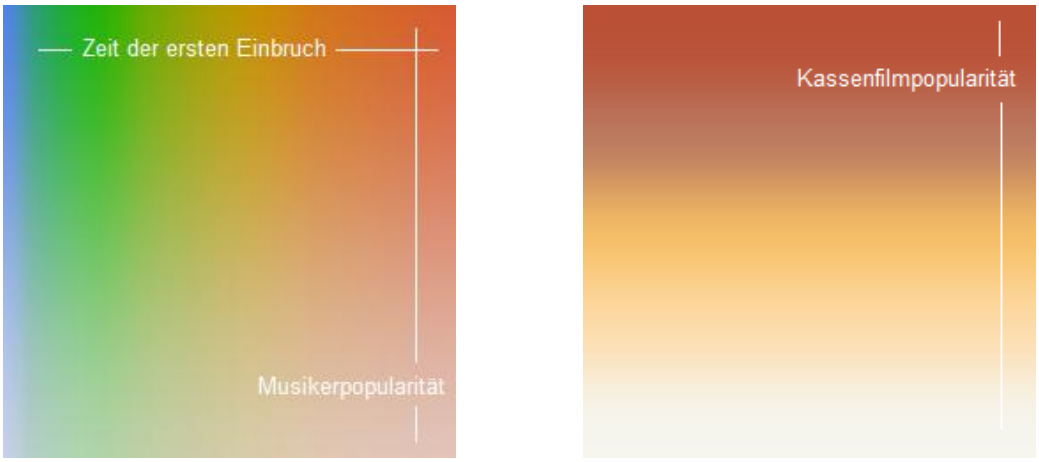
Eine einfache Kalkulation zeigt, dass das Themeriver Layout die Summe der Steigung von g_0 und g_n an jedem Punkt minimiert.

Vor diesem Hintergrund produziert das Themeriver Layout nicht nur eine ziemliche Symmetrie, sondern ist optimal im Sinne der Minimierung bestimmter mathematischer Maßnahmen der Verzerrung.

4.4.2 Auswahl der Farben

Bereits im Kapitel 2 werden die Grundlagen der Wahrnehmung durch das Auge erklärt, die im Folgenden Abschnitt hilfreich zur Auswahl der Farben sind. Das Färben der gestapelten Diagramme mit vielen Schichten ist eine Herausforderung. Obwohl Farbe ein visuelles Feature ist, das mit zusätzlichen Daten korrespondiert, werden starke oder grelle Farben visuell ablenkend und machen die Grafik schwer zu lesen. Gleichzeitig muss genügend lokaler Kontrast zwischen den Schichten durch eine spezielle Farbabbildung vorhanden sein, um jede Schicht zu differenzieren. Das Designproblem wird durch das Bedürfnis der Ausgleichung der ästhetischen Überlegungen kompliziert. (Sieht die endgültige Grafik gut aus? Sind seine emotionalen Bedeutungen mit der Natur der Daten im Einklang stehend?)

Diese komplexen Kompromisse bedeuten, dass die Auswahl eines Farbschemas stark abhängig von den zugrundeliegenden Daten sowie dem Kontext ist, in dem sie präsentiert werden. In diesem Abschnitt beschreiben wir die Entscheidungen hinter unserem Projekt. In unserem Fall reflektieren die Dunkelheit und die Sättigung der subjektiven Farbe (siehe Kapitel 2.3) für bestimmte Zeitreihen die Gesamtsumme der Reihe. Dieses lenkte die Aufmerksamkeit auf Reihen mit größeren Summen (beliebte Themen), die tendenziell wichtiger waren.



(a) (b)

Abbildung 4.7: (a) Der 2D-Farbraum mit der Codierung der Einbruchszeit. (b) Der 2D-Farbraum ohne Berücksichtigung der Einbruchszeit.

Für die Auswahl eines Farbschemas bieten wir zwei Varianten an. Bei der ersten Variante kodiert die Farbe für jede Zeitreihe in unserer Grafik auch den Zeitpunkt des Einbruchs. In der Visualisierung wird die Einbruchszeit mittels eines visuellen Farbverlaufs von kühlen Farben zu warmen Farben gezeigt. Die Einbruchszeit kann wichtig aufgrund der besonderen Form der Daten (z.B. Datensatz A: Hörgewohnheiten) sein. Musiker können möglicherweise einmal ihren Höhepunkt erreichen, wenn sie entdeckt werden, aber dann erfährt man oft viel später das Wiederaufleben. Bei anderen Daten (z.B. Datensatz B: Kinofilm) wird nicht das gleiche Wiederaufleben als Tendenz erfahren. Wegen der kurzen Dauer von Kinofilmen und ihres Mangels an Wiederaufleben, gibt es keinen Bedarf an der auf Einbruchszeit basierten

Unterscheidung, da sie durch Platzierung allein fast immer offensichtlich ist. Die Farben für den Datensatz, der die Einbruchszeit nicht berücksichtigen muss, werden so ausgewählt, dass der Farbverlauf entlang der y-Achse Bedeutendes zu nicht Bedeutendes darstellt (siehe Abbildung 4.7 (b)).

Für den Datensatz, der mit Einbruchszeit intensiv zu tun hat, werden die Farben so ausgewählt, dass das Spektrum von Altem bis Neuem, von Bedeutendstem oder häufigsten Gehörtem, bis wenig bedeutendem Konstrukt als zweidimensionaler Farbverlauf dargestellt wird (siehe Abbildung 4.7 (a)). Der Farbverlauf quer durch die x-Achse ist eine ausführliche Bewegung durch die Farbe, die den „schwachen Kern" der bekannten Musiker gegenüber den „heißen neuen" Entdeckungen der neuen Musiker darstellt, während der Farbverlauf entlang der y-Achse Bedeutendes zu nicht Bedeutendes darstellt, durchweg verringert in der Sättigung und etwas in der Helligkeit erhöht.

Die verwendeten Farben in der ersten Variante sind nicht strikt computergeneriert und kein reiner Übergang durch den Farbton. Stattdessen sind sie optisch ausdrucksstark aus Sicht des Designers zusammengebaut. Die Farben werden von in hohem Grade gesättigten Bildern der Natur gewählt. Das Blau ist von einem klaren Himmel, das Grün von einem Baumblatt und das Rot, das Orange und das Gelb von den Bildern der Flamme. Diese Farben werden dann in einem Farbverlauf unter Verwendung von Photoshop gebildet und geben spezifische Sorgfalt zur Interpolation der Farbe zwischen diesen Kernpunkten und kompensieren die Unterschiede zwischen numerischer und Wahrnehmungskonsistenz. Insbesondere sehen diese Farben natürlich und angenehm aus und sie sind nicht übersättigt.

Das Farbspektrum sollte nicht mit einer „Rainbow Map" verwechselt werden [BDT07]. Erstens stellt dieses Spektrum nur die Hälfte des verfügbaren Farbtons dar und kennzeichnet einen klaren Unterschied zwischen den Extremen des Datensatzes. Vielmehr wird das Farbspektrum gewählt, um ein ergänzendes Farbschema zwischen den alten und den neuen Schichten und gleichzeitig auch analoge Farbschemata zwischen den einzelnen Schichten darzustellen.

Der Kernmusiker, der früh in dem Datensatz erscheint, umfasst meist viel mehr Fläche als ein kürzlich entdeckter Musiker. Um eine stark blau farbige Graphik zu vermeiden, ist dieser Farbverlauf in Richtung zu den wärmeren Farben verschoben. Das wirkt der Neigung des gemeinsamen Bereichs in Richtung zur früheren Einbruchszeitreihe entgegen und gibt der resultierenden Grafik ein Gleichgewicht zwischen den warmen und kühlen Farben.

Wie kann man der Themeriver Graph und Knoten-Kanten-Diagramm besser integrieren? Die Farbauswahl spielt wieder eine wichtige Rolle. Da Themeriver Graph schon stark farbig ist, somit ist die Überlegung einer Kodierung der Kantengewichte in Farben nicht mehr sinnvoll. Wenn viele farbige Kanten quer über den Flüssen gezeichnet würden, so konnte man die Kanten und Flüsse nicht so leicht unterscheiden. Hierfür wird die Kanten einfach in schwarzer Farbe dargestellt.

4.4.3 Anordnung der Schichten

Eine endgültige Auswahl im Entwurf unseres Themeriver Graphen ist die Reihenfolge der Schichten. Die Reihenfolge kann gewählt werden, um die Lesbarkeit zu erhöhen oder einen besser aussehenden Graph zu erzeugen. Im Rest dieses Abschnitts beschreiben wir, wie diese

Wahl für den Datensatz, der mit Einbruchszeit intensiv zu tun hat, und für den Datensatz, der die Einbruchszeit nicht berücksichtigen muss, getroffen wurde. Die Beispiele illustrieren sowohl die Wechselwirkung zwischen ästhetischen und kommunikativen Anteilen als auch wie bestimmte statistische Eigenschaften des Datensatzes möglicherweise die Geometrie beeinflussen.

Eine bestimmte Art der Explosionsartigkeit charakterisiert sowohl den Datensatz A (Hörgewohnheiten) als auch den Datensatz B (Kinofilm). Typische Zeitreihen in jedem Satz fangen bei Null an - ein Musiker ist unbekannt oder ein Film wird noch nicht freigegeben - und bleiben Null für eine Weile und dann explodieren sie plötzlich zu einem Maximum - ein Musiker wird entdeckt, ein Film wird freigegeben - gefolgt vom Verfall der Werte - ein Musiker wird langweilig, ein Film verblasst aus der Öffentlichkeit. Dieses Muster stellt eine Herausforderung für den Themeriver Graph dar, da Explosionen wackelnde Unterbrechungsartefakte in der Geometrie verursachen können (siehe Abbildung 4.8). Aus dem gleichen Grund, wird die Einbruchszeit einer Zeitserie - d.h., die Zeit, zu der sie zum ersten Mal ungleich Null ist - zu einer Variablen, die der Benutzer möglicherweise hervorgehoben sehen will.

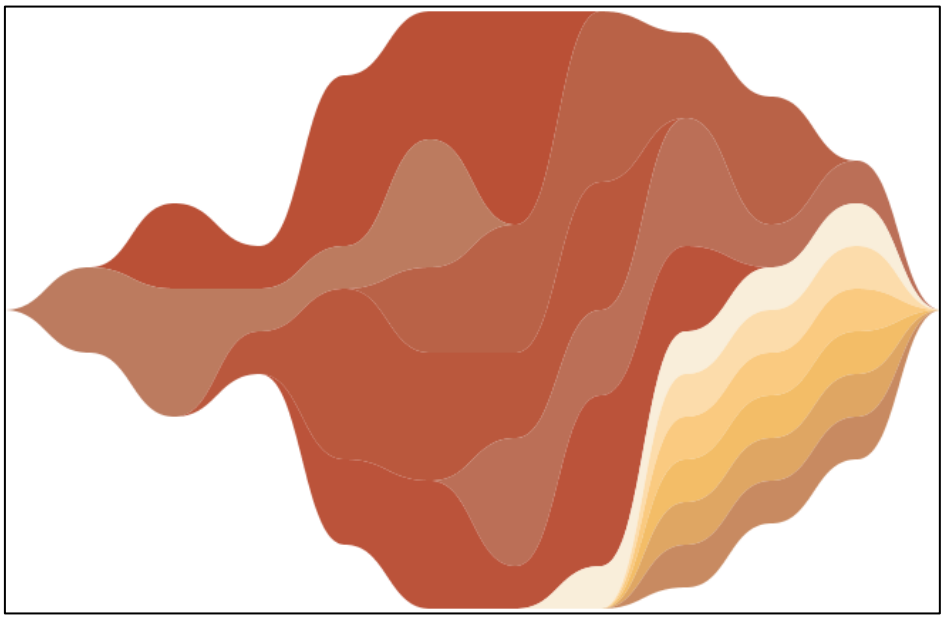


Abbildung 4.8: Ein unsortierter Datensatz, präsentiert die Art der „Burstiness“ (das Verhältnis des Spitzenwerts zu dem durchschnittlichen Wert), die offensichtlich in den Datensätze A und B wird.

Es ist denkbar, dass der Datensatz durch die Einbruchszeit sortiert werden kann. Wenn die „neuen“ Schichten immer entlang dem Boden hinzugefügt werden, nimmt der Graph ein nach oben ablenkendes diagonales Streifenmuster an, außerdem einen nach unten gerichteten Winkel zur gesamten Silhouette aufgrund des Aufwands des Layout Algorithmus', um die Summe von Steigungen niedrig zu halten (siehe Abbildung 4.9).

Um das zu verhindern, wird den Schichten eine „inside-out“ Reihenfolge gegeben, in der frühe Einbrüche in den Zeitreihen an der Mitte später Einbrüche der Zeitreihen an der Ober- und Unterkante auftreten. Außerdem hat das drei Vorteile zur Vermeidung des Schrägstreifeneffektes. Zuerst legt es die größten Explosionen in die Schichten - der erste Wert ungleich

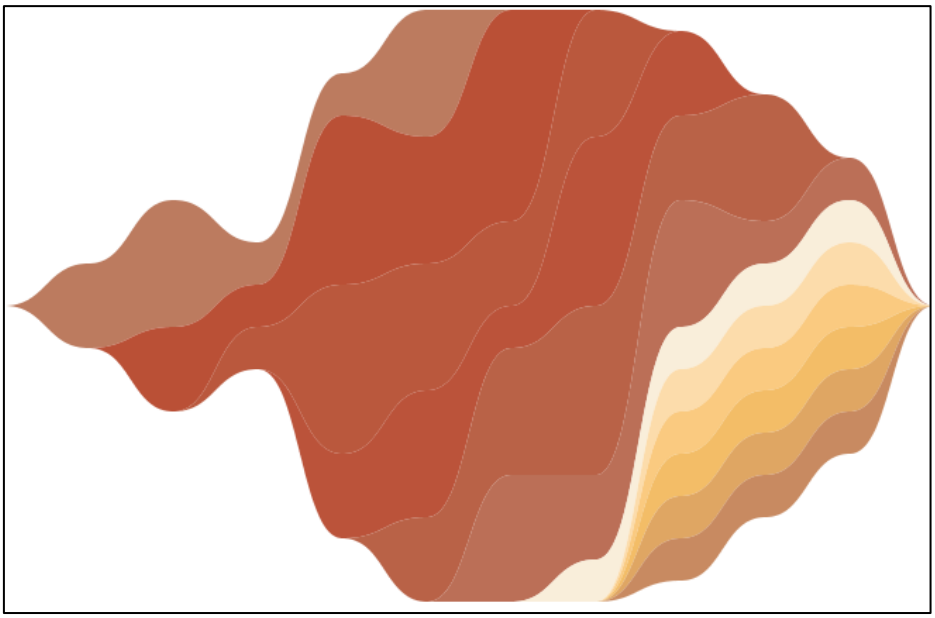


Abbildung 4.9: Der gleiche Datensatz, der naiv in der Reihenfolge der Einbruchszeit sortiert wird, präsentiert den ablenkenden diagonalen Streifendeffekt.

