

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart  
Universitätsstraße 38  
D-70569 Stuttgart

Bachelorarbeit Nr. 85

# **Visual Analytics zur Auswertung von Eyetracking und Mousemovement Daten**

Daniel Hertl

<b>Studiengang:</b>	Softwaretechnik
<b>Prüfer/in:</b>	Prof. Dr. Thomas Ertl
<b>Betreuer/in:</b>	Dipl.-Inf. Dennis Thom Dipl.-Phys. Michael Raschke
<b>Beginn am:</b>	23. Juli 2013
<b>Beendet am:</b>	23. Januar 2014
<b>CR-Nummer:</b>	H.2.8 H.5.1 H.5.4





## **Kurzfassung**

Das Internet ist eines der einflussreichsten Medien der heutigen Zeit. Dabei tritt immer mehr die Usability der Webseiten in den Vordergrund. In Forschungen sollen Eyetracking-Experimente dabei helfen, das Verhalten von Benutzern auf Weboberflächen zu untersuchen. Diese Eyetracking Systeme sind jedoch sehr kostspielig. Um eine Investition in diese Systeme umgehen zu können, soll eine andere Möglichkeit den Eyetracker ersetzen.

Das Aufzeichnen der Mousemovements könnte eine solche Alternative sein. Begleitend zu dieser Arbeit wurden deshalb in einer Studie beide Bewegungen parallel erfasst. Gegenstand dieser Ausarbeitung ist es, ein Visual Analytics Ansatz zu finden, der es möglich macht diese Daten in bestimmten Bereichen auf Korrelation der Augen- und Mausbewegungen zu untersuchen. Dabei steht die Filterung, statische Auswertung und zeitliche Exploration im Vordergrund.

## **Abstract**

The internet is one of the most powerful media in this day and age. Though the usability more and more come to the fore. In researches eyetracking experiments shall help to explore the behavior on websurfaces. However these eyetracking systems are very expensive. In order to be able to avoid this investment another eventuality shall replace the eyetracker.

The capture of mousemovements could be an alternative. Therefore attendant to this assignment both movements have parallel captured. Subject of this elaboration is to develop a visual analytics approach that enables these data to research eye and mousemovement data in correlation. Thereby the action of filtering, static analysis and temporal exploration are in the focus.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>9</b>
<b>2</b>	<b>Grundlagen</b>	<b>11</b>
2.1	Visual Analytics . . . . .	11
2.2	Eyetracking Visualisierung . . . . .	16
2.3	Technische Grundlagen . . . . .	20
<b>3</b>	<b>Verwandte Arbeiten</b>	<b>23</b>
3.1	Visualisierung von Maus- und Augebewegungen im Internet . . . . .	23
<b>4</b>	<b>Lösungskonzept</b>	<b>29</b>
4.1	Datenerfassung . . . . .	30
4.2	Vorüberlegungen zur Visualisierung . . . . .	32
4.3	Datentransformation . . . . .	36
4.4	Finaler VA Ansatz . . . . .	39
<b>5</b>	<b>Implementierung</b>	<b>47</b>
5.1	Architektur des Grundsystems . . . . .	47
5.2	Realisierung der VA . . . . .	49
<b>6</b>	<b>Fallstudie</b>	<b>57</b>
6.1	Hypothesenuntersuchung . . . . .	57
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>63</b>
	<b>Literaturverzeichnis</b>	<b>65</b>

# Abbildungsverzeichnis

---

2.1	Teildisziplinen in der Visual Analytics. Die Endgültige Visualisierung ist abhängig von jedem Teilgebiet [KKEM10]. . . . .	12
2.2	"Parallel Coordinates" bei Nährstoffwerten von Nahrungsmitteln. Es wurde ein Bereich von 30-70 g Kohlenhydrate ausgewählt und die Visualisierung zeigt alle Lebensmittel, die für diese Auswahl in Frage kommen.[Exa13] . . . . .	13
2.3	Abstrakter Daten-Visualisierungsprozess [KKEM10] . . . . .	15
2.4	Ein Scanpath, der Fixationspunkte der Augen darstellt. Die Kreisgröße an den einzelnen Punkten beschreibt die Aufenthaltsdauer der Augen an diesem Punkt[LN12] . . . . .	17
2.5	Beispiel für Visual Clutter [HW09] . . . . .	19
2.6	Beispiel für Edge Bundling[HW09] . . . . .	19
3.1	Oben ist die Fixations-Heatmap der Augen und unten die Heatmap der Mausposition zu sehen. Anfangs ist die Position der Werbung rechts oben fixiert. Dann wird die Position zufällig über 6 Positionen variiert. Dabei ändert sich die Position von Auge und Maus von fokussiert oben links, zu Mitte der Seite [NC12] . . . . .	24
3.2	Einfluss der Positionierung von Werbung auf das Muster der Augen- und Mausbewegungen. Erscheint die Werbung oben auf der Seite werden mehr Fixationspunkte der Augen gezeichnet als bei Positionierung der Werbung in der Mitte oder am Ende der Seite. Erscheint die Werbung rechts werden mehr Fixationspunkte gezeichnet als bei Positionierung der Werbung links auf der Seite.[NC12] . . . . .	25
3.3	Example of the mouse following the eye in the vertical direction. Plotting the Y coordinate against time, in the visualization on the right, makes clear a pattern that is hidden in the visualization on the left.[RFASo8] . . . . .	26
4.1	Abbildung, die den Datenbeschaffungsprozess verdeutlichen soll . . . . .	30
4.2	Visualisierungsansatz von Mausbewegungen als Pfad. Entfernungen der zeitlich zugehörigen Augenpositionen sind in bestimmten Farben dargestellt . . .	34
4.3	Übersicht der Weboberflächenbasierten Visualisierung. . . . .	35
4.4	Bildpunkte nach dem Scrollen entsprechen nicht der Realität . . . . .	38
4.5	Transformationsskript arbeitet mit den aufgezeichneten Mausdaten aus Datenbank . . . . .	39
4.6	Der Idealfall. Zeitpunkt der Aufzeichnung von Maus- und Augenbewegung ist derselbe . . . . .	40

4.7	Veranschaulichung der linearen Interpolation, die zur Ermittlung des Augenabstands dient . . . . .	42
4.8	Visualisierungsbeispiel der Scanpath aller Probanden auf einer Seite. Die Maus hat noch keinen Pfad überfahren. . . . .	44
4.9	Visualisierungsbeispiel der Scanpath aller Probanden auf einer Seite. Die Maus hat einen Pfad berührt. Dieser Pfad wird selektiert und alle anderen werden ausgeblendet. . . . .	44
4.10	Der Analyst selektiert einen Bereich mit der Maus. Dieser selektierte Bereich beinhaltet nicht die Menge aller existierenden Pfade. Als Folge dessen wird ein Pfad aus der Selektion ausgeschlossen und ausgeblendet . . . . .	45
4.11	Visualisierungsbeispiel der Scanpath aller Probanden auf einer Seite. Die Maus Hat einen Pfad Überfahren. Dieser Pfad wird selektiert und alle anderen werden ausgeblendet. . . . .	45
4.12	Visualisierungsbeispiel der Scanpath aller Probanden auf einer Seite. Die Maus Hat einen Pfad Überfahren. Dieser Pfad wird selektiert und alle anderen werden ausgeblendet. . . . .	46
5.1	Gesamtsystem mit den Hauptkomponenten . . . . .	47
5.2	Modell des Visual Analytics Tools . . . . .	49
5.3	Abfahren des Pfades mit der Maus, zeichnet zeitlich dazugehörige Punkte der Augendaten . . . . .	53
5.4	Auf dem nächsten Pfad wurde vor den nächsten Mausaktion ein großer Teil des Textes gelesen . . . . .	53
6.1	Zu einer Mausbewegung zugehörige Augenbewegungen . . . . .	58
6.2	Selektion eines Probanden . . . . .	59
6.3	Setzen zweier Kreisselektionen auf der Oberfläche, einer in der Studie untersuchten Webseiten. Hier wird zum Beispiel untersucht, welche Probanden beim Lesen der Frage mit der Maus über dem Text waren und anschließend die Lösung auch mit Hilfe der Hand gesucht haben. . . . .	61
6.4	Auf dieser Seite waren die Probanden an der Stelle der Studie, an der sie einen Beitrag suchen mussten. An Hand der Abbildung kann man ablesen, welcher Beitrag gesucht war. Des weiteren sind in Betracht gezogene Alternativmöglichkeiten zu erkennen, falls der Beitrag nicht sofort ersichtlich war. Das Suchfeld oben recht war dabei, die Alternative . . . . .	62

## Tabellenverzeichnis

---

2.1	Auflistung der von Ben Shneidermann abstrahierten Anforderungen an eine Informationsvisualisierung . . . . .	16
-----	--	----



4.1	Realisierung der Baumstruktur in einer Tabelle . . . . .	32
6.1	Auflistung der Hypothesen, die mit diesem VA Tool untersucht wurden. Die vollständige Beschreibung der Hypothesen und der Ergebnisse der Untersuchung sind der Arbeit [Kru14] zu entnehmen . . . . .	57
6.2	Use Case zum Beispielszenario für die Filterung nach Probanden. Die Vorbedingung ist, dass der Analyst die Visualisierung einsatzbereit vorliegen hat. . . . .	59
6.3	Use Case zum Beispielszenario für die Bereichsselektion. Die Vorbedingung ist, dass der Analyst die Visualisierung einsatzbereit vorliegen hat. . . . .	60