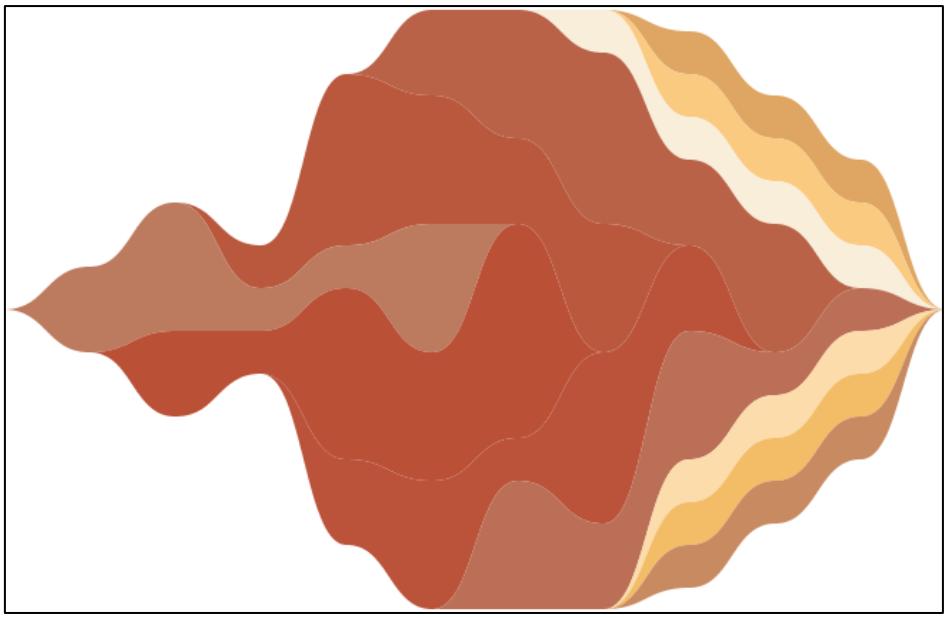


Abbildung 4.10: Der gleiche Datensatz wird unter Verwendung des belasteten „inside-out“-Strategie sortiert, um die Einbruch jeder Zeitreihe hervorzuheben.

Null - an der Außenseite des Graphen, wodurch sie das Layout der anderen Schichten am wenigsten stören aber drastisch die Lesbarkeit verbessern. Zweitens spekulieren wir, dass die Ober- und Unterkantenregionen des Graphen tendenziell die meisten vorstehenden Bereiche sind, da sie nahe der hochauflösenden Silhouette vorkommen. Der zentrale „Kern“ des Graphen (die Mitte) wird möglicherweise zweitrangig gelesen. Da die Explosionen der interessanteste Teil der Daten in vielen Fällen sind, legt das „inside-out“-Layout sie in die möglicherweise herausragende Position (siehe Abbildung 4.10).

Drittens, es verhindert eine Abweichung des Layouts weg von der x-Achse und ein Artefakt, das in Abbildung 4.9 erkannt werden kann.

Die Einzelheiten der „inside-out"-Reihenfolge werden wie folgt definiert. Es ist zu beachten, dass eine einfache Methode, die Schichten durch die Einbruchszeit einfach zu sortieren, wäre und dann die Schichten alternativ dem Anfang und dem Ende einer Schichtliste hinzuzufügen wären. Leider könnte diese einfache Methode möglicherweise zu einer sehr asymmetrischen Grafik führen, wenn die Schichten, die am Anfang der Liste enden, viel größere Werte als diejenigen am Ende der Liste repräsentieren.

Um diese Asymmetrie zu verhindern, verwenden wir den folgenden Algorithmus, wenn wir die Schichten anordnen. Zuerst definieren wir das „Gewicht" der Zeitreihen als die Summe aller seiner Werte. Dann nach der Sortierung durch die Einbruchszeit, fügen wir die Zeitreihen nacheinander in die Liste, und versuchen dabei das Gewicht zwischen der oberen und unteren Hälfte auszugleichen: Genauer gesagt, wenn die Summe der Gewichte der ersten Hälfte der aktuellen Liste größer als die Hälfte der gesamten Gewichte ist, fügen wir die Reihen ans Ende ein. Ansonsten, fügen wir diese an den Anfang ein.

4.4.3.1 Minimierung der Kantenlängen

Je nach Art der Zeichnung des Graphen wird die enthaltene Information besser oder schlechter zum Ausdruck gebracht. Deshalb wurden ästhetische Kriterien festgelegt, welche die Lesbarkeit von Graphen erhöhen sollen. Eines dieser Kriterien ist, den Graphen so anzuordnen, dass möglichst die Kanten mit minimierten gesamten Längen gezeichnet werden. Aber um dies zu erreichen, müsste wiederum die Anordnung der Knoten manipuliert werden. Da der Startknoten und Zielknoten von einer Kante sich in der Mitte auf der jeweiligen Schicht befindet, kann ein Tausch der Reihenfolge durchgeführt werden, damit die Gesamtlänge der Kanten dadurch kleiner wird.

Bevor die Relationsbögen auf dem Themeriver Graph gezeichnet werden, muss zuerst eine Frage beantwortet werden: Ist die Zeichnung der Relationsbögen überhaupt notwendig? Oft tritt der Fall ein, dass der Schwellwert (die Summe der Relationen) zu hoch eingestellt wird, und dadurch wird kein einziger Bogen gezeichnet. Wenn die Zeichnung der Bögen notwendig ist, dann wird ein Algorithmus zur Minimierung der Kantenlängen verwendet. Es kann also die Reihenfolge der Knoten geändert werden. Für einen Graph mit n Knoten existieren n! viele Möglichkeiten, die Knoten anzuordnen. Bei einem Themeriver Graph mit 100 Flüssen, die jeweils in der Mitte einen Knoten haben, gibt es 100! Kombinationen, die überprüft werden müssten. Dies ist auch bei einer effizienten Implementierung und schnellen Rechnern nicht in kurzer Zeit durchzuführen. Das Problem wird so gelöst werden, dass der Algorithmus erst bei einem Themeriver Graph mit weniger als 10 Flüssen verwendet wird.

Bei einer Reihenfolge der Flüsse werden die zu zeichnende Kanten in den ganzen Relationsdatensätzen durchgesucht. Dabei werden alle Kantenlängen zusammenaddiert und die Summe aller Kantenlängen wird in einer Liste gespeichert. Nach der Überprüfung mit allen möglichen Kombinationen der Reihenfolge wird eine minimale gesamte Kantenlänge in der Liste herausgefunden. Zum Schluss wird die Anordnung der Flüsse, die eine minimale Gesamtkantenlänge aufweisen, zur Darstellung des erweiterten Themeriver Graphen übernommen.

4.5 Interaktive Features

Wie bereits erwähnt, soll das Visualisierungswerkzeug die Themeriver Visualisierung interaktiv erweitern. Der angezeigte Themeriver soll nach den Wünschen des Benutzers möglichst interaktiv manipulierbar sein. Es muss daher ein fester stets sichtbarer Bereich zur Parametereinstellung reserviert werden. Somit wird das Programm durch das Werkzeugpanel und die Menüleiste (siehe Abbildung 4.11) vervollständigt, welche die Funktionalität um einige Optionen erweitern.

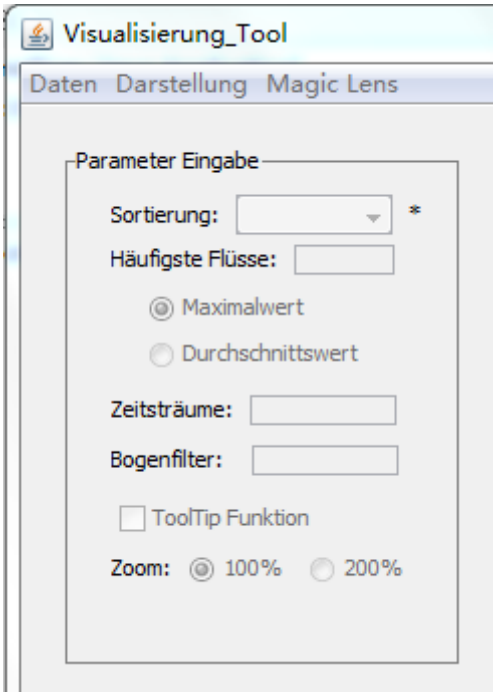


Abbildung 4.11: Werkzeugleiste mit interaktiven Funktionen.

Das Werkzeugpanel ist mit Texteingabefeldern, Schaltflächen und Auswahlboxen versehen, mit denen man interaktiv Einfluss auf die Visualisierung nehmen kann. Es sind unter anderem Algorithmen zur Bestimmung von kürzesten Gesamtkantenlängen, Anordnung der Schichten und Farbschemaauswahl, sowie Filter für die dynamischen Graphen anwendbar. Schließlich sind noch einige weitere Funktionen in der Menüleiste untergebracht. So gibt es dort die Möglichkeit, die Visualisierung als Bild zu exportieren oder zu drucken und einige Einstellungen am Werkzeug vorzunehmen.

Der Benutzer bekommt in der Startansicht (Abbildung 4.12) eine grundlegende Übersicht über den Graph, den er nun Schritt für Schritt mit Hilfe von unterschiedlichen interaktiven Funktionen und Tooltips erforschen kann, welche in den folgenden Kapiteln (4.5.1-4.5.7) genauer beschrieben werden.

4.5.1 Modi für die Anordnung der Schichten

Mit der Auswahlbox „Sortierung“ kann die Anordnung der Schichten manipuliert werden. Je nach Datensatz sind unterschiedliche Einstellungen sinnvoll.

- **Keine:** Die erste Einstellung ist keine Anordnung der Schichten. Bei dem Datensatz, der die Explosionsartigkeit nicht berücksichtigen muss, ist keine Sortierung der

Schichten sinnvoll. Darüberhinaus wird das Farbschema verwendet, das in Abbildung 4.8 (b) dargestellt ist, da die Farbe für jede Zeitreihe den Zeitpunkt des Einbruchs nicht kodieren muss. Hier wird der Farbverlauf entlang der y-Achse in der Form Bedeutendes zu nicht Bedeutendes dargestellt.

- **Onset:** Bei dem Datensatz mit explosionsartigem Charakter soll die Option „Onset“ gewählt werden. Hierbei werden die Schichten unter Verwendung der belasteten „inside-out“ Strategie sortiert und das Farbschema verwendet, das in der Abbildung 4.8 (a) dargestellt ist. In der Visualisierung wird die Einbruchszeit mittels eines visuellen Farbverlaufs entlang der x-Achse von kühlen Farben zu warmen Farben gezeigt, wobei der Farbverlauf entlang der y-Achse in der gleichen Bedeutung wie bei der Option „Keine“ angegeben ist.

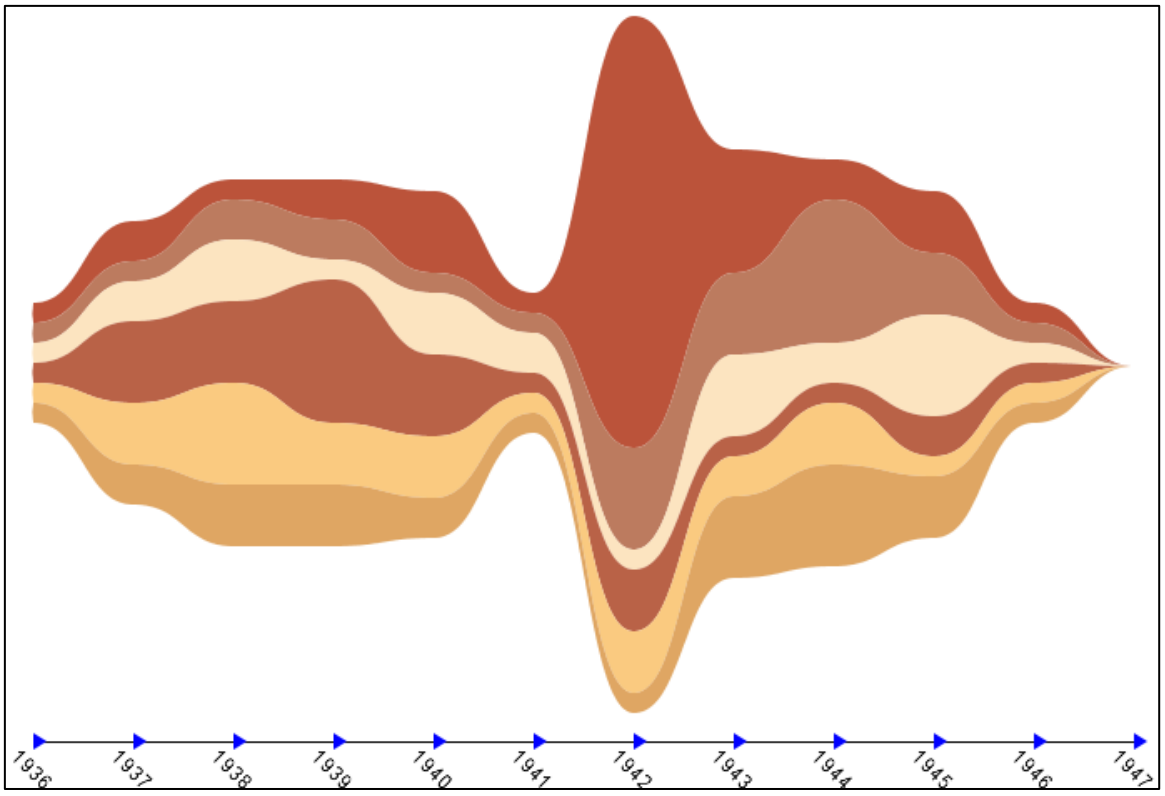


Abbildung 4.12: Startansicht - Überblick

In Abbildung 4.9 und 4.11 sieht man den Vergleich zwischen den Optionen „Keine“ und „Onset“. Die Abbildung 4.9 zeigt also offensichtlich die „Explosionsartigkeit“ im Datensatz. Da Explosionen wackelnde Unterbrechungsartefakte in der Geometrie verursachen können, zeigt die erste Visualisierung (Keine) nicht mehr die Einbrüche jeder Zeitreihe richtig an. Dies ist allerdings oft wünschenswert und wird mit der Sortierung des Datensatzes erreicht. So kann der Einbruch jeder Zeitreihe sinnvoll hervorgehoben werden, wie es in Abbildung 4.11 dargestellt wird. Zum Beispiel erkennt man hier, dass die Explosionen in der linken Hälfte des Graphen unübersichtlich sind, wie es die Darstellung in Abbildung 4.9 ohne Sortierung der Schichten andeutet. Stattdessen soll hervorgehoben werden, wie es in Abbildung 4.11 durch die Sortierung der Schichten signalisiert wird.

4.5.2 Flussfilter

Im Werkzeug existiert ein Filter, mit dem man die Anzahl der gezeichneten Flüsse einstellen kann, um die zu betrachtenden interessanten Flüsse einzuschränken. Oft sind die Themenflüsse mit schmalen Segmenten für den Benutzer uninteressant, aber müssen trotzdem auch visualisiert werden und führen deshalb zu einem sehr dichten Graph. Sinnvoller wäre es hier, diese Flüsse nicht mit darzustellen, weil sie für den Betrachter vernachlässigbar sind. Hierbei kommt der Filter zum Einsatz. Der Filter hat nicht nur die untere Schranke als Parameter, sondern auch eine unsichtbare obere (die Anzahl der darzustellenden Flüsse darf nicht größer als die Anzahl der originalen Themenflüsse sein). Es kann zum Beispiel benutzt werden, um mehr Platz für die Darstellung der für den Betrachter interessanten Flüsse bereitzustellen.

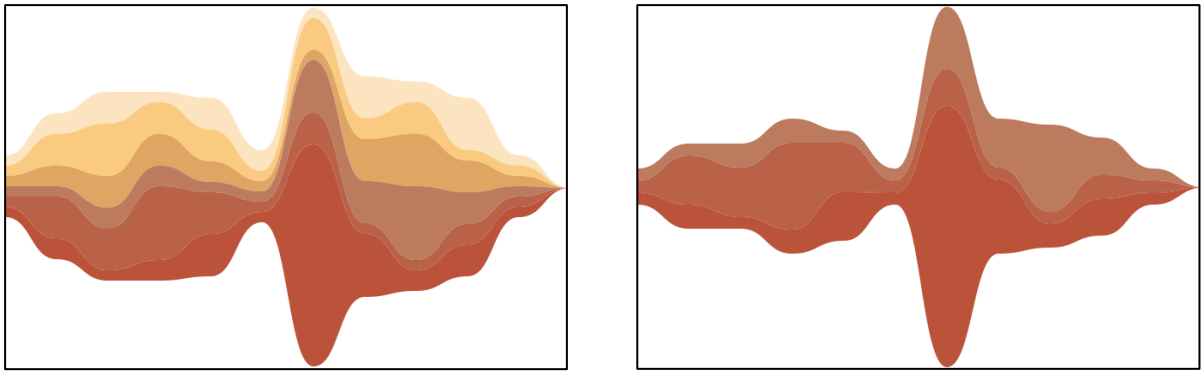


Abbildung 4.13: Flussfilter (6 Flüsse / 3 Flüsse)

In Abbildung 4.13 sieht man den Vergleich zwischen den Flussfiltereingaben „6“ und „3“. Die linke Visualisierung mit der Eingabe „6“ stellt ein Themeriver Graph mit 6 Themenflüssen dar. So bestimmt die aktuell eingegebene Zahl „3“ (die häufigsten aufgetretenen Flüsse) die auf der rechten Visualisierung darzustellenden Flüsse.

In das Texteingabefeld „Häufigste Flüsse“ im Werkzeugpanel kann ein Filterkriterium eingegeben werden. Unter dem Texteingabefeld befinden sich noch zwei Optionsfelder, die zwei Sortierungsarten des einzelnen Flusses zur Auswahl bereitstellen. Mit dem Optionsfeld „Maximalwert“ können die Themenflüsse durch den Maximalwert vom einzelnen Fluss sortiert werden und anschließend die entsprechenden Flüsse anhand des Texteingabefeldes „Häufigste Flüsse“ im Graph darstellen. Der Vorteil ist, dass der Zeitpunkt, zu dem das Thema vorherrschend war, im Graph mehr Fokus bekommt. Der Nachteil ist allerdings, dass unerwünschte Extremwerte dabei auftreten könnten. Wenn die meisten Variablen gleichmäßig im Wertebereich verteilt sind, können die Themenflüsse durch den Durchschnittswert von einzelnem Fluss mit dem Optionsfeld „Durchschnittswert“ sortiert werden, damit solche Ausreißer vernachlässigbar sind. Diese können mit dem Filter dann entfernt werden.

4.5.3 Datensatzfilter

Über das Texteingabefeld „Zeiträume“ können die Datensätze mit unterschiedlichen Zeitstempeln anhand des Benutzerinteresses gefiltert werden, damit mehr Platz für die Darstellung mit interessanten Datensätzen zur Verfügung steht. Bei zeitabhängigen Datensätzen kommt die Explosion der Datenwerte nicht stets von Anfang an vor. Aus diesem Grund ist die Interaktionsmöglichkeit sehr sinnvoll und hilfreich.

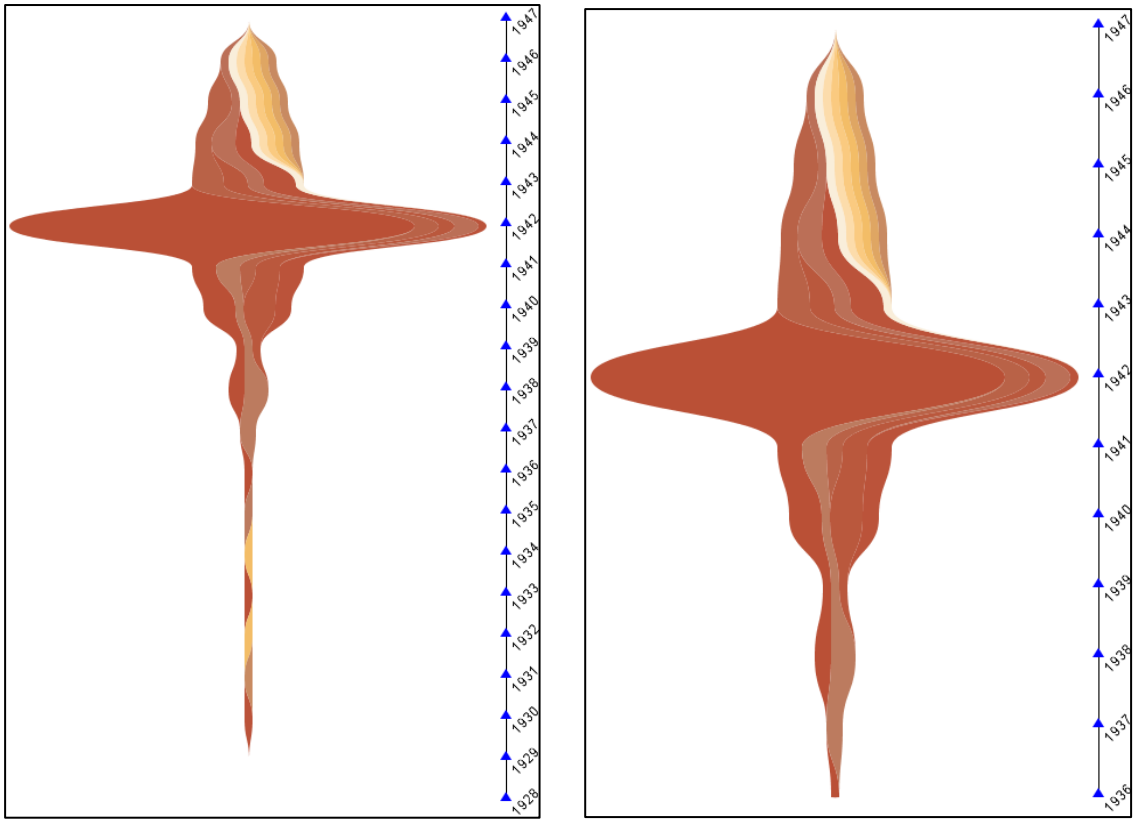


Abbildung 4.14: Datensatzfilter (Zeitraum 1928-1947 / Zeitraum 1936-1947)

In Abbildung 4.14 sieht man den Vergleich vor und nach der Filtereingabe „1936-1947“. Das linke Bild ohne Filtereingabe stellt einen Themeriver Graphen mit Datensätzen im Zeitraum 1928-1947 dar. Es ist leicht erkennbar, dass die Datensätze im Zeitraum 1928-1936 vernachlässigbar sind, da die Explosion des Datensatzes erst am Zeitpunkt 1936 vorkommt.

Durch die Datensatzfilterung vergrößert sich der interessante Teil des Graphen, wie es in rechtem Bild in Abbildung 4.14 dargestellt ist.

Zusätzlich kann man die sich auf der Zeitachse befindlichen blauen Dreiecke anklicken. Der angezeigte Themeriver wird so in einzelnen Schritten interaktiv manipulierbar gemacht und es wird signalisiert, dass der Datensatz an dem angeklickten Punkt gefiltert wird.

4.5.4 Kantenfilter

Um die Anzahl der gezeichneten Kanten zu verkleinern, existiert im Werkzeug noch ein anderer Filter, mit dem man den Wertebereich der gesamten Gewichte einschränken kann. Diese Kanten werden auch bei der Berechnung von kürzester Gesamtkantenlänge ignoriert. Oft trifft der Fall ein, dass man Kanten mit kleinen Werten hat, die nicht viel zum Verständnis des Graphen beitragen, aber trotzdem mit visualisiert werden und zu sehr dichten Zeichnungen führen. Es wäre also sinnvoller, diese Kanten nicht mit anzuzeigen, weil sie vernachlässigbar sind. Dies kann mit dem Kantenfilter erreicht werden. Der Filter hat die untere Schranke als Parameter. Diese kann zum Beispiel benutzt werden, um die Einblendung des Graphen wegen zu dichten Kantenzeichnungen zu verhindern. Wenn man nur das gewöhnliche Verhalten des Graphen darstellen will, sind solche uninteressante Kanten vernachlässigbar und können mit dem Filter entfernt werden.

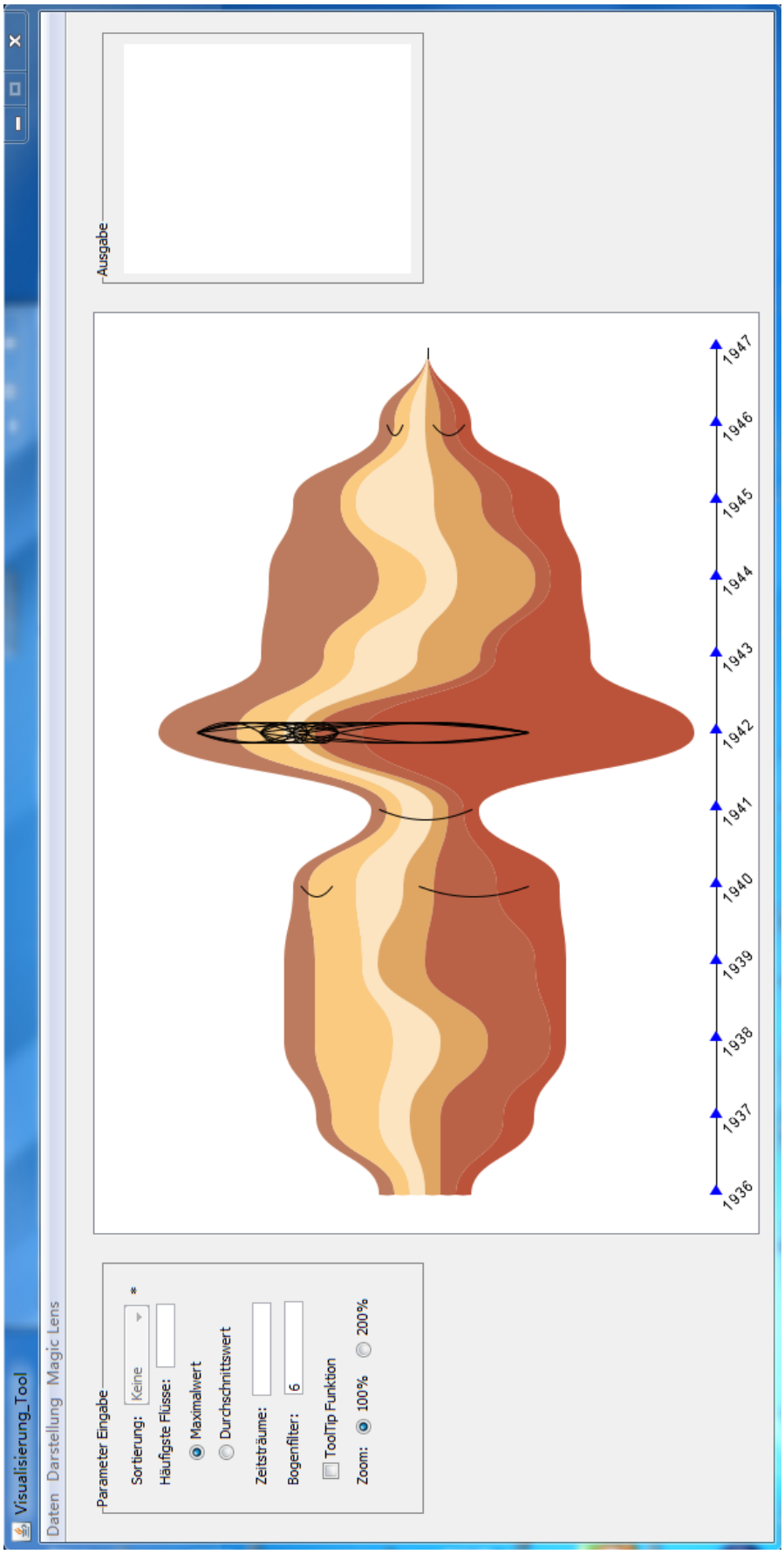


Abbildung 4.15: Kantenfilter zur Einschränkung des Darstellungsintervalles

In der Abbildung 4.15 werden die Kanten mit gesamten Gewichte < 6 herausgefiltert. Die Relationsbögen mit niedrigen Relationswerten sollen also nicht angezeigt werden.

Das Texteingabefeld „Bogenfilter“ im Werkzeugpanel, in dem die Parameter eingestellt werden können, aktiviert den Filter. Beim Aktivieren des Flussfilters wird der Filter wieder deaktiviert.

4.5.5 Tooltips

Per Mouseover kann der Benutzer detailliertere Angaben zu den einzelnen Attributwerten abrufen und den Weg der reinen textuellen Daten zur Visualisierung wieder zurückverfolgen. Hierzu können bei Bedarf Informationen in Textform eingeblendet werden. Eine erste Variante sind Tooltips, die erscheinen, wenn der Mauszeiger längere Zeit auf einer Stelle verbleibt. Zudem wird die selektierte Layer Form ebenso per Mouseover hervorgehoben. Es wird dann neben dem Mauszeiger ein Textfeld mit Informationen zum Element unter dem Zeiger eingeblendet. Als Beispiel wird der Tooltip über einen Fluss im Themeriver Graph erklärt. Er beinhaltet aktuelle Häufigkeitsdaten und Flussnamen aber auch aktuelle Zeitstempel (momentan die nächsten Daten neben dem Mauszeiger, siehe Abbildung 4.16). Die Auswahlbox namens „Tooltip Funktion“ befindet sich im Werkzeugpanel und kann die Tooltip Funktion aktivieren oder deaktivieren.

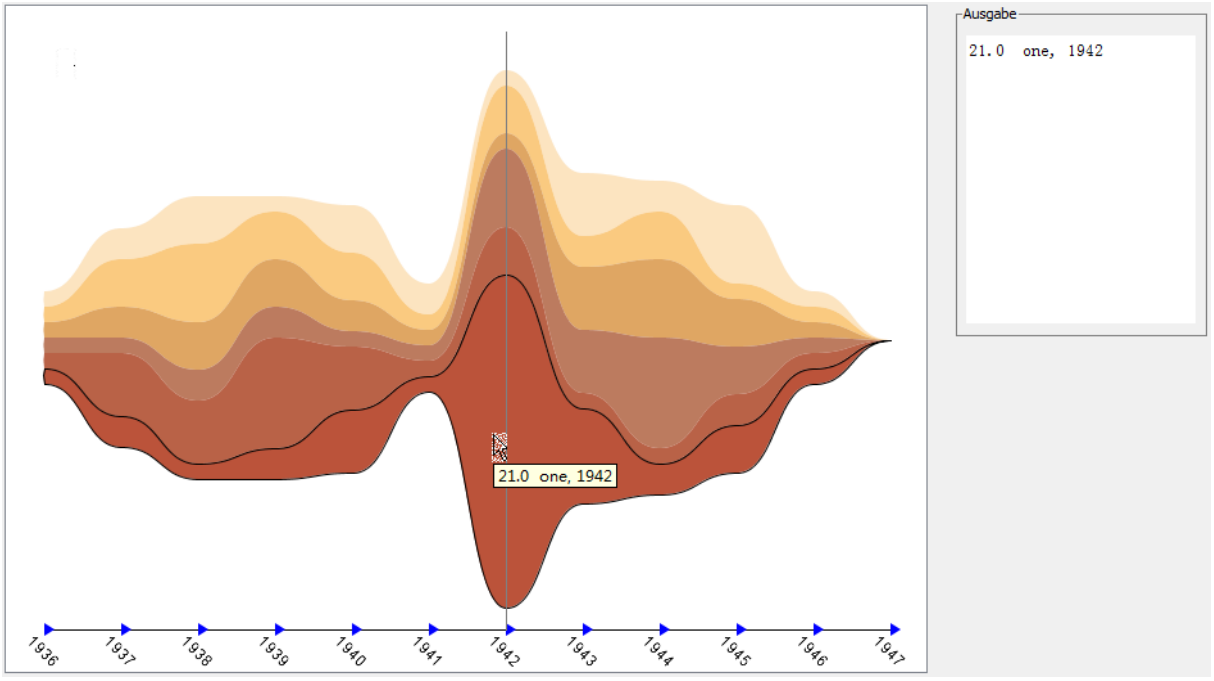


Abbildung 4.16: Details-On-Demand durch Tooltips und Highlighting der selektierten Layer Form per Mouse-Over

Die zweite Variante ist eine dauerhaft angezeigte Beschriftung. Standardmäßig ist sie nur im Ausgabepanel sichtbar. Dort ist sie unerlässlich, um den Graphen ohne Ausblenden der detaillierten Ansicht analysieren zu können. Ohne Beschriftung der aktuellen Themaveränderung ist der Vergleich zwischen unterschiedlichen Themen nicht nachvollziehbar (Es kann sein, dass die Flussdichten sich zu dem entsprechenden Zeitpunkt nur wenig unterscheiden.). Je schmaler die Farbschicht an einer bestimmten Stelle ist, desto weniger vorherrschend war das Thema zu dem entsprechenden Zeitpunkt in dem untersuchten Datensatz.