## Verzeichnis der Listings

---

2.1	Beispiel wie man mit der Heatmap.js eine Heatmap zeichnen kann. . . . .	21
5.1	JSON Datensatz der Mausdaten. Dieser Datensatz wird anschließend mit d3 visualisiert . . . . .	50
5.2	Da muss ich noch was machen . . . . .	50
5.3	Datensätze der Mausbewegungen werden mit d3 als Pfade gezeichnet und je nach Abstandswert eingefärbt . . . . .	51
5.4	Event-Handler regelt die Scanpathselektion bei Überfahren mit der Maus . . .	52
5.5	Registrierung der Event-Handler um die Kreisselektion realisieren zu können. Implementelle Umsetzung der dynamischen Interaktion zwischen Oberfläche und Cursor . . . . .	53
5.6	Implementelle Umsetzung der Pfade die durch den gezeichneten Kreis definierten Bereich an Koordinaten . . . . .	55
5.7	Implementelle Erzeugung der Abstandsheatmap . . . . .	56

# 1 Einleitung

Das Web ist heutzutage nicht mehr aus dem Alltag und dem Arbeitsleben wegzudenken. Es ist eines der einflussreichsten und mächtigsten Medien der heutigen Zeit. Es gibt kaum noch Firmen, die keine qualitativ hochwertige Webpräsenz besitzen. Große Firmen, können es sich kaum leisten, einen unordentlichen oder unübersichtlichen Webauftritt anzubieten. Die Usability der Homepage ist für die Außenwirkung eines Unternehmens oder einer Marke von entscheidender Bedeutung. Die Frage ist nun, wie ein Webentwicklungsunternehmen sicherstellen kann, dass ein gewisser Grad an Usability erreicht werden kann. Es gibt einige Grundregeln an die sich ein Webentwickler bei der Entwicklung eines Homepagedesigns halten sollte. Diese Grundregeln wurden möglicherweise aus Resultaten vergangener Studien abgeleitet und mögen bis zu einem gewissen Stand der Realität entsprechen. Doch sind Webseiten heutzutage so komplex, das es immer schwerer wird das menschliche Nutzungsverhalten auf diesen zu beschreiben.

Im Zentrum aktueller Forschungsfragen auf dem Gebiet der Web-Usability stehen mit zunehmendem Maße Eye-Tracking-Experimente. Auf Basis dieser Experimente erhofft man sich, Erkenntnisse zu gewinnen, die Webseitenentwicklern helfen sollen, ihre Webseite benutzerfreundlicher zu gestalten. Die Augen des Menschen sollen bei einem Experiment anhand speziell ausgewählter Szenarien, der Schlüssel zur Denkweise des Gehirns sein. Während des Experiments werden permanent die Augenbewegungen aufgezeichnet. Sie sollen am Ende verdeutlichen bei welchen Elementen einer Webseite sich das Verhalten verändert. Viele erfolgreich abgeschlossene Forschungen auf dem Gebiet des Eyetrackings zeigen, dass dies eine effektive Methode darstellt, um in die Gedankenwelt und die Absichten eines Benutzers blicken zu können.

Die Fixationspunkte des Auges sind für die Untersuchung des Benutzerverhaltens sehr wertvoll. Jedoch ist das für die Erfassung dieser Punkte benötigte Eyetracking System sehr kostspielig. Nun benötigt man an Methode mit der man Daten gewinnt die eventuell gleichermaßen wertvoll sind. Nun besteht eine interessante Forschungsfrage darin, inwieweit sich Mausbewegungen und Augenbewegungen gleichen.

Ziel dieser Arbeit ist die Entwicklung eines Visual Analytics Ansatzes, mit dem sich Eye-tracking und Mousetracking Daten geeignet filtern, visualisieren und zeitlich explorieren lassen. Die zu visualisierenden Daten werden aus einer Studie gewonnen. Die Studie besteht aus verschiedenen Szenarien, die sich auf Webseiten abspielen. Hauptaugenmerk liegt dabei darauf, ob sich in bestimmten Bereichen der untersuchten Webseiten Korrelation zwischen Maus- und Augendaten ergeben. Mit dem entwickelten Visual Analytics Ansatzes soll es möglich sein, bestimmte Bereiche zu markieren, um diese näher zu untersuchen. Des weiteren soll die Möglichkeit bestehen einzelne Probanden oder bestimmte Probandengruppen auszuwählen und auch diese im Detail zu untersuchen.

## Gliederung

Die Arbeit ist in folgender Weise gegliedert:

**Kapitel 2 – Grundlagen:** In diesem Kapitel werden die Grundlagen zu dieser Arbeit erläutert. Diese sollen eine Basis für das Verständnis dieser Arbeit schaffen. Zu den Grundlagen gehört ein Einblick in die Visual Analytics, Eyetracking Visualisierungen und technische Grundlagen zur Implementierung des Konzepts.

**Kapitel 3 – Verwandte Arbeiten:** Dieses Kapitel geht auf verwandte Arbeiten zu diesem Thema ein. Diese behandeln ähnliche Themen auf dem Gebiet der Visual Analytics, die im Rahmen dieser Arbeit unterstützend in die Entwicklung der Lösungskonzepte eingehen.

**Kapitel 4 – Lösungskonzept:** In diesem Kapitel wird das Lösungskonzept präsentiert. Einzelne Schritte beschreiben den Weg zur entgeltigen Entwicklung des Visual Analytics Konzepts basierend auf dem Schwerpunkt, die Untersuchung von Eye- und Mouse-tracking Daten zu gewährleisten.

**Kapitel 5 – Implementierung:** Inhalt dieses Kapitels ist die Implementierung des Visual Analytics Systems. Dabei wird auf die einzelnen Komponenten des kompletten Systems im Detail eingegangen. Ausschnitte aus dem Code verdeutlichen die Funktionsweise der Komponenten.

**Kapitel 6 – Fallstudie:** Dieses Kapitel demonstriert die Realisierung des finalen VA Ansatzes dieser Arbeit. Es werden die Funktionen vorgestellt und Abläufe beschrieben, wie die Implementierung bei der Untersuchung von Korrelationen von Eyetracking und Mastracking Daten verwendet wird.

**Kapitel 7 – Zusammenfassung und Ausblick** Im letzten Kapitel wird ein Resumé über die Arbeit gezogen. Des weiteren ist ein Ausblick verfasst, der zukünftige Schritte in Richtung einer Verfeinerung dieses Visual Analytics Konzepts beschreibt.

## 2 Grundlagen

Für die Untersuchung der Korrelation zwischen Augen- und Mausdaten benötigt es eine große Menge an Daten. Diese Daten gewinnt man aus Eyetracking Studien. Jedoch sind diese Daten Rohdaten der Bewegungen, die bei bloßer Betrachtung noch keine Lösung des Problems offenbaren. An dieser Stelle muss auf Datenvisualisierung zurückgegriffen werden, um diese Rohdaten für die Untersuchung verwendbar zu machen. paragraph

### 2.1 Visual Analytics

*Visual Analytics ist ein interdisziplinärer Ansatz, der die Vorteile aus unterschiedlichen Forschungsgebieten verbindet. Das Ziel der Visual Analytics-Methode ist, Erkenntnisse aus extrem großen und komplexen Datensätzen zu gewinnen. Der Ansatz kombiniert die Stärken der automatischen Datenanalyse mit den Fähigkeiten des Menschen, nämlich schnell Muster oder Trends visuell zu erfassen. Durch geeignete Interaktionsmechanismen können Daten visuell exploriert und Erkenntnisse gewonnen werden.[KPSNo4]*

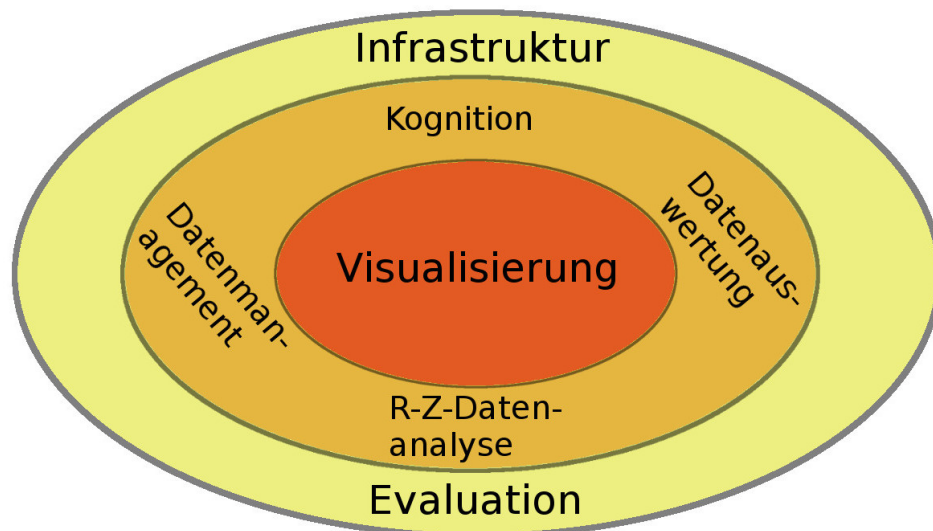
Gegenstand dieser Arbeit ist es, Mausmovements und Eyetracking Daten auf Korrelation untersuchen zu können. Um diese Analyse möglich zu machen, benötigt man eine große Menge an Daten. Visualisiert man diese Daten statisch, ist es unmöglich hinsichtlich des Problems, aus dem Gesamtbild Erkenntnisse zu gewinnen. An dieser Stelle ist der Einsatz von Visual Analytics notwendig. Es muss möglich sein, die Daten zu filtern, sie auf verschiedene Eigenschaften untersuchen zu können und die zeitliche Entwicklung zu betrachten. Der Mensch kann Bilder schnell scannen, wiedererkennen, benennen und Änderungen in Größe, Farbe, Form, Bewegung und Struktur wahrnehmen[Shn96]. Daher ist es wichtig eine einfache Interaktion von Visualisierung und Benutzer einzuplanen, um beispielsweise Muster und Auffälligkeiten schnell zu realisieren.