4.5.6 Zoom

Die Darstellung großer und dichter Graphen mit vielen Kanten bringt Probleme dergestalt mit sich, dass Kanten stellenweise sehr dicht aneinander gezeichnet werden müssen und dass viele Kreuzungen existieren. Es kann auch hin und wieder vorkommen, dass eine große Menge an Knoten in dem Graph existiert (während viele Flüsse in der Themeriver Darstellung gezeichnet werden). Der Betrachter hat Schwierigkeiten, diese Kanten (bzw. die verschiedene Schichten) auseinander zu halten. Dazu steht die Interaktionsmöglichkeit des Zoomings zur Verfügung. Aus der Vorstellung der verwandten Arbeiten 3.4 zeigt sich, dass die Zooming Technik bei der Visualisierung der TimeArcTrees sehr gut gelungen ist. Daher soll sie auch hier ähnlich erfolgen. Die eingebaute Zoomfunktion, die sich in der Werkzeugleiste mittels Schaltfläche ein- und ausschalten lässt, hilft, solche Ballungsstellen in der Zeichnung genauer zu betrachten, indem sie die Ansicht der Visualisierung durch zahlreiche Zooming-Stufen vergrößert und Scroll-Balken-Positionen navigiert. Beispielsweise kann der Zoomausschnitt in doppelter Größe dargestellt werden, um die genauere Untersuchung in der Region des Interesses vom Anwender auszuführen.

4.5.7 Sonstige Funktionen

In der Menüleiste unter dem Punkt „Daten“ befindet sich die Funktion „Bild exportieren“ mit der man die aktuelle Darstellung der Graphen ausgeben kann. Mit der Exportfunktion wird die Darstellung als JPG- oder PNG-Bilddatei an einem frei wählbaren Speicherort abgelegt. Dabei werden die aktuellen Exporteinstellungen benutzt. Diese können durch das Aufrufen von Daten/Exporteinstellungen geändert werden.

Im Menü „Darstellung“ können unterschiedliche Einstellungen bezüglich der Darstellung der Flussgrenzen (Interpolationsart) und der Anzeige der Datenpunkte eingestellt werden. Weiteres kann zwischen einer Histogrammdarstellung, einer Stackedgramm-Darstellung und der eigentlichen Themeriver-Darstellung entschieden werden. Durch Aufrufen von Darstellung/Farbeinstellungen wird ein eigenes Fenster geöffnet, das das Ändern, Speichern und Laden der Farben für die Flüsse gestattet.

Durch das Aufrufen von Magic Lens/Erstellen wird ein Bereich in der aktuellen Visualisierung erstellt, in dem eine andere Darstellungsart angezeigt wird. Die Darstellungsart in diesem Bereich kann über das Magic Lens Menü gesteuert werden. Der Bereich kann durch Ziehen mit der Maus verschoben werden. Durch Ziehen des schwarzen Rahmens, der den Bereich umgibt, kann die Magic Lens verkleinert und vergrößert werden. Der Bereich kann mit Hilfe des Menüs Magic Lens/Löschen wieder gelöscht werden.

4.6 Skalierbarkeit

Im Allgemein spielt die Skalierbarkeit eines Visualisierungswerkzeug eine wichtige Rolle bei der Anzeige der Informationen. Dabei ist die Herausforderung zweifältig. Auf einer Seite muss die Visualisierung eine größere Menge an Daten verarbeiten können. Hier meint man zunächst einmal nicht die Anzeige, sondern allein die Datenhaltung. Auf der anderen Seite gilt es zusätzlich, die angezeigte Menge an Informationen übersichtlich beziehungsweise für die Fachkraft wahrnehmbar zu gestalten. Beide Aspekte werden in diesem Abschnitt näher interpretiert und Lösungskonzepte erklärt.

Unser Visualisierungswerkzeug eignet sich nicht nur für die konventionelle Themeriver Darstellung als auch für die erweiterte Version durch die Kombination mit der TimeArcTrees Technik. Die eventuell zur Darstellung der TimeArcTrees benötigen Relationsdaten in Eingabedateien werden nur ausgelesen, sofern sie vorhanden sind. Außerdem werden bei der Visualisierung eines Datensatz die Daten nur einmal geladen und anschließend in den Arbeitsspeicher gelagert. Dadurch kann die Ladezeit einer größeren Menge an Daten beispielsweise bei einem möglichen Nachladen erspart werden.

Die Daten, die innerhalb der Visualisierung gehalten werden müssen, können im Laufe der Zeit stark anwachsen. Dies kann dazu führen, dass die Visualisierung ausgebremst wird. Hierbei ist eine Datenbereinigung empfohlen. Teile der Daten, die nicht länger gebraucht werden, können wieder verworfen werden, um den Speicherbedarf so zu reduzieren, dass die Visualisierung nach wie vor reaktionsfähig bleibt. Der Vergleich zur Sicherung der Datenrepräsentation ist ein wichtiges Argument, ob die aktuelle Datenrepräsentation weiherhin von Interesse ist. Eine Möglichkeit ist somit die aktuelle Datenrepräsentation zu löschen, die nicht identisch zu der Sicherung der Datenrepräsentation ist. Das heißt, nach jedem Interaktionsschritt wird die originale Datenrepräsentation beibehalten und die eventuell unbrauchbare wieder entfernt.

Algorithmus Datenbereinigung

1: **if** $datrepBackup \neq null \cap isLayerAccountChanged$ **then**

2: $datRep \leftarrow datRepBackup$

3: $tmpDatRep \leftarrow createTempDatRep(datRep)$

4: $datRepBackup \leftarrow datRep$

5: $datRep \leftarrow tmpDatRep$

6: **end if**

Aus der Sicht der graphischen Repräsentation ist unser Visualisierungswerkzeug ebenfalls gut skalierbar, da es die Vorteile der Themeriver Technik und TimeArcTrees Technik nutzt. Die gesamte Themeriver Visualisierung kann in jeder Größe, unabhängig von der tatsächlichen Anzahl der dargestellten Datenelemente, skalierbar gemacht werden, weil der Graph aus natürlichen skalierbaren Flussbändern besteht. Mittels Aggregation kann man dies sowohl in der Fluss- als auch in der Zeitdimension erreichen.

Im anderen Teil, der die TimeArcTrees Visualisierung mit Knoten-Kanten-Diagrammen behandelt, ist eine gute Skalierbarkeit ebenso erreichbar. Die Kanten in den Graphen der Folge verlaufen links bzw. rechts der Vertikalen, dadurch werden mögliche Überlappungen und Kreuzungen der Kanten reduziert. Ein Darstellungsintervall der gerichteten Kanten kann eingeschränkt werden, damit nur wenige Kanten gezeichnet werden müssen. Dies bedeutet, dass das Visual Clutter Problem auch bei einer großen Menge an Knoten stark reduziert wird. Im Prinzip kann die TimeArcTrees Visualisierung beliebige Folgen von Graphen darstellen. Al-

lerdings kann sie mit sehr dichten Graphen nicht so leicht umgehen und lässt das Bild nicht mehr überblicken als Folge von Visual Clutter verursacht durch viele Kantenkreuzungen.

4.7 Performanz

Das Visualisierungswerkzeug wurde in der Programmiersprache JAVA unter der Java-Laufzeitumgebung 1.6 entwickelt. Als Testsystem kommt ein HP G62 Notebook PC mit einem Windows-7-Betriebssystem zum Einsatz. Verbaut sind folgende Hardware-Komponenten:

- Prozessor: Intel Core i5 M460 @ 2.53 GHz Dual-Core (4 Hardware-Threads)
- Arbeitsspeicher: 4 GB
- Graphikkarte: ATI Mobility Radeon HD 5470 @ 750 MHz GPU Takt mit 900 MHz GDDR5

Die leistungsstarke CPU ermöglicht im Zusammenspiel mit der ebenso leistungsstarken Grafikkarte das Laden und Filtern bzw. Rendern großer Datensätze.

Benutzerinteraktionen können reibungslos bei einer Bildrate von mehr als 20 Bildern pro Sekunde durchgeführt werden, wenn mittelgroße Datensätze geladen werden. Um eine bessere Benutzererfahrung während der Interaktionen zu nutzen, haben wir unser System auch mit einem Lenovo Ideapad A7 Tablet PC als Eingabegerät geprüft. Interaktionen wie Brushing und Hovering können am Tablet PC mit einem Stift leistungsfähiger durch das Skizzieren als durch eine Maus und eine Tastatur durchgeführt werden.

Testnr.	Anzahl Themen	Laderzeit [ms]	RAM [MB]
1.	10	72	45
2.	100	451	64
3.	200	1353	136
4.	300	3354	285
5.	400	7210	533
6.	500	13948	702
7.	600	18830	807
8.	700	23428	915
9.	800	26540	950
10.	1000	30557	1005

Tabelle 4.1: Ladezeit und Speicherbedarf der Themeriver Visualisierung

Es wurde die Zeit gemessen, die zum Laden der Daten nötig war. Tabelle 4.1 gibt die genauen Messwerte an. Das Diagramm aus Abbildung 4.17 zeigt die Abhängigkeit zwischen der Anzahl der dargestellten Themen und den Größen Ladezeit und Hauptspeichernutzung, wobei die Anzahl der Zeitreihen 30 ist.

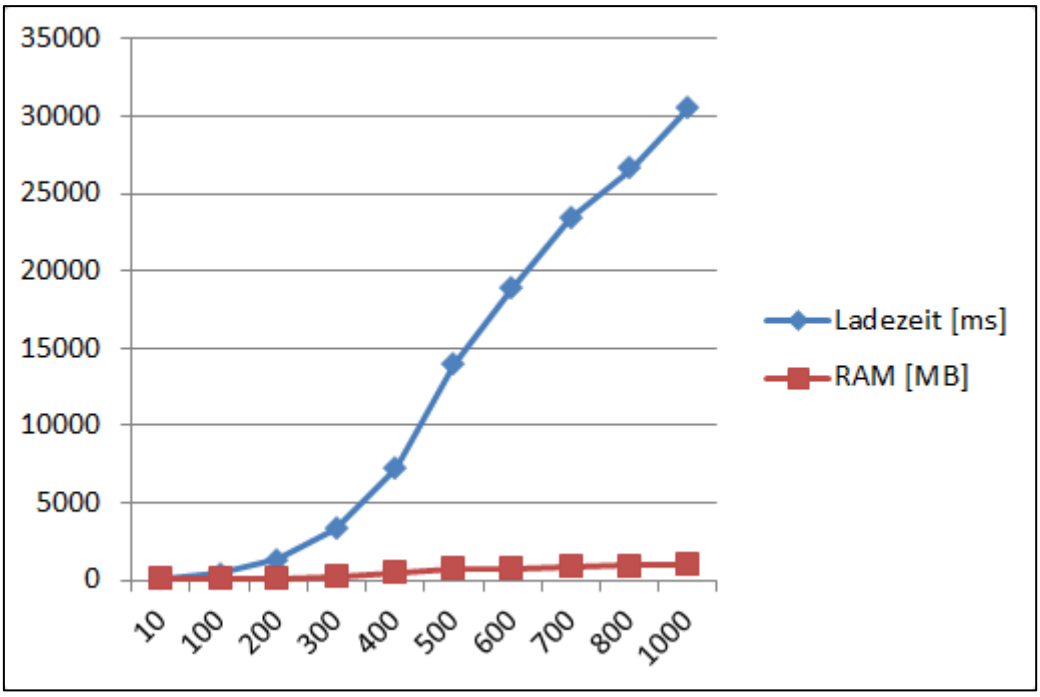


Abbildung 4.17: Ladezeit und genutzter RAM in Abhängigkeit von der Themenanzahl

5. Implementierung

Gemäß den im vorausgegangenen Kapitel 4 angestellten konzeptionellen Überlegungen wird nun die konkrete Implementierung des Visualisierungswerkzeuges beschrieben, wie sie vom Autor im Rahmen dieser Arbeit vorgenommen wurde. Das Visualisierungswerkzeug wird mit der objektorientierten und plattformunabhängigen Programmiersprache Java von Sun Microsystems implementiert. Zunächst wurde die Grundstruktur bei der Entwicklung festgestellt, die die grobe Aufteilung des Programms enthält. Dies resultiert dann einem Klassenentwurf, der alle wichtigen Elemente beinhaltet. Schritt für Schritt wurden die Klassen implementiert. Zuerst wurde ein Prototyp erstellt, der nicht nur den Themeriver mit Curved Links visualisieren konnte, sondern auch noch die Manipulationsmöglichkeiten in Form von interaktiven Features integriert. Der Entwicklungsprozess wurde in verschiedene Phasen untergliedert, in denen Funktionalität ergänzt oder die Visualisierungstechnik verbessert wurde.

In diesem Kapitel wird zuerst die Grundstruktur vorgestellt. Es folgt dann die Klassenbeschreibung, aufgeteilt in die logischen Elemente GUI, Dateninformationen und Zeichnung. Schließlich werden einige der wichtigen Algorithmen im Abschnitt „Ausgewählte Implementierungsdetails“ zur interaktiven Visualisierung erläutert.

5.1 Grundstruktur

Die wichtigsten Teile des Werkzeuges lassen sich durch den Grobentwurf erkennen. Nach dem Modell-Präsentation-Steuerung Muster (MVC Model), das eine Aufteilung in Komponenten beschreibt, wird das Visualisierungswerkzeug entwickelt, in dem Datenmodell-, Präsentations- und Programmsteuerungsklassen vorhanden sind.

Wie bereits im Kapitel 4.5 erwähnt, steht ein Bereich zur Parametereinstellung auf dem Hauptfenster zur Verfügung. Ein weiterer Hauptbereich ist die Darstellungsfläche mit der eigentlichen Visualisierung. Noch ein Bereich muss zur Anzeige der Ausgabeinformationen belegt werden. Damit ergibt sich automatisch die Aufteilung des Layouts in die graphischen Hauptkomponenten Eingabebereich, Darstellungsbereich und Ausgabebereich. Diese Aufteilung untergliedert den Gesamtaufbau in die drei Hauptbereiche Laden von Information, Darstellung der Information und Analyse der Information. Neben dem unbenannten Paket (Default Package) mit den Anwendungsklassen gibt es im Projektverzeichnis drei Pakete namens *data*, *drawing* und *gui*. Das erste Paket enthält die Klassen zur Verarbeitung der Daten. Das zweite besteht aus den Klassen zur Präsentation des Themerivers. Das dritte beinhaltet die Klassen, die sich mit dem Layout der Visualisierungsbereiche befassen, und daher mit der Darstellung von Information zu tun haben. Außerdem existieren Kontrollklassen in Form von MouseListenern, die alle graphischen Oberflächenelemente überwachen und Änderungen an die Datenmodellklassen weiterleiten. Diese geben dann ihre Änderungen von relevanten Daten im Modell bekannt und informieren die Präsentationsklassen, dass eine Änderung geschehen ist und die Visualisierung neu dargestellt werden muss.

Die Realisierung unser Visualisierungstechnik baut auf einer bereits vorhandenen Implementierung dieser Technik von Michael Wohlfahrt und Jürgen Platzer auf, die im Rahmen der Vorlesung und Übung Informationsvisualisierung [WP04] an der Universität Wien entstanden

ist. Dabei werden einige kleine Anpassungen an existierenden Klassen gegeben, während einige Klassen zur Implementierung der interaktiven Funktionen neu erstellt werden müssen.

Die Hauptklasse ThemeRiver erzeugt das Hauptfenster *Mainframe*, welches der Kern des gesamten Programmes darstellt, der alle Elemente verbindet. Nach dem Programmstart wird zuerst das Hauptfenster geöffnet, welches alle graphischen Elemente mit Hilfe der mitgelieferten Pakete AWT und Swing schon enthält. Eine Menüleiste und ein Werkzeugpanel stehen zur Steuerung verschiedener Interaktionsmöglichkeiten am oberen Fensterrand zur Verfügung. Ein Darstellungsbereich, worauf die eigentliche Darstellung der Informationen geschieht, ist ebenfalls bereitgestellt. Bei Bedarf wird nach einem Verzeichnis der Eingabedateien gefragt, die anschließend geladen werden. Die Hauptpräsentationsklasse Mainframe und die Hauptdatenklassen *DataloaderTask* übernehmen diese Aufgabe, die den Text der Dateien einlesen und diese zu den Hauptdatenklassen *StandardData* sendet. Dort wird die gesamte Datenstruktur aufgebaut. Dabei wird eine Instanz der Datenmodellklasse erzeugt.

Nach dem vollständigen Laden des Datensatzes wird die Visualisierung auf dem Anzeigegerät dargestellt. Daher kann die Anwenderinteraktion durch das Abstract Window Toolkit (AWT) durchgeführt werden, andererseits ermöglicht das Swing Toolkit den Aufbau einer graphischen Oberfläche. Somit bauen Buttons, Auswahlboxen und Menüs mit dem Toolkit die Themeriver Oberfläche auf. Das Hauptfenster der Oberfläche ist vom Typ *Mainframe*, welches von *JFrame* abgeleitet wird. Es enthält die Werkzeugleiste *Control*, die Zeichenfläche *Plotter* und die Informationsfläche *Legend*, die beide von *JPanel* abgeleitet werden und Zeichenmethoden der Curved Links bzw. Ausgabetexttafel implementieren.

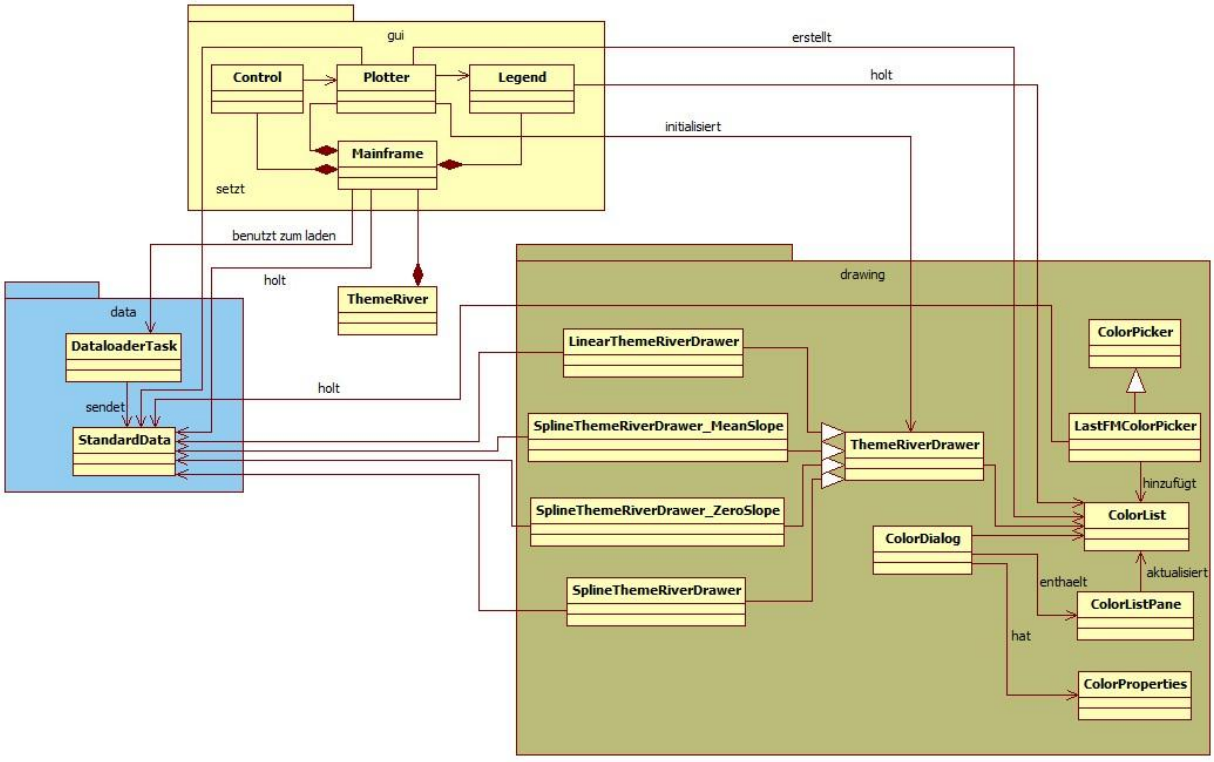


Abbildung 5.1: UML-Klassendiagramm mit den wichtigsten Klassen

In Abbildung 5.1 wird das UML Klassendiagramm des Programms dargestellt (Aufgrund der besseren Übersichtlichkeit werden Methoden und Felder der jeweiligen Klassen nicht angezeigt). Die wichtigsten Klassen verteilen sich auf die drei Pakete *data*, *drawing* und *gui*. Als Grundstruktur ergibt sich innerhalb der Pakete die aus den Kernklassen bestehende mit *StandardData*, *ThemeRiverDrawer* und *Mainframe*, die wiederum aus mehreren einzelnen Elementen bestehen. Alle Klassen werden im folgenden Abschnitt kurz erläutert.

5.2 Klassenbeschreibung

- **ThemeRiver:** Diese Klasse instanziiert ein neues Mainframe-Objekt und zeigt es zentriert auf dem Anzeigegerät.

5.2.1 Die GUI-Klassen

Das Paket *gui* enthält vier Klassen *Mainframe*, *Control* und *Plotter* sowie *Legend*. Sie bauen eine komfortable Schnittstelle mittels grafischer Elemente für den Anwender auf, damit sie gegenseitig kommunizieren können. Die Abbildung 5.2 stellt eine Übersicht über das Paket dar.

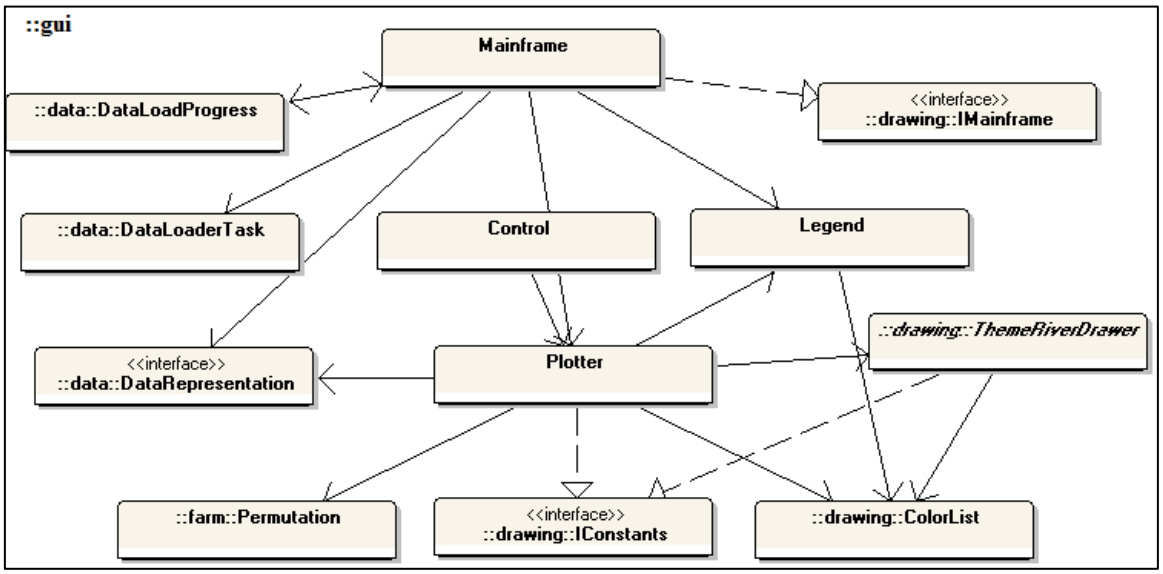


Abbildung 5.2: Übersicht des Pakets *gui*

- **Mainframe:** Die Kernklasse des Hauptfensters wird von der Klasse *JFrame* abgeleitet. Dieses erstellt und handhabt die Funktionalität der Anwendungsschnittstelle, und enthält die Eingabefläche (Klasse *Control*) und Darstellungsfläche (Klasse *Plotter*) sowie die Ausgabenfläche (Klasse *Legend*). Desweiteren hat das Fenster eine Menüleiste. Sie registriert alle Veränderungen der visualisierten Elemente und lässt die betroffenen Teile neu zeichnen. Zusätzlich bietet Sie Möglichkeiten, die aktuelle Visualisierung als Bild gemäß den Exporteinstellungen zu exportieren. Falls die Farbzusammenstellung geändert werden soll, etwa weil einzelne Farben zu ähnlich sind und die Rivers nicht mehr ausreichend unterschieden werden können, ist es über den Menüpunkt „Farbeinstellungen“ möglich, jedem Fluss eine individuelle Farbe zuzuweisen. Diese Klasse benutzt die Klasse *DataLoaderTask* zum Einlesen der Eingabedateien. Dadurch wird die Unterbrechung des Ladeprozesses realisierbar.

- **Control:** Diese Klasse ist abgeleitet von *JPanel* und enthält Texteingabefelder, Schaltflächen und Auswahlboxen, um die interaktive Visualisierung zu manipulieren. An den mit der Komponente gebundenen Listener wird das Ereignis weitergeleitet, damit die für die Visualisierung des Themerivers zuständige Klasse *Plotter* über entsprechende Änderungen an der Darstellung informiert wird.

- **Plotter:** Die interaktive Visualisierung der Information geschieht durch diese Klasse. Sie ist abgeleitet von *JPanel*. Diese setzt den neuen Datensatz *StandardData*. Sie initialisiert deshalb von hier aus anschließend den *ThemeRiverDrawer* gemäß der Repräsentationsart (ThemeRiver, Histogramm sowie Stackedgramm) und der Interpolationsart (durch die Klassen *LinearThemeRiverDrawer*, *SplineThemeRiverDrawer*, *SplineThemeRiverDrawer_MeanSlope* sowie *SplineThemeRiverDrawer_ZeroSlope*) eines Datensatzes, der die Themeriver Visualisierung darstellt. Somit generiert die Rechenmethode hier schon die Kontrollpunkte, welche durch die Splines verlaufen müssen. Die Zeichenmethoden können die horizontale Zeitlinie, Curved Links zweier Flüsse, Datenpunkte und Form eines einzelnen Flusses, sowie die vertikale Linie zur Identifizierung eines am nächsten liegenden Zeitschrittes zeichnen.

- **Legend:** Die Klasse ist ebenso abgeleitet von der Klasse *JPanel* und gibt die textuellen Informationen aus einem aktuellen Fluss ohne Ausblendung der kompletten Visualisierung aus.

5.2.2 Die Datenmodell-Klassen

Der Model-Teil innerhalb der MVC-Architektur unseres Beispiels besteht aus der Datenklasse *StandardData*, Schnittstelle *DataRepresentation* sowie der Service-Klasse *DataLoadProgress* und *DataLoaderTask* (siehe Abbildung 5.3).

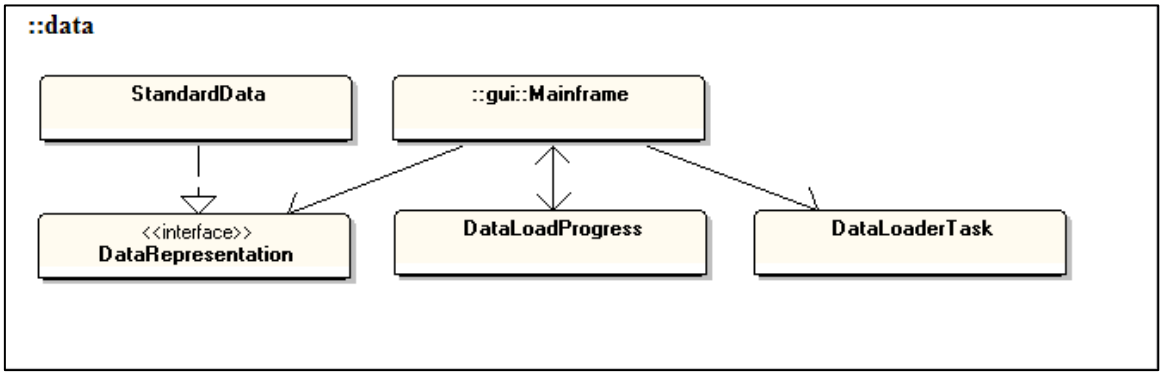


Abbildung 5.3: Übersicht des Pakets *data*

- **DataLoaderTak:** Sie erweitert die Klasse *SwingWorker* und sorgt für die eventuelle Unterbrechung des Ladeprozesses der Dateien. Da das Programm beim Laden eines riesigen Datensatzes „eingefroren“ zu sein scheint, ist das Anzeigen des Ladeprogresses und die Möglichkeit zur Prozessunterbrechung dem Anwender sehr wichtig. Die Lademethode läuft im Hintergrund, um den Text der Dateien einzulesen und gleich danach an die Klasse *StandardData* zu senden.

- **DataLoadProgress:** Sie ist abgeleitet von der Klasse *JDialog* und enthält einen Fortschrittsbalken, eine Beschriftung über die Information der aktuell verarbeiteten Eingabedatei und einen „Abbrechen“-Button. Sobald der Button bestätigt wird, informiert sie die Klasse *DataLoaderTask*, um den Prozess abzubrechen.

- **StandardData:** Diese Klasse dient als Modelklasse, die die Daten zur Visualisierung aus den Eingabedateien während der Programmausführung beinhaltet. Hier werden die Daten als Flussnamen, Zeitschritte und Häufigkeitswerte sowie Relationswerte verstanden. Sie speichert die Onset-Indizes der Flüsse und kann die Indizes der häufigsten (im Sinne als Maximalwert oder Durchschnittwert) aufgetreten Flüsse zurückliefern. Zusätzlich kann mit den Rechenmethoden die Summe numerischer Daten an dem angegebenen Zeitpunkt bzw. der größte Wert der gesamten Flussbreite berechnet werden.

- **DataRepresentation:** Sie ist eine Schnittstelle für die Datenrepräsentation. Alle zur Datenrepräsentation benötigte Methoden für die Klasse, die die Schnittstelle implementiert, werden hier definiert.

5.2.3 Die Präsentation-Klassen

Das größte Paket *drawing* in unserem Programm besteht aus den Klassen, die die Hauptaufgabe zur Präsentation des Themerivers übernehmen (siehe Abbildung 5.4). Von hier aus wird der Themeriver Graph dargestellt. Auf dieser Basis können zusätzliche graphische Teile wie Curved Links darauf gezeichnet werden.