Im Folgenden wird auf die Visual Analytics im Detail eingegangen. Alle aufgeführten Inhalte der Unterkapitel „Visual Analytics als neues Forschungsgebiet“ und „Der Prozess von Visual Analytics“ basieren auf dem Buch *Mastering the Information Age Solving Problems with Visual Analytics* [KKEM10].

#### 2.1.1 Visual Analytics als neues Forschungsgebiet

Visual Analytics wird immer wichtiger und aufwendiger, da es in der heutigen Welt einen immer rasanteren Anstieg an jeglicher Art von Daten gibt. Es ist zu einer eigenen technischen

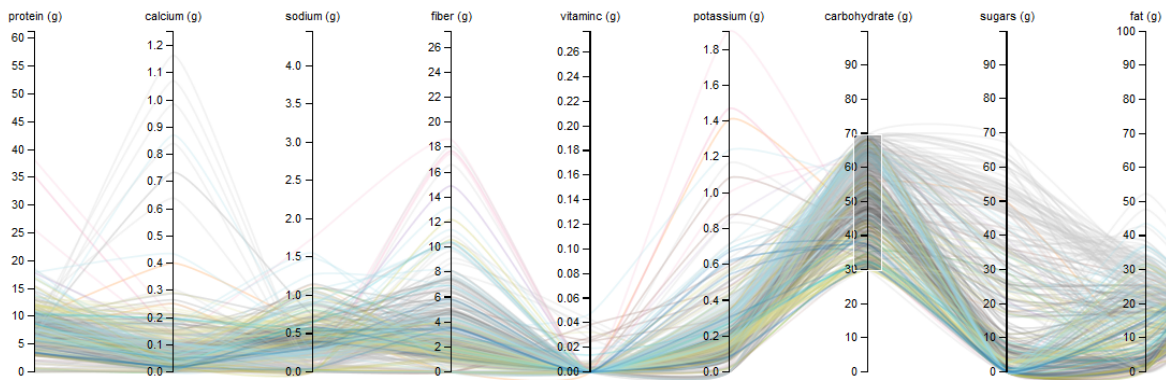
Disziplin geworden, die aus einer Vereinigung von wissenschaftlichen und technischen Gebieten besteht. Dieser neue Bereich des Softwareengineering soll mit Abbildung 2.1 verdeutlicht werden. Der Kern des ganzen ist die Visualisierung. Dieser Kern setzt sich aus allen anderen Schichten zusammen. Das bedeutet, dass zwischen den einzelnen Schichten gewisse Abhängigkeiten bestehen. Diese fließen somit auch maßgeblich in die Qualität des „Endprodukts“ mit ein. Ist beispielsweise die Infrastruktur des testenden Systems schlecht oder eingeschränkt beeinflusst das die Erfassung der Daten und am Ende die Visualisierung. Ebenso ist eine schlechte Evaluation der Daten ein Faktor, der sich durch den gesamten Prozess bis zur Qualität der Visualisierung durchzieht.



**Abbildung 2.1:** Teildisziplinen in der Visual Analytics. Die Endgültige Visualisierung ist abhängig von jedem Teilgebiet [KKEM10].

**Datenvisualisierung** Grundsätzlich kann man das Gebiet der Visualisierung in zwei Bereiche unterteilen. Zum Einen behandeln sie wissenschaftliche Daten aus Bereichen der Mathematik, der Physik oder anderer wissenschaftlicher Fachbereiche. Sie bewegen sich oft im 3 dimensionalen Raum und verwenden meist geometrische Körper, Flächen oder eine Art 3 dimensionale Wolke. Wissenschaftliche Daten haben das Charakteristikum, dass sie durch die jeweilige Wissenschaft fest definierte Einheiten sind. Daher halten sich diese Größen an Gesetze der Natur und sind oft raum-zeitlich darstellbar.

Zum Anderen geht es um abstrakte Daten, die sich nicht an feste „Formen“ und Einheiten halten. Sie können beliebig viele Attribute haben. Beispiele dafür sind Daten aus sozialen Netzwerken, Businessdaten, Daten die eine bestimmte Software generiert, alle Daten die abstrakte Datenstrukturen aufweisen. Auch die Daten, die eine Eye- oder Mousetracking Software aufgezeichnet hat, sind solche abstrakten Daten. Diese sind Gegenstand dieser Arbeit. Wo wissenschaftliche Daten in den meist drei Dimensionen aufweisen, haben solch abstrakte Daten mehrere Dimensionen. Abbildung 2.2 soll diese Dimensionen an einem Beispiel der Nährstoffwerte in Nahrungsmitteln verdeutlichen.



**Abbildung 2.2:** "Parallel Coordinates" bei Nährstoffwerten von Nahrungsmitteln. Es wurde ein Bereich von 30-70 g Kohlenhydrate ausgewählt und die Visualisierung zeigt alle Lebensmittel, die für diese Auswahl in Frage kommen.[Exa13]

In den Fällen der Mehrdimensionalität, ist es dann nicht mehr so einfach die Daten mit einfachen Balkendiagrammen oder Graphendarstellungen erfolgreich und anschaulich zu visualisieren. An dieser Stelle sind andere Visualisierungstechniken notwendig, um eventuell schon vorhandene Techniken mit eigenen Ideen zu verschmelzen.

**Datenmanagement** Das Datenmanagement ist ebenso ein wichtiger Faktor für die schnelle und qualitativ hochwertige Entwicklung von aussagekräftigen Visualisierungen. Daten müssen irgendwo organisiert werden. Heutzutage sind die meisten Datenbanken. Sie sollten sicher und für die Visualisierungssoftware immer ohne Probleme zugänglich sein. Das wichtigste am Datenmanagement ist, dass ein für die Visualisierung treffendes Datenbankmodell entworfen wird. Ist dies nicht gegeben, kann das nicht nur den weiteren Prozess der Visualisierung erheblich behindern, es kann sogar die Ergebnisse teilweise verfälschen.

**Datenauswertung** Oft ist es der Fall, dass man aus erzeugten Datensätzen nicht alle Informationen benötigt. Daher muss eine Datenanalyse erfolgen, bei der man bezüglich des Problems nur diese Daten für die Visualisierung verwendet, die auch wirklich benötigt werden. Wichtig ist auch, zu überprüfen ob die Ausprägungen der erfassten Daten ausreichend sind, um die Visualisierung perfekt gestalten zu können.

**Spatiotemporalen Datenanalyse** Großes Interesse genießen auch Daten aus der realen Welt. Beispielsweise geografische Messungen, GPS-Koordinaten oder Daten die von jeglicher Art von Sensoren aus der realen Welt gesendet und aufgezeichnet werden.

Den Teil der spatiotemporalen Daten, kann man, oft in Maps, Diagrammen oder eigens für die Visualisierung entworfenen Mustern darstellen. Den anderen Teil der dazugehörigen zeitlichen Daten muss man in Form einer Funktion der Zeit realisieren. Das heißt die Daten werden sich mit der Zeit verändern. Es stellt oft eine Herausforderung dar, diese zwei Ausprägungen zu verschmelzen.

**Menschliche Wahrnehmung und Kognition** Ein weiteres interessantes Gebiet der Visualisierung ist die Untersuchung der menschlichen Wahrnehmung und Denkweisen. Mit Visualisierung ist es möglich Reaktion des Menschen auf bestimmte Aktionen sichtbar zu machen. Daraufhin kann man Rückschlüsse ziehen, bei welchen Aktionen der Mensch, wie reagiert. Auf dem Gebiet der Wahrnehmung wird dazu ein Eyetracker verwendet. Das liegt nahe, da die Augen dafür zuständig sind, visuelle Reize wahrzunehmen und diese direkt über die Sehnerven in das Gehirn zu übertragen.