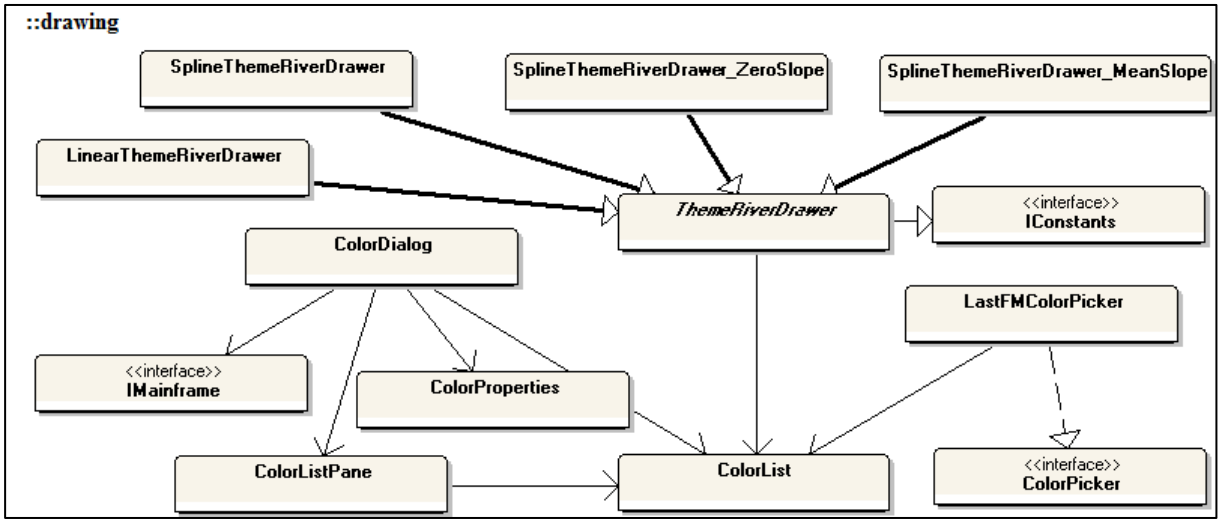


Abbildung 5.4: Übersicht des Pakets *drawing*

- **ThemeRiverDrawer:** Diese Klasse ist eine abstrakte Klasse, zu der keine konkreten Objekte existieren. Sie dient als Oberklasse aller Themeriver Zeichner, die sich durch verschiedene Interpolationsarten unterscheiden. Sie speichert die Positionen der Kontrollpunkte und die Größe der Zeichenflächen sowie die Farben der einzelnen Flüsse und sorgt damit dafür, dass der Themeriver Graph an der korrekten Position und in korrekter Größe gezeichnet wird. Außerdem wird die Abbildung zwischen den einzelnen Flüssen und auf der entsprechenden Zeichenfläche gespeichert, damit die detail-

lierten Informationen des selektierten Flusses von der GUI-Klasse Plotter korrekt ermittelt werden können.

- **LinearThemeRiverDrawer:** Der Themeriver Graph wird vom *LinearThemeRiverDrawer*-Objekt durch lineare Interpolation dargestellt.

- **SplineThemeRiverDrawer:** Der Themeriver Graph wird vom *SplineThemeRiverDrawer*-Objekt durch Spline-Interpolation gezeichnet.

- **SplineThemeRiverDrawer_MeanSlope:** Das *SplineThemeRiverDrawer_MeanSlope*-Objekt realisiert den Themeriver Graph durch die Durchschnitt-Steigung-Spline-Interpolation.

- **SplineThemeRiverDrawer_ZeroSlope:** Das *SplineThemeRiverDrawer_ZeroSlope*-Objekt visualisiert den Themeriver Graph durch die Null-Steigung-Spline-Interpolation.

- **ColorList:** Sie verwaltet die Farbliste für die darzustellenden Flüsse. Hier wird zusätzlich die Abbildung zwischen den einzelnen Flüssen und den entsprechenden Farben gespeichert, damit bei der Darstellung des Themeriver Graph die passenden Farben verwendet werden können.

- **ColorDialog:** Diese Klasse ist von der Klasse *JDialog* abgeleitet. Sie enthält ein *ColorListPane*, es zeigt die aktuell verwendeten Farben untereinander in einer vertikalen Liste an und erlaubt dem Anwender für jeden Fluss entsprechend seines Interesses eine individuelle Farbe zuzuweisen. Außerdem beinhaltet sie *ColorProperties*, um die Veränderung der Farbeinstellungen vorzunehmen.

- **ColorListPane:** Sie ist abgeleitet von der Klasse *JPanel*. Das *ColorListPane*-Objekt kann die aktuell zur Visualisierung des Datensatzes verwendeten Farben als Bild auf dem Panel darstellen und verarbeitet die Farbliste mittels des Ereignisses aus *ColorDialog*.

- **ColorProperties:** Sie ermöglicht die Farbeinstellungen aus einer Datei *colprop.ser* einzulesen bzw. in diese abzuspeichern. Die Datei kann dabei mehrere Farblisten enthalten.

- **CorlorPicker:** Sie ist eine Schnittstelle für neue Färbungsalgorithmen, in der festgelegt wird, über welche Methoden die Klassen, die das Interface implementieren, verfügen müssen.

- **LastFMColorPicker:** Diese Klasse implementiert die Schnittstelle *ColorPicker*. Sie lädt eine zweidimensionale Bilddatei und jedes Pixel auf dem Bild hat zwei Koordinatenwerte, aus denen man den RGB Farbton berechnen kann. Der x-Koordinatenwert wird mittels eines Onset-Index herausgefunden, während man den y-Koordinatenwert durch den Beliebtheit-Index (im Sinne der Häufigkeit) des zu zeichnenden Flusses gewinnt. Somit wird dem Fluss die Farbe an diesem Punkt zugewiesen.

5.2.4 Die Hilfsklassen

- **PairKey:** Diese Klasse kann beispielsweise ein Objekt der Klasse A mit einem Objekt der Klasse B verknüpfen, damit enthält man ein einziges Objekt. Dies hilft beim Aufbau der Container-Klasse *Map*, deren Schlüssel dabei früher lediglich ein Objekt einer (fast) beliebigen Klasse darstellte.

- **IndexValue:** Sie erlaubt die Elemente einer Liste zu sortieren. Dabei können die originalen Indizes noch beibehalten werden.

- **Permutation:** Das *Permutation*-Objekt zählt alle möglichen Permutationen von n Elementen auf. Es kommt beim Algorithmus zur Kantenlängenminimierung zum Einsatz. Hier wird die Reihenfolge der Knoten durch die Permutation der Flussfolgen bestimmt, die überprüft werden müssten.

5.3 Ausgewählte Implementierungsdatails

5.3.1 Berechnung des optimalen Darstellungsintervalles der Kanten

Wie es bereits im Kapitel 4.3 erwähnt wurde, soll ein optimales Darstellungsintervall der gerichteten Curved Links anhand der dynamischen Abhängigkeiten herausgefunden werden, um zusätzlich die dynamischen Relationen zwischen den Flüssen über die Zeit zusätzlich zu visualisieren. Dazu muss man beachten, das Visual Clutter Problem zu reduzieren. Wenn die Kanten überhaupt an jedem Zeitpunkt (falls das Gewicht > 0) gezeichnet werden, dann wird der dahinter liegende Themeriver Graph ausgeblendet.

Ein erster naiver Ansatz des Problems ist, einen Kantenfilter zu verwenden. Falls die gesamten Kantengewichte erst den vom Anwender eingegebenen Schwellwert überschreiten, dann wird die Kante an diesem Zeitpunkt dargestellt.

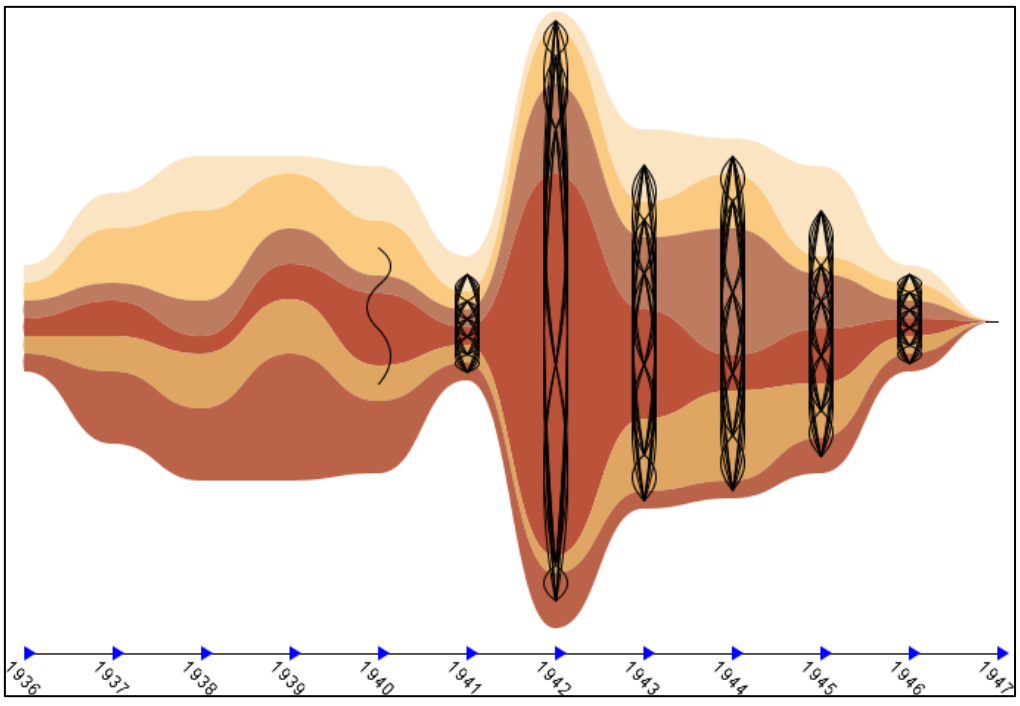


Abbildung 5.5: Naiver Ansatz für gesamte Kantengewicht >= 5

Dieser naive Ansatz hat allerdings einen entscheidenden Nachteil. Wie in Abbildung 5.5 dargestellt, haben die Flüsse erst im Jahr 1940 ausreichende Relationen, d.h. ab diesem Zeitpunkt wird eine Kante zwischen zwei Flüsse gezeichnet. Nach einem Jahr muss noch mal die Kante gezeichnet werden, da sich die Kantengewichte inzwischen vergrößert haben. So entstehen vier weitere Kanten, so dass wir es wieder mit einem dichten Graph zu tun haben.

In der Implementierung werden die Relationsdaten mit Hilfe der Klasse *PairKey* und der Methode *drawRelation()* im Format ((*indexXInMatrix*, *indexYInMatrix*), *Relationswert*) als (*Key*, *Value*)-Paar in der Map Datenstruktur gespeichert, da die Matrix gegebenenfalls viele Nullen enthält. Hier dient der Schlüssel als zwei Knoten Paar. Wir suchen alle möglichen Schlüssel (Position in der Matrix) in der kompletten Zeitspanne aus und müssen für jeden Schlüssel den entsprechenden Relationswert von jeder Zeitspanne aufsummieren. Sobald der Schwellwert erreicht ist, wird an der Stelle zwischen zwei Knoten eine Kante gezeichnet. Das gesamte Kantengewicht wird beibehalten und zählt für die nächste eventuell darzustellende Kante mit. Das heißt, wenn zwischen den zwei Flüssen im nächsten Zeitschritt noch ein Relationswert vorhanden ist, so entsteht eine neue Kante.

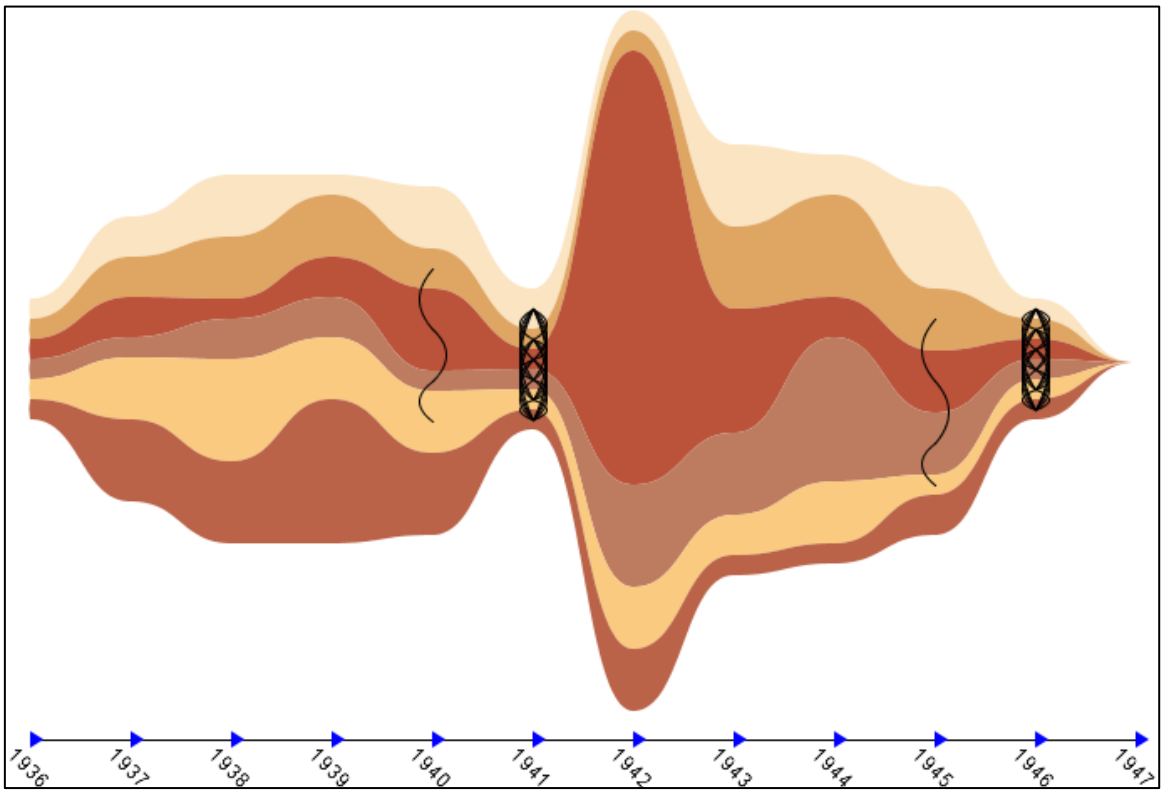


Abbildung 5.6: Verbesserter Ansatz für gesamte Kantengewicht >= 5

Wir können den Algorithmus so umsetzen, dass ein dichter Graph vermieden werden kann. Nach der Erreichung des Schwellwertes wird das Gesamtkantengewicht nicht beibehalten, sondern auf Null setzen. Dies zählt für die nächste eventuell darzustellende Kante nicht mehr mit. So entsteht ein Graph mit wenigeren Kanten, wie es in Abbildung 5.6 dargestellt wird. Die Kanten werden nach einer bestimmten Zeitspanne wieder erscheinen, so dass der Graph immer noch anschaulich betrachtet werden kann.

6. Fallstudie

In diesem Kapitel studieren wir einen Anwendungsbereich für unser erweitertes Themeriver System mit zusätzlichen dynamischen Relationen, dargestellt als Curved Arcs wie in der TimeArcTrees Technik. Abbildung 6.1 stellt eine Visualisierung der Datenerfassung in der DBLP (Informatik-Bibliografie) [Ley09] dar, welche eine bibliographische Datenbank von Publikationen hauptsächlich im Bereich Informatik ist. Der Datensatz wurde aus den 2,089,529 Publikationstiteln zwischen den Jahren 1936 und 2012 extrahiert. Dieser enthält die Tausend am häufigsten auftretenden Wörter und ihre Vorkommenszahlen sowie ihre dynamischen Relationsdaten. Jeder Publikationstitel besteht aus einer Menge an Wörtern. Zwei Wörter sind verwandt, wenn sie im gleichen Publikationstitel auftreten. Das Gewicht 1 wird jedes Mal bei einem gemeinsamen Vorkommen beider Wörter einer Kante, d.h. Relation, zugewiesen.

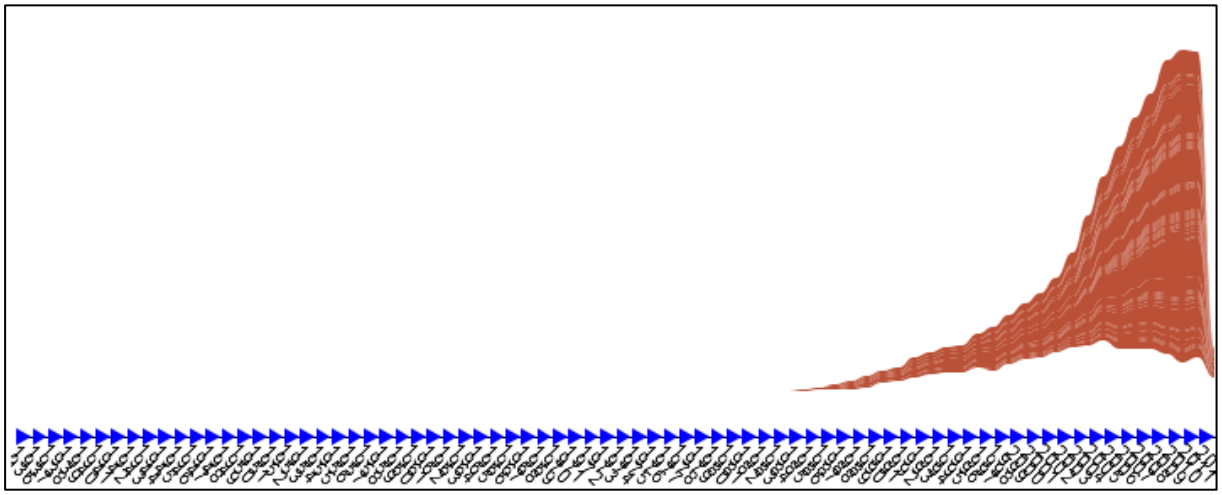


Abbildung 6.1: Startansicht - erweiterte Themeriver Visualisierungstechnik wird auf den DBLP Datensatz mit insgesamt Wörtern in 77 Graphen angewendet.

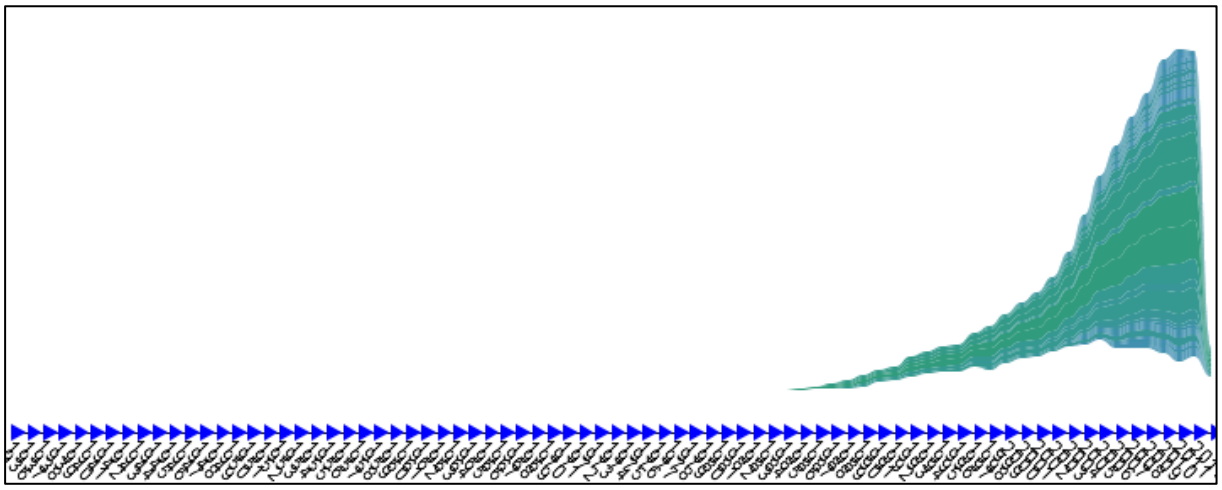


Abbildung 6.2: Ansicht nach neuer Anordnung der Flüsse

In der Startansicht bekommt man schon eine grundlegende Übersicht über den Graph. Per Mouseover erfährt der Anwender, dass es sich um einen Datensatz mit explosionsartigem Charakter handelt.

Hierfür werden die Themenflüsse des Graphs durch die „Onset“-Option umsortiert und mit einem neuen Layout und entsprechenden Farben wie in Abbildung 6.2 dargestellt. Da der Datensatz ziemlich groß ist, wird hier eine kleine Menge an Themenflüssen betrachtet, indem die 40 am häufigsten auftretenden Wörter aus den 1000 gegebenen durch den Flussfilter herausgefiltert wurden. Dabei werden die Füllwörter wie „a“, „and“, „or“, „it“, „of“ usw. von der Wörterliste befreit (siehe Abbildung 6.3).

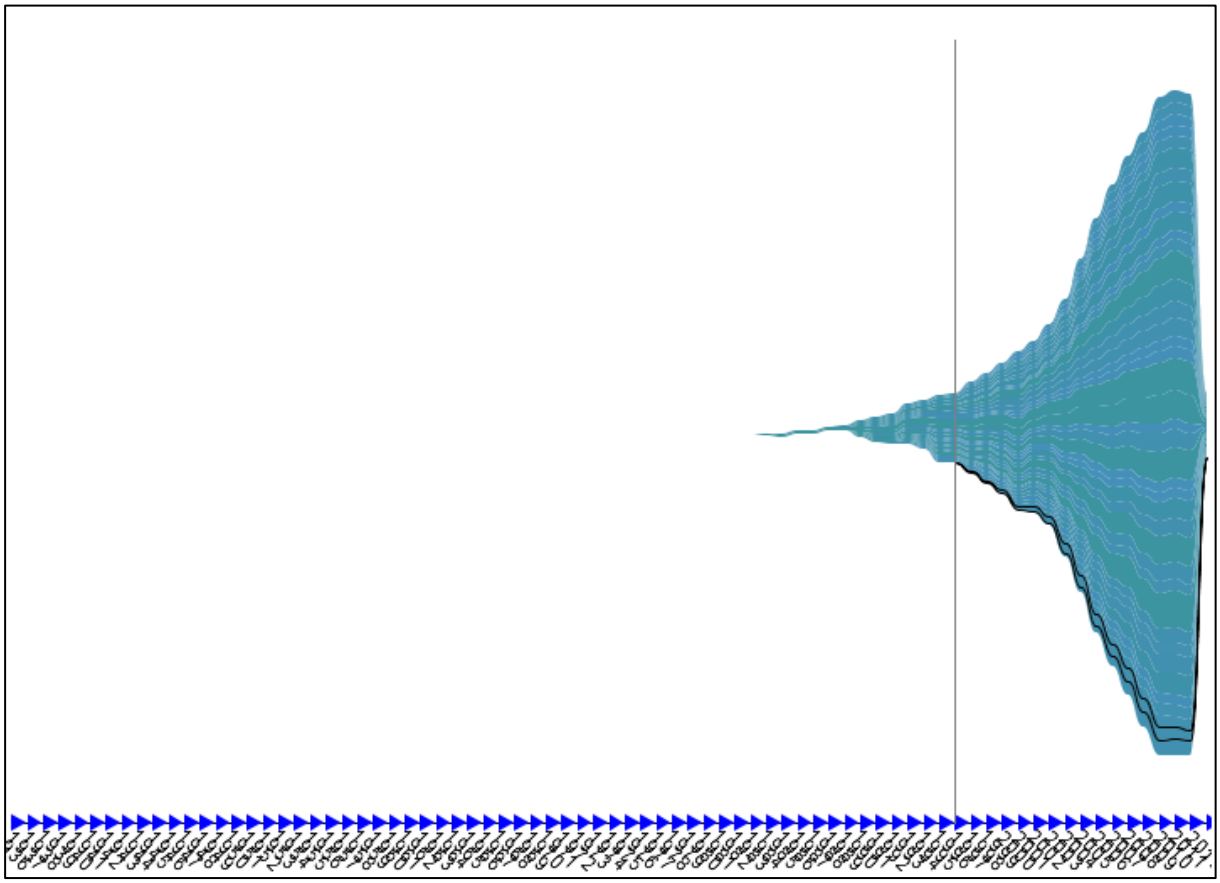


Abbildung 6.3: Der Graph wurde aus den Wörtern der Publikationstitel ohne die Füllwörter generiert, die 40 Wörter inklusiv „Wireless“ und „Web“ enthalten. 77 Graphen sind für die Jahre 1936 bis 2012 dargestellt. Der selektierte Bereich zeigt, dass das Wort „Web“ erst im Jahr 1996 vorkommt.

Nach dem Durchsehen der 40 Themenflüsse sind wir jetzt daran interessiert, wie die Beziehung zwischen den zwei Wörtern „Web“ und „Wireless“ und wie sich alle lexikalisch semantischen Beziehungen zu einem oder beiden dieser Wörter entwickeln. Das Auftreten des Wortes „Web“ beginnt erst im Jahr 1996. Nach 4 Jahren kommt dann das Wort „Wireless“ häufiger vor. Um das Visual Clutter Problem weiter zu reduzieren, konzentrieren wir uns auf die Graphen des Jahres 1996 bis 2012 mit Hilfe des Datensatzfilters in Abbildung 6.4. Jetzt können 17 Graphen in größeren Skalen dargestellt werden und erlauben eine detailliertere Sicht für den Benutzer. Desweiteren filtern wir alle Kanten heraus, deren Gewichtswert kleiner als

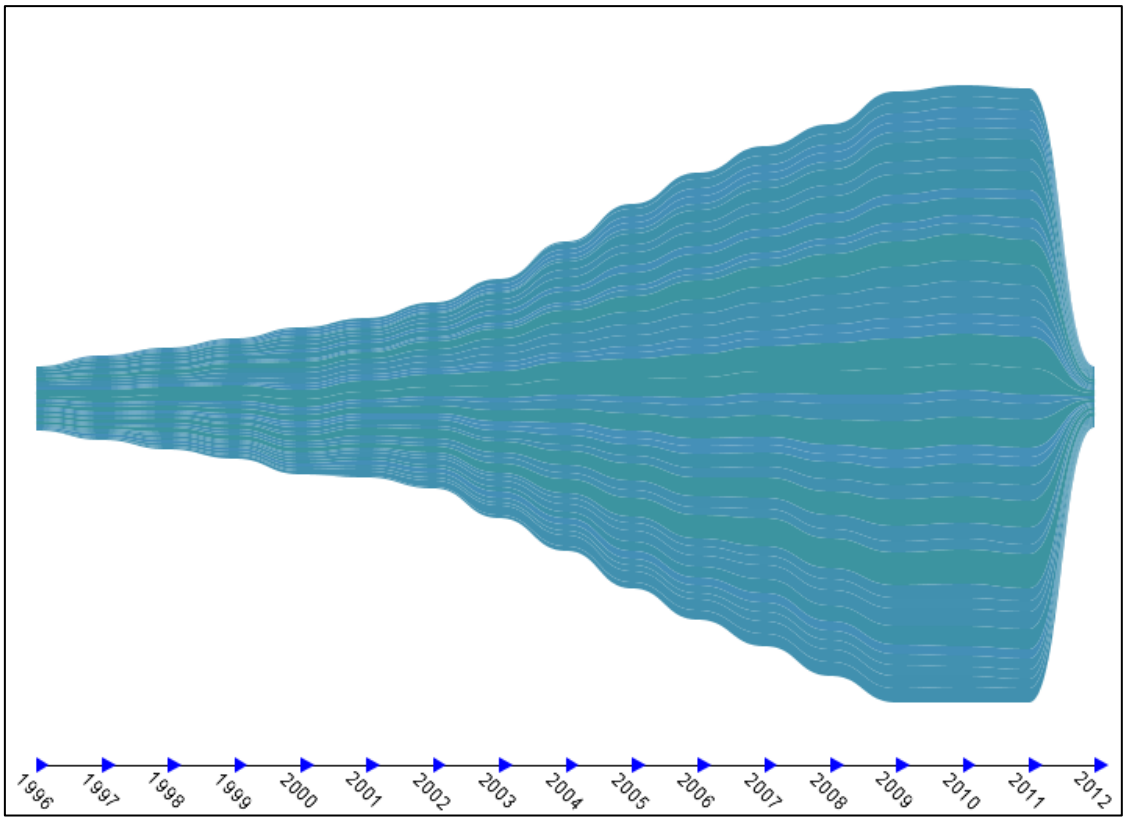


Abbildung 6.4: 17 Graphen sind für die Jahre 1996 bis 2012 nach der Filterung dargestellt.

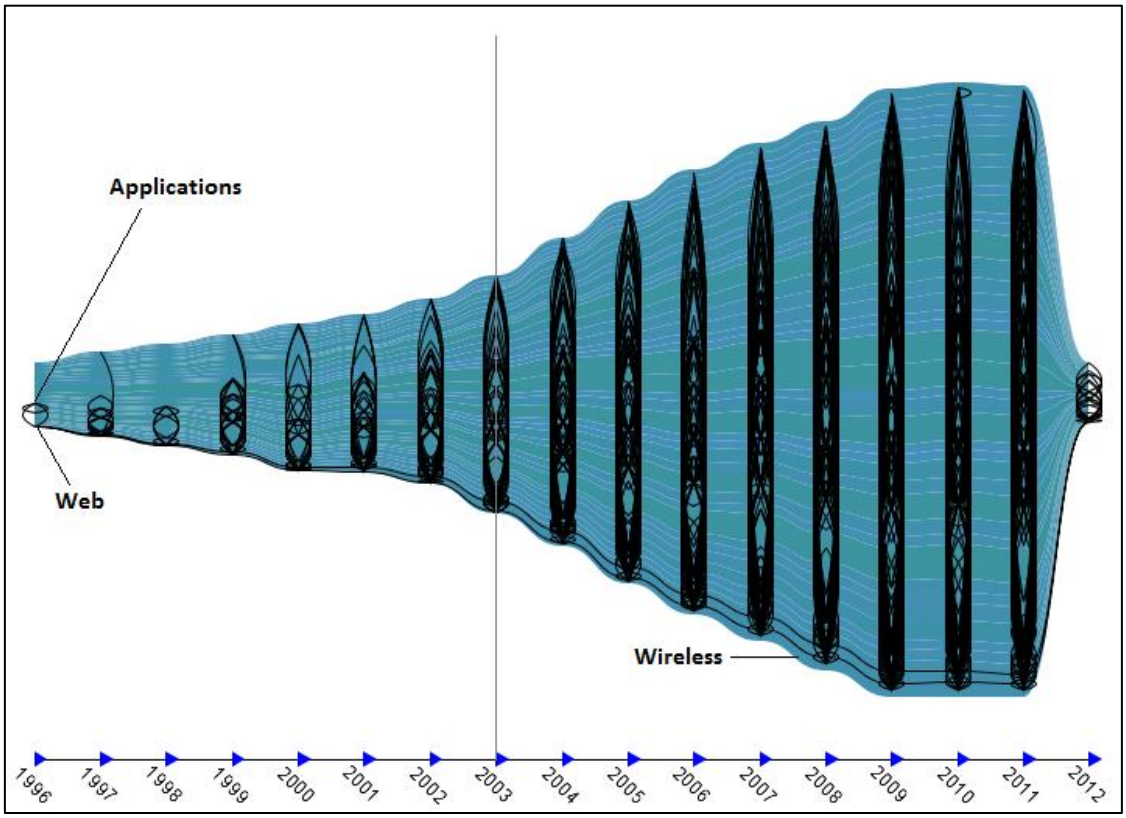


Abbildung 6.5: Kantenfilterung mit einem Gewicht 3500 in 17 Graphen: Die Korrelation zwischen „Web“ und „Applications“ kommt schon im Jahr 1996 vor, wobei das Wort „Wireless“ erst im Jahr 2000 auftritt.