**Infrastruktur** Dieser Teil ist der Technischste am ganzen Visual Analytics Prozess. Bei der Infrastruktur muss darauf geachtet werden, dass alle Komponenten einwandfrei miteinander funktionieren. Eine Visualisierung basiert auf einer extrem großen Menge an Daten. Diese müssen auf effizientem Wege in das Visualisierungstool gelangen. Das Tool an sich, kann beispielsweise auf einem Server laufen, was gewisse Vorteile birgt, wenn sich die Daten in einer Datenbank befinden. Auch wenn die Visualisierung nicht auf einem Server realisiert wurde, benötigt das ganze System eine gut organisierte Infrastruktur mit genau definierten Schnittstellen.

**Evaluation** Nur wenn ein Analyst in der Lage ist, die durch die Visualisierung verändert dargestellten Daten auf spezielle Hypothesen zu untersuchen und daraus Schlüsse zu ziehen, ist eine Visualisierung gelungen. Dazu gehört das Filtern, zeitliche Explorieren und das statistische Auswerten der Daten.

### 2.1.2 Der Prozess von Visual Analytics

In diesem Abschnitt wird der gesamte Prozess der Visual Analytics beschrieben. Er soll helfen im Folgenden zu erkennen, welche Teilschritte und Teilprozesse bis hin zur endgültigen Visualisierung durchlaufen wurden.

Visual Analytics ist ein Prozess. Dieser soll in abstrahierter Form durch Abbildung 2.3 verdeutlicht werden. Im Weiteren soll dieser Prozess im Detail erklärt werden.

**Daten Transformation** In vielen Fällen der Datenerfassung müssen die Daten vorher transformiert werden. Das heißt sie müssen vorher in irgendeiner Weise bearbeitet werden, damit mit der Modellbildung begonnen werden kann. Mit Transformieren ist das Filtern von für die Visualisierung wichtiger Parameter aus den erfassten Datensätzen gemeint. Weitere Transformationen sind beispielsweise das Säubern der Daten, Normalisierung der Daten<sup>1</sup> oder das Gruppieren von gleichen Datensätzen.

<sup>1</sup>Unter Normalisierung [...] versteht man die Aufteilung von Attributen [...], sodass eine Form entsteht, die keine vermeidbaren Redundanzen mehr enthält. [Wik13b]

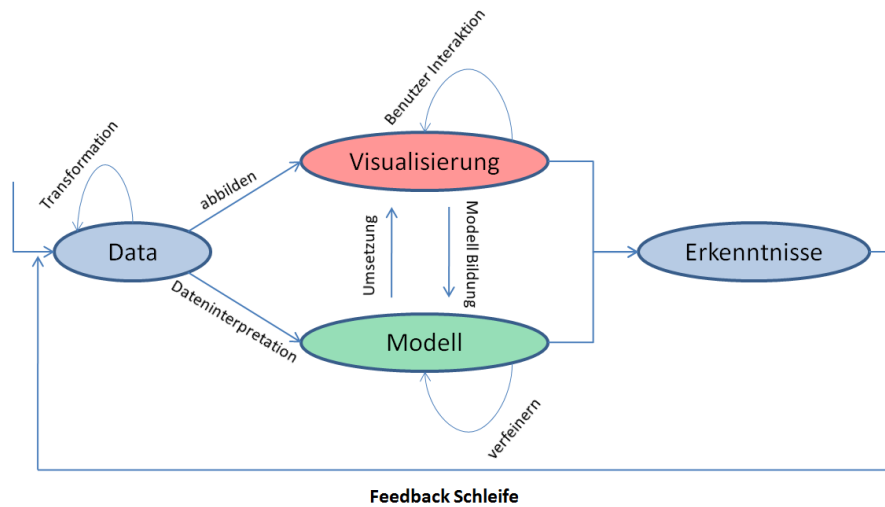


Abbildung 2.3: Abstrakter Daten-Visualisierungsprozess [KKEM10]

**Visualisierung und Modellbildung** Nach der Transformation der Daten, gibt es zwei mögliche Wege die bearbeiteten Daten zu verwenden. Entweder wurden die Daten durch die Transformation so vereinfacht, dass schon eine Visualisierung dieser möglich ist, oder es müssen geeignete Modelle für die komplexen Daten entworfen werden. Mit diesen definierten Modellen kann es dann möglich sein eine passende Visualisierung zu finden. Das Entwerfen solcher Modelle hat gewisse Vorteile, denn es macht eine Art Interaktion des Entwurfs mit den Daten möglich. Mit einem ständigen Wechsel der Visualisierung von abstrakten Modellen und anschließendem Verfeinern dieser Modelle, ist es möglich, sehr gute Darstellungen zu finden. Nicht nur das kontinuierliche Verfeinern ist gegeben, es kann auch frühzeitig erkannt werden, ob das entworfene Modell unbrauchbar ist oder nicht zum erwarteten Ergebnis führen kann.

**Erkenntnisse** Ist die Visualisierung oder Modellbildung abgeschlossen, ist es möglich daraus Erkenntnisse zu ziehen. An diesem Punkt erkennt der Analyst, ob es irgendwo im Prozess fehlende Aspekte gibt. Dies ergibt sich daraus, dass es nicht möglich ist die Visualisierung auf bestimmte Eigenschaften zu untersuchen oder zu filtern. Oft ist es auch der Fall, dass alles an Informationen in der Visualisierung stecken, diese aber so schlecht umgesetzt wurden, dass es gar nicht oder nur schwer möglich ist das entwickelte Modell zu untersuchen. Ist der Visual Analytics Prozess jedoch Schritt für Schritt ordnungsgemäß ausgeführt und eventuell in einem kleinen Review besprochen worden, dann kann dies dazu führen, dass diese Probleme auf ein Minimum reduziert werden.

Ist es doch der Fall, dass die Visualisierung zu mangelhaften Ergebnissen führt, dann muss der gesamte Prozess noch einmal überarbeitet werden. Denn das Problem kann schon an einer schlechten Transformation der Daten liegen.



### 2.1.3 Visual Information Seeking Mantra

Das *Visual Information Seeking Mantra* von Ben Shneiderman [Shn96] beschreibt ein Grundprinzip, dass bei der Entwicklung von Informationsvisualisierungen beachtet werden sollte. Er schlägt dabei eine *Task by Type Taxonomy* vor. Nach Shneidermann gibt es sieben Arten von Datentypen. Diese sind die 1-,2-,3- dimensionalen Daten, zeitabhängige und mehrdimensionalen Daten, Baum- und Netzwerkdaten. Anhand dieser Datentypen ergeben sich mögliche Aufgabenbereiche, die Anforderungen an die Visualisierung sind. Diese Anforderungen fasst Shneidermann sehr abstrahiert zusammen. Die sieben Anforderungen sind.

Overview:	Gewinne einen Überblick, über alle Daten
Zoom:	Betrachte interessante Daten genauer
Filter:	Uninteressante Daten ausblenden
Details-on-Demand:	Selektiere einen Datensatz oder Gruppen und gib Details falls benötigt
Relate:	Zeige Beziehungen zwischen Daten
History:	behalte eine Historie um Undo und Redo zu gewährleisten
Extract:	Erlaube Abwandlung von Untermengen und Suchanfragen

**Tabelle 2.1:** Auflistung der von Ben Shneidermann abstrahierten Anforderungen an eine Informationsvisualisierung

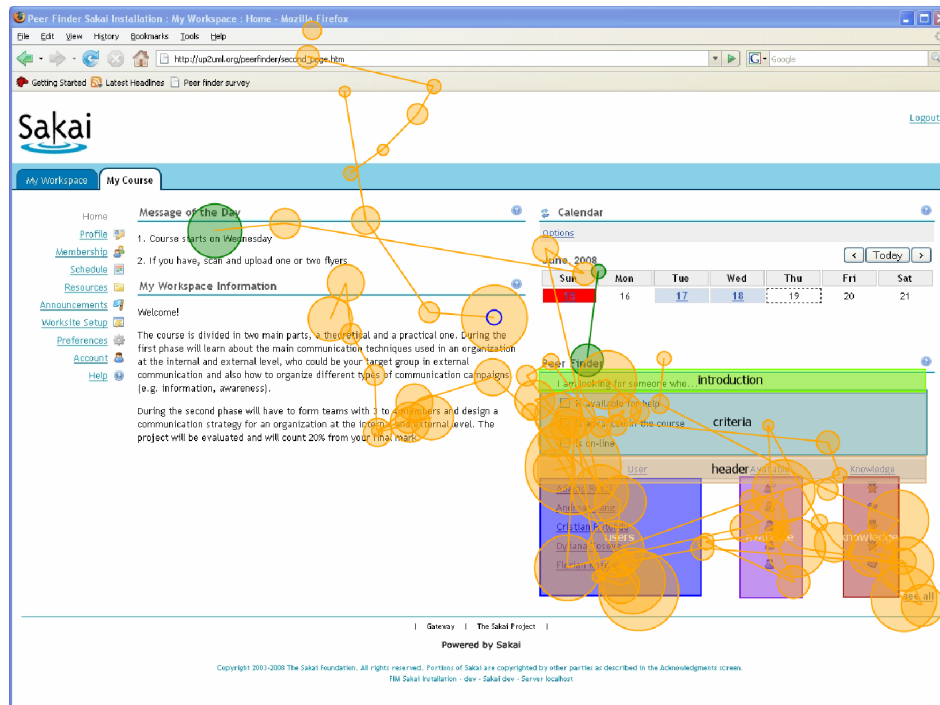
## 2.2 Eyetracking Visualisierung

Eyetracker sind Systeme, die es möglich machen, die Augenbewegungen eines Benutzers während der Arbeit präzise aufzuzeichnen. Dabei wird in bestimmten Zeitintervallen die Augenposition auf dem Bildschirm ermittelt. Wenn diese Zeitintervalle entsprechend klein gewählt sind, ist es möglich Punkte herauszufiltern, an denen der Benutzer längere Zeit verharrete. Im Weiteren werden diese Punkte Fixationspunkte oder Gazepunkte genannt. Diese Fixationspunkte werden dann mit einigen wichtigen Attributen aufgezeichnet und abgespeichert. Merkmale dieser Aufzeichnungen sind dann beispielsweise die Koordinaten der Position, ein Zeitstempel, wann die Aktion geschah und möglicherweise auch das zugehörige Objekt, das im Fokus stand.

### 2.2.1 Scan Path

Während einer Eyetracking Studie werden die Augenpunkte der Probanden aufgezeichnet. Diese Punkte sind Grundlage für die Visualisierung mit einem Scan Path. Ein Scan Path verbindet die einzelnen Punkte miteinander, um so den Weg, den die Augen während einer Studie genommen haben, sichtbar zu machen. Häufig werden dabei die Punkte in einer

unterschiedlichen Größe dargestellt. Die Größe der Punkte soll verdeutlichen, wie lange die Augen eines Probanden auf diesem Punkt verharren.



**Abbildung 2.4:** Ein Scanpath, der Fixationspunkte der Augen darstellt. Die Kreisgröße an den einzelnen Punkten beschreibt die Aufenthaltsdauer der Augen an diesem Punkt[LN12]

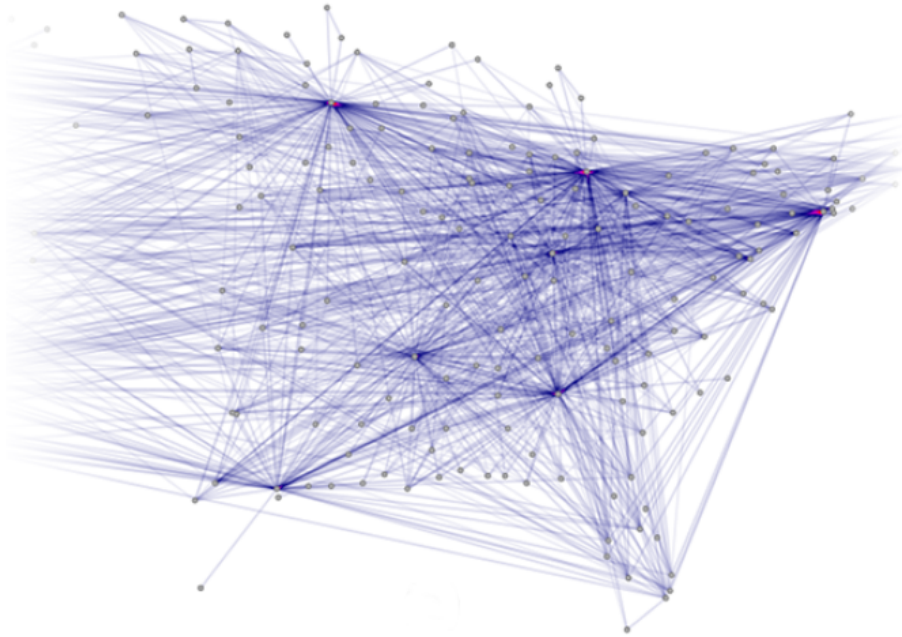
**Visual Clutter und Edge Bundling** Bei der Visualisierung von Augendaten mit Scan Path entstehen große Graphen. Graphen sind eine Visualisierung, die Knoten oder besser Einheiten in Form von Kanten miteinander verbindet. Sie verdeutlichen die Verbindungen zwischen einzelnen Einheiten, auf direktem Wege oder über einen Pfad von mehreren Kanten.

Ein Problem an Graphen ist, dass sie bei einer großen Anzahl an Daten nicht sehr übersichtlich bleiben. Je mehr Knoten und Kanten ein Graph hat, umso mehr entsteht ein riesiges Netz aus unübersichtlichen Kanten-Knoten-Verbindungen. Es entsteht ein Chaos. Dieses Chaos nennt man in der Fachsprache „Visual Clutter“. Abbildung 2.5 zeigt ein Beispiel für diesen Sachverhalt.

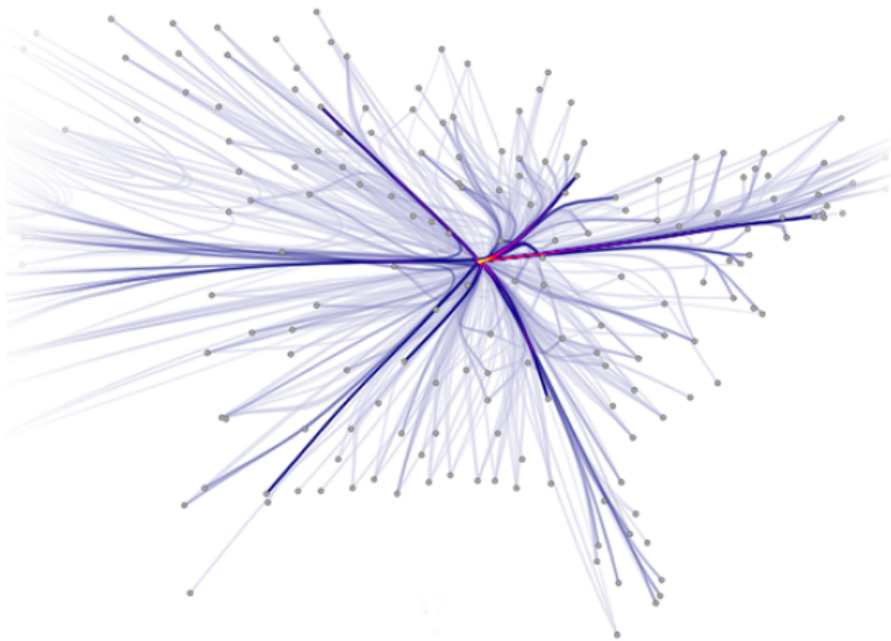
In einem Graphen, in dem dieses Phänomen auftritt, kann keiner mehr den eigentlichen Zusammenhang der Daten erkennen. Um diesem Problem Abhilfe zu schaffen, gibt es Möglichkeiten den „Visual Clutter“ zu reduzieren. Dabei sollte das Ziel sein keine oder nur sehr wenig Informationen aus dem Graphen zu verlieren. Es sollte also vermieden werden, den Graphen in irgendeiner Form zu reduzieren. Es sei denn, man setzt den Fokus einfach, wie bei einer Lupe näher in den Graphen auf spezielle Bereiche. Dann würde man

den Graphen auf eine spezielle Art reduzieren und außenstehende Kanten und Knoten ausblenden.

Eine Möglichkeit dieses Chaos zu dämmen, ist eine alternative Art den Graphen darzustellen. Diese Darstellung erreicht man mit Edge Bundling [HW09]. Edge Bundling folgt einem Algorithmus, der einen Graphen als Input hat. Dieser Algorithmus schaut sich zwei Kanten des Graphen an und unterteilt diese in kleinere Segmente. Er markiert die Kanten also mit einer bestimmten Anzahl an Punkten, die alle samt den gleichen Abstand zueinander haben. Wenn diese einzelnen Segmentpunkte in einem bestimmten Abstand stehen, dass sie für das Bündeln in Frage kommen, wird für beide Kanten und deren Segmentpunkte ein neuer Wert berechnet, der näher aneinander liegt, als der vorherige. Die grundsätzliche Idee dabei ist, bestimmte Kanten zu einer Gruppe zusammenzufassen und somit eine Art Hauptadernetz zu generieren.



**Abbildung 2.5:** Beispiel für Visual Clutter [HW09]



**Abbildung 2.6:** Beispiel für Edge Bundling[HW09]

### 2.2.2 Heatmap

Eine Heatmap ist ein Diagramm zur Visualisierung zweidimensionaler Daten. Jeder hat im Alltag schon einmal eine Heatmap gesehen. Die wohl Bekannteste, ist die Wettervorhersage. Dabei werden aufkommende Gewitter mit einem dunkelroten Bereich über dem Gebiet markiert. *Sie besitzen [...] einen großen Vorteil. Ihnen steht durch die farbliche Darstellung eine weitere Dimension zur Veranschaulichung zur Verfügung [Ale13].* Für diese Dimension wird eine Farbskala definiert, bei denen das eine Ende der Skala für wenig Aktivität und das Andere für sehr viel Aktivität steht.

Das menschliche Auge ist sehr aktiv und beschäftigt sich ohne Pause mit den Dingen, die in seiner Umwelt geschehen. Betrachtet das Auge ein Bild, wird es dieses in bestimmter Weise tun. Würde man an jedem Punkt, auf den das Auge geschaut hat, einen Punkt zeichnen, so würde sich ein gewisses Muster ergeben. Dieses Muster beschreibt im Prinzip eine Art Heatmap.