3500 ist. Abbildung 6.5 zeigt das Ergebnis nach der Kantenfilterung. Das Wort „Web“ hat die erste Korrelation mit dem Wort „Applications“ im Jahr 1996 und kommt in den nächsten Jahren sehr häufig im Zusammenhang mit diesem anstatt mit dem Wort „Wireless“ vor. Da das Wort „Wireless“ erst im Jahr 2000 erscheint, filtern wir den Teil des Datensatzes zwischen den Jahren 1996 bis 1999 heraus. Wir betrachten jetzt die Kanten, denen das minimale Kantengewicht 3180 zugewiesen sind. Abbildung 6.6 zeigt das Ergebnis, dass das Wort „Wireless“ die erste Korrelation mit dem Wort „Sensor“ im Jahr 2000 hat und stark mit diesem anstatt mit dem Wort „Web“ gekoppelt ist. Noch mehr Einsichten können durch die interaktiven Features des Visualisierungswergzeugs erlangt werden. Weitere Ergebnisse zu präsentieren, würden jedoch den Rahmen dieser Arbeit sprengen. Deshalb dient diese Case Study nur als Repräsentant für die vielen Möglichkeiten der Einsichtsgewinnung, die ein Benutzer der neuen Visualisierungstechnik hat.

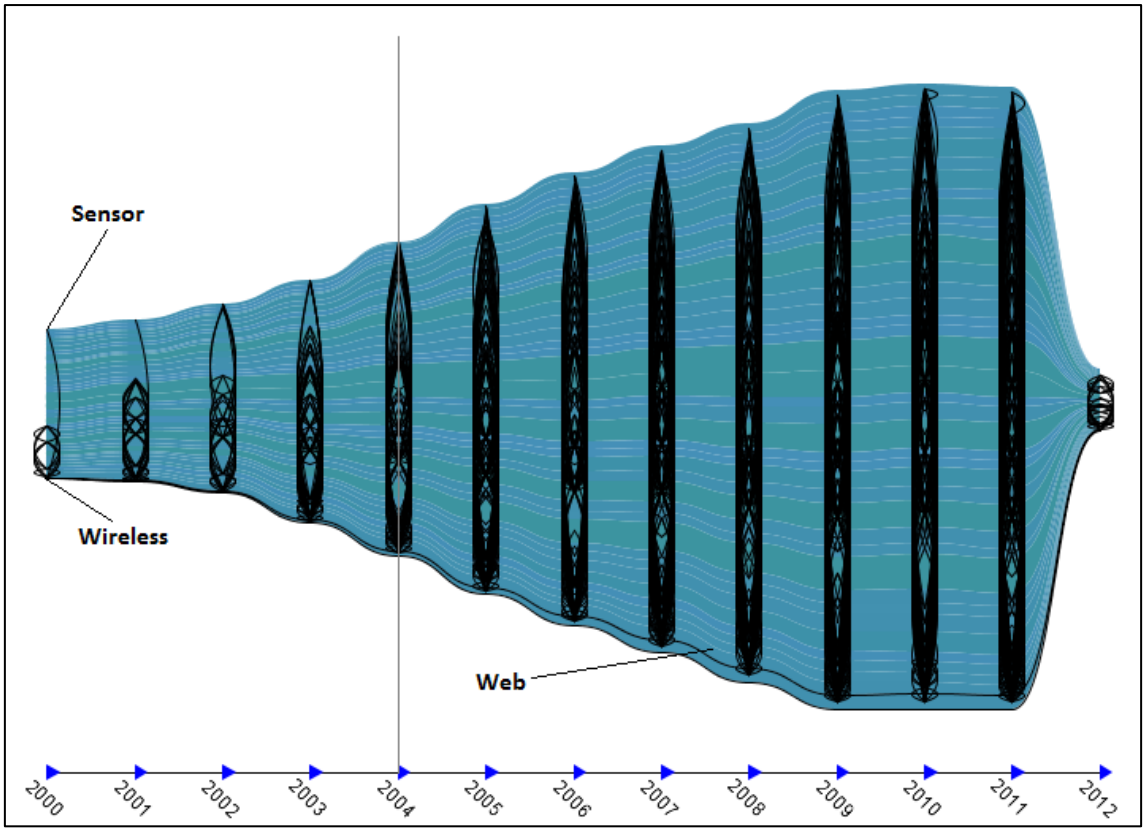


Abbildung 6.6: Kantenfilterung mit einem Gewicht von 3800 in 12 Graphen: Die Korrelation zwischen „Wireless“ und „Sensor“ tritt schon im Jahr 2000 auf, wobei das Wort „Wireless“ gerade im Jahr 2000 erscheint.

7. Zusammenfassung und Ausblick

Ziel der vorliegenden Arbeit war die konzeptionelle Entwicklung eines Visualisierungswerkzeuges, wie es in einer Trend-Analyse eingesetzt werden könnte unter Erweiterung des Themeriver Graphen mit zusätzlichen dynamischen Relationen in der Form von Knoten-Kanten-Diagrammen. Dazu wurden Grundkenntnisse der menschlichen Wahrnehmung dargestellt, wie man bestmögliche Visualisierungen erstellen kann. Anschließend wurden verwandten Arbeiten untersucht, die sich mit ähnlichen Problemstellungen beschäftigen. Dies waren konkret die Techniken *Themeriver System*, *NameVoyager*, *Many Eyes System*, *StreamGraph*, *Kanshin*, *BlogPulse*, *FolkRank*, *TimeMines*, *ZTree*, *TreeMaps*, *Overlaying Graph Links on TreeMaps*, *Trees in a Treemap*, *ArcTrees*, *TimeArcTrees*, *TimeRadarTrees*, *Timeline Trees*, *Parallel Edge Splatting* und *Perspective Wall*.

Ein Visualisierungswerkzeug wurde nun aufgrund der gewonnenen Erkenntnisse konzipiert, dass es ermöglicht, einen dynamischen gerichteten und gewichteten Graph zusätzlich in die Themeriver Standarddarstellung zu integrieren. Die Hauptaspekte waren dabei eine gute Lesbarkeit des Graphen und möglichst viel Interaktivität in Hinsicht auf die Wünsche des Anwenders.

Gemäß dem erstellten Konzept wurde eine Implementierung mit Java erzeugt, die die Visualisierung der zeitbasierten Daten zusammen mit dem relationalen Verhalten unter den Werten realisiert. In einer abschließenden Fallstudie wurde ein Anwendungsfall des Werkzeuges mithilfe eines Datensatzes präsentiert. Die entstandenen Abbildungen dienen dabei als Beispiele für Anwendungen des erweiterten Themeriver Visualisierungswerkzeuges. Die Ergebnisse zeigen, dass die Visualisierungstechnik die in der Einleitung aufgestellten Anforderungen erfüllt.

Bereits während Konzeption, als auch Umsetzung, wurden einige Aspekte deutlich, die Potential für Verbesserungen oder Weiterentwicklungen bieten. Bisher wurden die Graphdaten ständig in den Arbeitsspeicher gelagert. Das Werkzeug arbeitet effizient, solange die darzustellende Informationsmenge die Kapazität des Hauptspeichers nicht überschreitet. Da die Relationsdaten der Eingabedatei die Form einer quadratischen Matrix haben, die viele Nulleinträge enthält, wird relativ viel Speicherplatz bei der Datenhaltung verbraucht. Momentan werden Relationsdaten im Format ((*indexXInMatrix, indexYInMatrix*), Relations-wert) als (*Key*, *Value*)-Paar in der Map-Datenstruktur gespeichert. Es ist denkbar, sie als dünnbesetzte Matrizen zu interpretieren und effizienter im Speicher zu halten.

Eine Labelinformation könnte auf dem Graph zusätzlich angezeigt werden, damit der Anwender die einzelnen Flüsse besser zuordnen kann. Dazu muss ein neuer Algorithmus entworfen werden, der optimale Positionierungen der Labelinformationen bestimmt.

Es könnten Methoden implementiert werden, um Flüsse so zu verschieben, dass der Anwender interessante Flüsse zusammenlegen und direkt vergleichen kann.

Bisher wurde ein Algorithmus zur Berechnung der minimalen Gesamtkantenlängen verwendet. Dieser verhält sich allerdings nur optimal, wenn die Anzahl der darzustellenden Flüsse kleiner als 10 ist. Da das allgemeine algorithmische Problem NP-hart ist, müsste ein heuristi-

sches Verfahren entworfen werden, um eine gute Lösung für die Flussordung bezüglich der Gesamtkantenlängenminierung zu finden.

Die vorliegende Arbeit hat auf dem Gebiet Themeriver mit dynamischen Relationen eine Basis geschaffen. Jetzt können weitere Arbeiten darauf aufbauen, weiter verbessern und erweitern.

Abbildungsverzeichnis

Literaturverzeichnis

[Aig06] Aigner, W.: Visualization of Time and Time-Oriented Information: Challenges and Conceptual Design, Vienna University of Technology, Institute of Software Technology and Interactive Systems, PhD Thesis (2006).

[AMSH11] Aigner, W., Miksch, S., Schumann, H., Tominski, C.: Visualization of Time-Oriented Data. Springer, London (2011)

[BBD08] Burch, M., Beck, F., Diehl, S.: Timeline trees: visualizing sequences of transactions in information hierarchies. In AVI, pages 75-82. ACM Press (2008).

[BBD09] BECK F., BURCH M., DIEHL S.: Towards an aesthetic dimensions framework for dynamic graph visualisations. In Information Visualisation, 13th International Conference, pp. 592–597 (2009).

[BD06] Burch, M. and Diehl, S.: Trees in a treemap: Visualizing multiple hierarchies Proceedings of SPIE, 6060: pp. 224–235 (2006).

[BD08] Burch, M. and Diehl, S.: Timeradartrees: Visualizing dynamic compound digraphs. Comput. Graph. Forum, 27(3): 823-830 (2008).

[BDT07] Borland, D., Taylor, R.M.: Rainbow Map (Still) Considered Harmful. IEEE Comp. Graphics and Applications, 27 (2): 14-17 (2007).

[BUC00] Bartram, L., Uhl, A., Calvert, T.: Navigating complex information with the ztree. In Graphics Interface, pages 11-18. Canadian Human-Computer Communications Society (2000).

[BVBDV11] Burch, M., Vehlow, C., Beck, F., Diehl, S., Weiskopf, D.: Parallel edge splatting for scalable dynamic graph visualization. IEEE Transactions on Visualization and Computer Graphics, 17 (12):2344–2353 (2011).

[BW08] Byron, L. and Wattenberg, M.: Stacked Graphs - Geometry & Aesthetics. In: IEEE Trans. Vis. Comput. Graph. 14, Nr. 6, S. 1245-1252 (2008).

[Dah06] Dahm, M.: Grundlagen der Mensch-Computer-Interaktion. Pearson Studium (2006).

[DFN02] Daassi, C., Fauvet, M., Nigay, L.: Multiple Visual Representation of Temporal Data. Proceedings of the 13th international Conference on Database and Expert Systems Applications, pp. 701-709, 02. -06 September, 2002.

[Fuk05] Fukuhara, T.: Analyzing concerns of people using Weblog articles and real world temporal data. In: Proceedings of the 14th International Conference on World Wide Web - WWW 2005 (2005).

[FWDAP03] Fekete, J.-D., Wand, D., Dang, N., Aris, A., Plaisant, C.: Overlaying graph links on treemaps. In Poster Compendium of the IEEE Symposium on Information Visualization (INFOVIS'03), Los Alamitos, CA, USA. IEEE (2003).

[GBD09] Greilich, M., Burch and M., Diehl, S.: Visualizing the evolution of compound digraphs with TimeArcTrees. Computer Graphics Forum 28, 3 (2009), 975–982.

[Geg12] Gegenfurtner, K.G.: Farbwahrnehmung. Website, 01.09.2012. URL: http://www.allpsych.uni-giessen.de/karl/teach/farbe.html.

[GHT04] Glance, N., Hurst, M. and Tomokiyo, T.: Blogpulse: Automated trend discovery for weblogs. In: WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics. vol. 2004. Citeseer (2004)

[HHNW02] Havre,S., Hetzler, B., Nowell, L., Whitney, P.: Themeriver: Visualizing thema- tic changes in large document collections. Transactions on Visualization and Computer Graphics (2002).

[HJSS06] Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Trend detection in folksonomies. In: Avrithis, Y.S., Kompatsiaris, Y., Staab, S., O'Connor, N.E. (eds.) First International Conference on Semantics and Digital Media Technology (SAMT), pages 56-70. Springer (2006)

[Jan06] Jansen, D.: Einführung in die Netzwerkanalyse. Grundlagen, Methoden, Forschungsbeispiele. 3., überarbeitete Auflage. Wiesbaden (2006).

[Kei02] Daniel A. Keim.: Datenvisualisierung und data mining. Datenbank-Spektrum, 2 (2002).

[Kel06] Keller, R., Eckert, C.M. and Clarkson, P.J.: Matrices or node-link diagrams: which visual representation is better for visualising connectivity models? Information Visualization, 5: 62–76, (2006).

[Ley09] Ley, M.: DBLP - Some lessons learned. Proceedings of Very Large Data Bases, 2(2): 1493-1500, (2009).

[LV03] Lyman, P., Varian, H.R.: How Much Information. University of California at Berkeley (2003). URL: http://www2.sims.berkeley.edu/research/projects-/how-much-info-2003.

[NSC05] Neumann, P., Schlechtweg, S., and Carpendale, M. S. T.: Arctrees: Visualizing relations in hierarchical data. In EuroVis, pages 53-60. Eurographics Association (2005).

[Pla86] Playfair, W.: Commercial and Political Atlas and Statistical Breviary (1786).

[RM05] Rosenholtz, R. and Mansfield, J.: Feature congestion: a measure of display clutter. ACM SIGCHI Conference on Human Factors in Computing Systems. pp. 761-770, 2005.

[Shn96] Shneiderman, B.: The eye have it: A task by data type taxonomy for information visualizations. In Visual Languages, (1996).

[SJ00] Swan, R. and Jensen, D.: TimeMines: Constructing Timelines with Statistical Models of Word Usage. In The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2000).

[SM00] Schumann, S., Müller, W.: Visualisierung – Grundlagen und allgemeine Methoden. Springer-Verlag, Berlin Heidelberg (2000).

[Son08] Sonja, Ö.: Visualisierungs- und Interaktionsdesign für multivariate, zeitabhängige Daten in sozialen Netzwerken. University of Konstanz. Diploma Thesis (2008).

[VKSKVFF11] von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., van Wijk, J., Fekete, J.-D. and Fellner, D.: Visual analysis of large graphs: State-of-the-art and future research challenges. Computer Graphics Forum, 30(6):1719–1749, 2011.

[VWVKM07] Viégas, F.B., Wattenberg, M., van Ham, F., Kriss, J., & McKeon, M. Many Eyes: A Site for Visualization at Internet Scale. In Proc. of IEEE InfoVis (2007).

[Wat05] Wattenberg, M.: Baby Names, Visualization, and Social Data Analysis. Proceedings of the IEEE Symposium on Information Visualization (2005).

[WK06] Wattenberg, M., Kriss, J.: Designing for Social Data Analysis. IEEE Transactions on Visualisation and Computer Graphics. 12(4) (2006).

[WP04] Wohlfahrt, M., Platzer, J.: ThemeRiver. Wien: Technische Universität Wien, Vorlesung und Übung Informationsvisualisierung (2004). URL: http://www.cg.tuwien.ac.at/courses/InfoVis/HallOfFame/2004/05_ThemeRiver/

Erklärung

Hiermit versichere ich, diese Arbeit selbstständig verfasst und nur die angegebenen Quellen benutzt zu haben.

_____

(Qi Hu)