## 2.3 Technische Grundlagen

In den Technischen Grundlagen wird hauptsächlich auf Techniken eingegangen, die das Web umfassen. Sie sollen im Verlauf der Arbeit zum Verständnis von Sachverhalten aus dem Internet beitragen.

### 2.3.1 DOM-Document Object Model

Document Object Model (DOM) ist eine Spezifikation einer Schnittstelle für den Zugriff auf HTML- oder XML-Dokumente. Sie wird vom World Wide Web Consortium definiert. Eine Implementierung, die dieser Spezifikation genügt, besteht im Sinne der objektorientierten Programmierung aus einem Satz von Klassen zusammen mit deren Methoden und Attributen. Sie erlaubt Computerprogrammen, dynamisch den Inhalt, die Struktur und das Layout eines Dokuments zu verändern[Wik13a].

### 2.3.2 Webbasierte Visualisierung: Externe Bibliotheken

Das heutige Web hat einiges zu bieten. Aufklappbare Menüs, Slider oder auch Elemente die sich dynamisch anordnen. Doch das ist noch lange nicht alles. Man hat die Möglichkeit die Elemente einer Webseite dynamisch zu verändern, zu erzeugen, zu bewegen oder zu löschen. Dies kann man unter anderem dazu nutzen, um Daten auf einer Webseite zu visualisieren. Der Grund dafür ist ein entscheidender Vorteil, den das Web gegenüber nativer Software bietet. Es kann clientunabhängig genutzt werden. Man benötigt nur einen Internetzugang und möglicherweise eine Berechtigung, die entsprechende Webseite besuchen zu dürfen. Wobei bei einem eigenständigen Programm das Problem bestünde, dass der entsprechende User das Programm besitzt und es lauffähig auf seinem Computer installieren kann.

Im Folgenden werden interessante Javascript Bibliotheken vorgestellt, die eine große Hilfe bei der Webbasierten Visualisierung sein können.

**D3.js - Data-Driven Documents** D3.js ist eine Bibliothek, die basierend auf Datensätzen, Webseiten manipuliert. Der Hauptaugenmerk liegt auf der Darstellung mittels SVG-Grafiken.

**jQuery** jQuery ist eine Bibliothek, die es ermöglicht auf die DOM-Elementen einer Webseite zuzugreifen und diese zu manipulieren. Anhand des id- oder class-Attributes können einzelne oder mehrer Elemente verwendet werden.

Des weiteren bietet jQuery die Möglichkeit GUI-Elemente ansehnlich darzustellen.

**Heatmap.js** Eine sehr bekannte Visualisierungstechnik ist die Heatmap. Heatmap.js ist eine Bibliothek, die es sehr einfach macht Heatmaps auf HTML-Elemente zu zeichnen. Um eine Heatmap zu erzeugen, muss eine Konfiguration geschrieben werden, die dann an eine „Factory“ übergeben wird. Dieser Heatmap wird dann ein recht einfaches JSON Objekt mit Koordinaten übergeben. Das folgende Beispiel soll dies verdeutlichen.

```
// Configure the heatmap you want to draw
var config = {
  "element": heatDiv,
  "radius": 30,
  "opacity": 50,
  "legend": {
    "position": 'br',
    "title": 'Example Distribution'
  }
};
var heatmap = heatmapFactory.create(config);

// Setting the Heatmap Data
heatmap.store.setDataSet({ max: 10, data: [{x: 10, y: 20, count: 5}, ...]});
```

**Listing 2.1:** Beispiel wie man mit der Heatmap.js eine Heatmap zeichnen kann.



## 3 Verwandte Arbeiten

In diesem Kapitel werden zwei schon veröffentlichte Arbeiten vorgestellt. Beide untersuchten Augenbewegungen und Mausbewegungen parallel bei der Bewältigung von Aufgaben im Internet. Im Bezug auf diese Arbeit, lag das Hauptaugenmerk auf die anschließende Visualisierung dieser beiden Bewegungen. Dabei wurden Visualisierungstechniken in diesem Kontext in Augenschein genommen. Auch die vorbereitenden Prozesse bis zum Endergebnis waren von großem Interesse. Diese Erkenntnisse flossen in diese Arbeit mit ein. Sei es als Motivation oder in Kombination mit neuen Ideen zur Visualisierung dieser Art von Daten.

### 3.1 Visualisierung von Maus- und Augebewegungen im Internet

#### 3.1.1 Untersuchung der Webseiten-Layouts

Viele Studien haben Mastracking als ein Werkzeug genutzt, um Verhaltensweisen auf Webseiten zu untersuchen. Diese erzeugten Daten aus Studien wurden dazu verwendet, um Verhalten und Vorhersagen zu Benutzern machen zu können. Insbesondere werden die Benutzerfreundlichkeit untersucht und ob ein Benutzer beim Lesen von Texten Frustration erzeugt und eventuelle Lesekämpfe austrägt. Diese Voraussagen sollen an speziellen Maus-Auge-Mustern erkannt werden. Zusätzlich kann man an aufgezeichneten Mausbewegungen erkennen, ob ein Seitenlayout geschickt gewählt ist, oder ob ein Benutzer eher Schwierigkeiten hat mit der Webseite und deren Inhalten umzugehen. Diese Problematiken untersuchte die Arbeit von Vidhya Navalpakkam and Elizabeth F. Churchill[NC12].

**Visualisierungen** In dieser Arbeit wurden zwei Visualisierungen verwendet. Zum einen Heatmaps und zum anderen einfache Statistiken. Zweiteres wurde mit Balkendiagrammen für bestimmte Bereiche der betrachteten Webseite dargestellt.

- **Heatmap:** Die Heatmaps wurden direkt auf der betroffenen Webseite generiert und sollten die Augen- beziehungsweise Mausbewegungen sichtbar machen. Der Analyst wurde so sofort auf Interessengebiete aufmerksam und es sind schnell Muster erkennbar.
- **Balkendiagramme:** Die Details und Fakten der einzeln betrachteten Gebiete der Webseite, wurden mit Balkendiagrammen veranschaulicht. Zum Beispiel wurden, bei einer Einblendung von sich ständig wechselnder Werbung, die Zeiten von Blick auf Text und

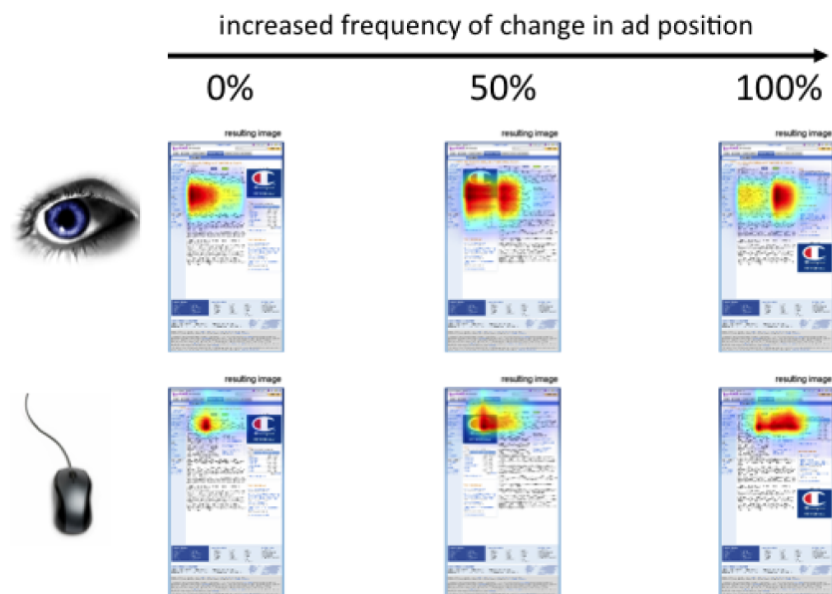


### 3 Verwandte Arbeiten

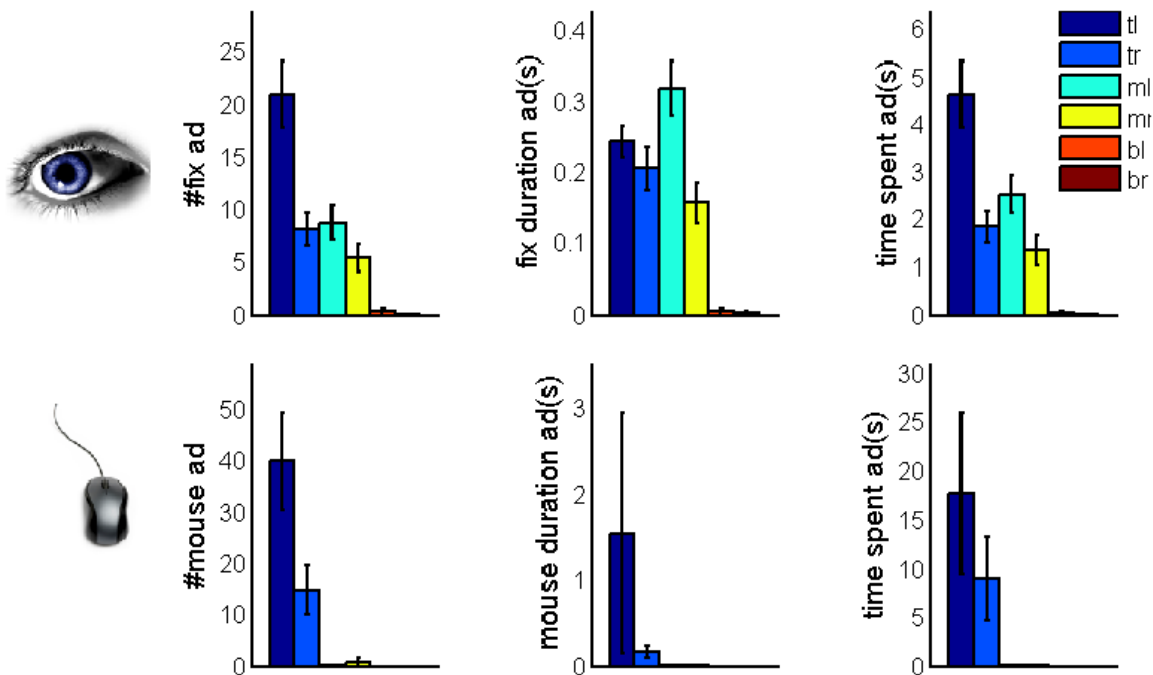
---

Blick auf Werbung dargestellt.

Solche einfachen Arten von Diagrammen sind sehr geschickt für die Verwendung in Details oder wenn der Fokus auf einer bestimmten Selektion von Interessengebieten liegt. Aus der Arbeit geht aber nicht hervor, ob die Balkendiagramme teil des Visualisierungstools waren oder nur aus der Auswertung entsprungen ist.



**Abbildung 3.1:** Oben ist die Fixations-Heatmap der Augen und unten die Heatmap der Mausposition zu sehen. Anfangs ist die Position der Werbung rechts oben fixiert. Dann wird die Position zufällig über 6 Positionen variiert. Dabei ändert sich die Position von Auge und Maus von fokussiert oben links, zu Mitte der Seite [NC12]



**Abbildung 3.2:** Einfluss der Positionierung von Werbung auf das Muster der Augen- und Mausbewegungen. Erscheint die Werbung oben auf der Seite werden mehr Fixationspunkte der Augen gezeichnet als bei Positionierung der Werbung in der Mitte oder am Ende der Seite. Erscheint die Werbung rechts werden mehr Fixationspunkte gezeichnet als bei Positionierung der Werbung links auf der Seite.[NC12]

**Abgrenzung** In dieser Arbeit von Vidhya Navalpakkam und Elizabeth F. Churchill werden verschiedene Layouts von Webseiten in Studien bereitgestellt. Daraufhin werden Fixationspunkte von Augen und Positionen der Maus für das jeweilige Szenario untersucht. Diese zwei Bewegungen werden mit Heatmaps visualisiert um später prozentuale Abschätzungen für bestimmte Bereiche der Webseite zu machen. Das Visualisierungskonzept, das diese Ausarbeitung beschreibt, soll es jedoch ermöglichen für bestimmte Bereiche von Webseiten feste Gesetzmäßigkeiten untersuchen zu können.

#### 3.1.2 Untersuchung der Ergebnisse von Suchmaschinen

Eine weitere Arbeit erforschte das Zusammenspiel zwischen Maus- und Augbewegungen auf Ergebnissen von Suchmaschinen. Suchmaschinen ermöglichen es einfach und schnell, das gewünschte Suchergebnis zu präsentieren. Die Betreiber dieser Suchmaschinen versuchen die Funktionalität immer weiter zu optimieren. Forscher fanden heraus, dass das Untersuchen von Mustern des Klickverhaltens Potential hat [RFASo8][JGP<sup>+</sup>o8]. Deshalb wurde eine Studie mit 32 Probanden durchgeführt, bei der Maus- und Augbewegungen erfasst wurden. Diese

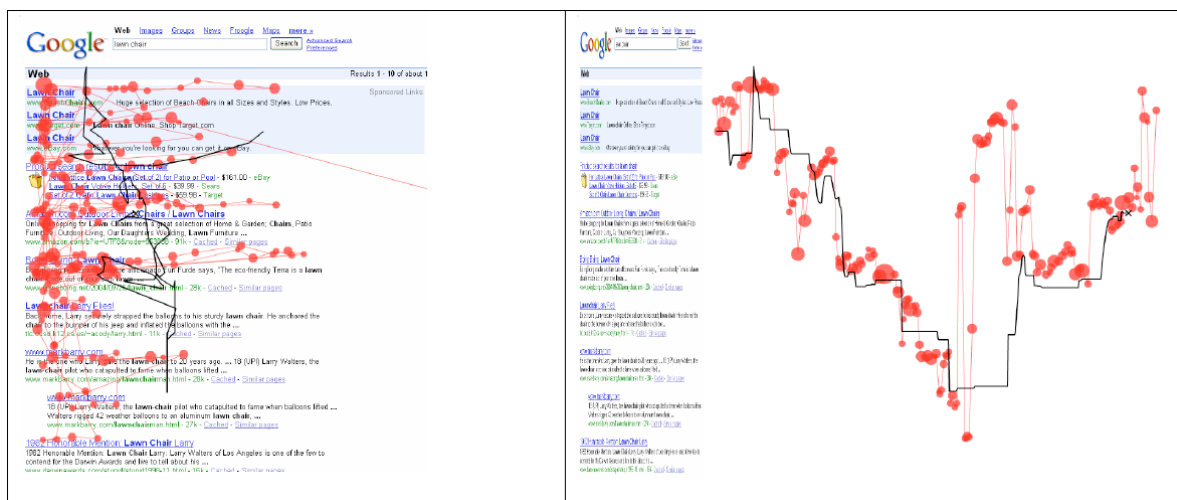
### 3 Verwandte Arbeiten

wurden dann visualisiert, um daraus Muster des Zusammenspiels zwischen Auge und Maus zu finden.

**Visualisierungen** Für die Visualisierung wurden zwei besondere Arten von Scanpaths verwendet. Diese Besonderheiten und deren Vorteile für die jeweilige Bewegungen werden im Folgenden erläutert.

- **Scanpath-Auge:** Die Scanpaths der Augenbewegungen werden in rot gezeichnet. An den Fixationspunkten der Augen, wurde ein Kreis visualisiert, dessen Größe die Aufenthaltsdauer der Augen auf diesem Punkt entspricht.
- **Scanpath-Maus:** Die Scanpaths der Mausbewegungen werden in schwarz gezeichnet. Sie zeigen die Wege, die die Maus auf dem Bildschirm gegangen ist.

Die Abbildung 3.3 zeigt die Visualisierungen, die in der Arbeit[RFASo8] verwendet wurden. Auf der rechten Seite werden die y-Koordinaten der Maus- und Augenbewegungen in Abhängigkeit der Zeit dargestellt. Dabei wird der Abstand der von Maus und Auge in y-Richtung besteht verdeutlicht dargestellt.



**Abbildung 3.3:** Example of the mouse following the eye in the vertical direction. Plotting the Y coordinate against time, in the visualization on the right, makes clear a pattern that is hidden in the visualization on the left.[RFASo8]

**Abgrenzung** In der vorgestellten Arbeit werden die Bewegungen der Maus und des Auges visualisiert. Aus diesen Visualisierungen wurde dann versucht Muster abzuleiten. Diese Muster sollen Verhaltensweisen bei bestimmten Aktionen beschreiben. Insbesondere sollte das Zusammenspiel von Auge und Maus im Fokus stehen.

Das Ziel dieser Bachelorarbeit jedoch soll es sein, die Daten auf Gesetzmäßigkeiten für bestimmte Bereiche einer Webseite zu untersuchen. Das Erkennen von Verhaltensmustern ist

hierbei mit inbegriffen. Vielmehr sollen hier Muster erkannt werden und diese anschließend genauer untersucht werden können.

Die Funktion, die Maus- und Augenpositionen als eine Funktion in Abhängigkeit der Zeit darzustellen, ist ein guter Ansatz. An dieser Funktion lässt sich der Abstand der beiden Bewegungen leicht ermitteln. Jedoch entsteht das Problem, dass diese Abstände nur in x- oder y-Richtung informativ wären.



## 4 Lösungskonzept

**Lösungsansatz** Zu Beginn muss ein Weg gefunden werden, die Mausaktionen eines Benutzers aufzeichnen zu können. Es muss eine Lösung gefunden werden in Echtzeit Bewegungen und Klicks zu erfassen und in eine Datenbank zu schreiben. Die Augendaten können mit Hilfe eines Eyetracker aufgezeichnet werden, welcher Eigentum der Universität Stuttgart ist. Während der Studie müssen dann parallel die Augen und Mausdaten der Probanden aufgezeichnet werden, um Synchronität zu gewährleisten.

Um den beschriebenen Visual Analytics Ansatz zu entwickeln ist es notwendig zu recherchieren, welche Visualisierungstechniken für Augen und Mausbewegungen geeignet sind. Da die Daten direkt aus dem Web kommen und die Aktionen auf der Oberfläche dieser Seiten Gegenstand der Visualisierung waren, werden webbasierte Techniken verwendet. Außerdem hat eine Webbasierte Visualisierung den Vorteil Client unabhängig zu sein. So ist es möglich die Visualisierung unabhängig von einer Anwendung, die installiert und instand gesetzt werden muss, nur mit Hilfe eines Internetzugangs und einem gängigen Webbrowser aufzurufen. Daraufhin wird ein Konzept entwickelt, dass auf webbasierten Visualisierungstechniken beruht. In den technischen Grundlagen(2.3) dieser Arbeit werden einige Externe Javascript Bibliotheken erläutert, die sehr hilfreich für die Umsetzung von Visualisierung im Web sind. Die Hauptaufgabe stellte schlussendlich die Implementierung dar, die gewährleisten sollte, es dem Analysten zu ermöglichen, die Maus- und Augendaten auf Korrelationen zu untersuchen.

### 4.1 Datenerfassung

Die Grundlage der meisten Visualisierung ist eine große Menge an Daten. Visualisierungen werden aus dem Grund verwendet, um sprichwörtlich Licht ins Dunkel zu bringen. Auch das Problem Korrelationen zwischen Maus- und Augenbewegungen zu finden, macht es erforderlich eine sehr große Menge an Daten zu sammeln. In diesem Zusammenhang macht der Einsatz eines Visual Analytics Ansatzes Sinn. Daher ist er auch Hauptbestandteil dieser Arbeit. Die Daten die visualisiert werden sollten, waren die Bewegungen des Auges und der Hand. Dazu musste man Wege finden um an diese Daten heranzukommen. Der Schwerpunkt dieser Datenerfassung, lag dabei diese zwei Bewegungen stets zeitlich synchron zu erfassen. Auch die Position von Blickpunkten und Cursorpositionen musste in Einklang gebracht werden (siehe auch Absatz4.3).

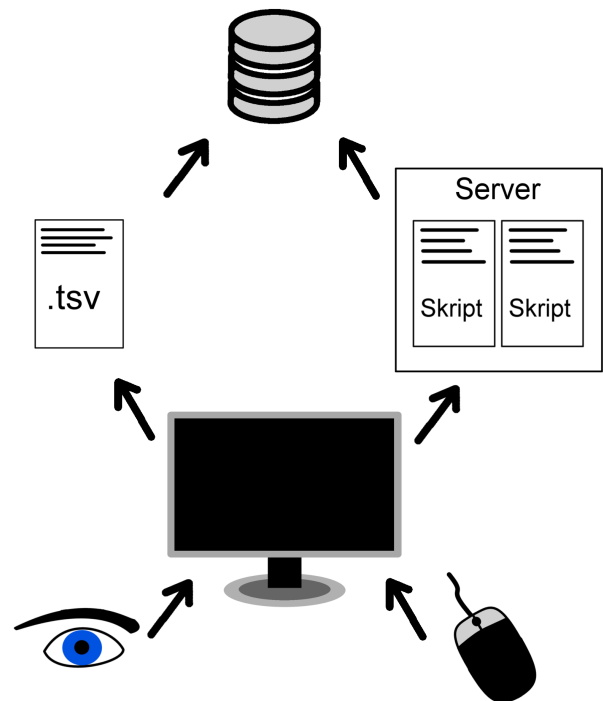


Abbildung 4.1: Abbildung, die den Datenbeschaffungsprozess verdeutlichen soll

#### 4.1.1 Benutzerstudien zur Datenerfassung

Um an ausreichende Datenmengen zu gelangen, anhand derer sich etwaige Korrelationen zwischen Maus- und Augenbewegungen untersuchen lassen, wurde durchgeführt, aus welcher die Daten erzeugt wurden. In dieser Studie wird ein Eyetracker für die Aufzeichnung der Augenbewegungen verwendet. Für die Bewegungen und Aktionen des Cursors wird ein Mousetracking Script verwendet, welches direkt in die Umgebung der Webseite eingebunden wird.

Die Studie besteht aus mehreren Szenarien. Der Begriff Szenarien beschreibt hier, eine spezielle Art von Webseite. Eine detaillierte Beschreibung zum Aufbau und Ablauf der Studie sind der Arbeit [Kru14] zu entnehmen.

#### 4.1.2 Eyetracking

Zur Aufzeichnung von Blickpunkten wurde ein Eyetracker verwendet, der kontinuierlich in Abhängigkeit von der Zeit die Blickpunkte des Auges aufzeichnen sollte. Ein Blickpunkt lieferte dabei eine große Menge an Attributen. Für die Untersuchung der Korrelation von

Maus- und Augendaten wurden nicht alle Datenausprägungen benötigt. Die folgende Aufzählung zeigt die für die Visualisierung relevanten Attribute.

- **Gaze Point X**  
Die x-Koordinate des Blickpunkts absolut zum Bildschirm
- **Gaze Point Y**  
Die x-Koordinate des Blickpunkts absolut zum Bildschirm
- **Recording Date**  
Das Datum des Tages, an dem die Aufzeichnung gemacht wurde. Das Datumsformat, war DD.MM.YYYY. Das Datum war für die Speicherung in der Datenbank und Datentransformation notwendig. Siehe auch 4.3
- **Date Timestamp**  
Der Date Timestamp wurde im Format hh:mm:ss.mmm zum Blickpunkt aufgezeichnet. Aufzeichnenfrequenz des Eyetrackers war dabei ungefähr 17 Millisekunden.
- **Event**  
Die Datenausprägung Event beschreibt den Start einer sogenannten Aktionssektion.

### 4.1.3 Mousetracking

Mit Mousetracking soll es möglich sein, nahezu alle Aktionen die der jeweilige Benutzer mit seiner Hand ausführt, aufzuzeichnen und speichern zu können. Sämtliche Daten die ein Benutzer auf der Webseite mit der Hand hinterlässt, sind relevant für die Untersuchung der Korrelation zwischen Auge und Maus. Das heißt nicht nur Klicks und Rechtsklicks sind wichtig, sondern insbesondere auch die Bewegung des Cursors auf der Oberfläche. Neben den essentiell wichtigen Mousemove- und Mouseover-Events zählt auch das Scrollen zu einem sehr wichtigen Attribut, auf das nicht verzichtet werden kann. Näheres dazu findet sich im Absatz 4.3. Im Folgenden werden nicht nur die Ausprägungen aufgelistet, welche direkt in den geforderten Visual Analytics Ansatz einfließen, sondern auch die, die für bestimmte Datenbankabfragen benötigt wurden. Zweiteres wird ebenso im Absatz 4.3 näher erläutert.

Attribute, die direkt in der Visualisierung sichtbar sind:

- **X Position**  
Die X Koordinate des Cursors absolut zur Homepage, bei auslösen eines entsprechenden Events auf der Oberfläche
- **Y Position**  
Die Y Koordinate des Cursors absolut zur Homepage, bei auslösen eines entsprechenden Events auf der Oberfläche
- **Timestamp**  
Der Timestamp der jeweiligen Events

Attribute, die für die Datentransformation oder Datenbankabfragen benötigt wurden:



- **Timestamp**  
Der Timestamp der jeweiligen Events
- **X Scroll**  
Absolute X Position der Seite beim Scroll Event
- **y Scroll**  
Absolute Y Position der Seite beim Scroll Event
- **DOM Objekt**  
Das Dom Element, auf dem das entsprechende Event registriert wurde.
- **Seitenwechsel**  
Zeitpunkt eines Seitenwechsels und dazugehörige vorherige Seite und Seite zu der gewechselt wurde.

Der Timestamp wurde sowohl für Visualisierungszwecke genutzt, als auch zur Datentransformation und Datenbankabfragen benötigt.

Die DOM Elemente (2.3.1) wurden mit allen Elternelementen bis hin zum Startelement in einer Baumstruktur in der Datenbank hinterlegt. Das Startelement war dabei die Seitenadresse. So konnten in der Vergangenheit angeklickte Elemente im Baum lokalisiert werden. Bei Aktionen auf schon vorhandenen Elementen, war es somit ausreichend einen Verweis auf die Stelle im Baum zu hinterlegen. Demzufolge konnte verhindert werden, eine Masse an unnötigen Informationen in Form von doppelten Elementen zu erzeugen. In Tabelle 4.1 ist ein kleines Beispiel zur Umsetzung der Baumstruktur in Tabellenform ersichtlich.

ID	Element	Parent	KindID
1	/shopsystem/index.php	0	0
2	html	1	0
3	body	2	0
4	div	3	1
5	div	4	2
6	a	5	0
:	:	:	:

**Tabelle 4.1:** Realisierung der Baumstruktur in einer Tabelle

## 4.2 Vorüberlegungen zur Visualisierung

Ist das Vorgehen der Datenerfassung festgelegt und geregelt, geht daraus eine abstrakte Datenstruktur für die Bewegungen des Auges und der Hand hervor. In den vorherigen Absätzen (4.1.3, 4.1.2) wurde auf die in der Studie relevanten zwei charakteristischen Bewegungen und deren Ausprägungen eingegangen. Diese Bewegung gilt es in einem Visual Analytics Ansatz zu realisieren, sodass der Analyst diese auf mögliche Korrelationen untersuchen kann. An dieser Stelle sollten Ansätze gefunden werden, welche Techniken und Methoden für die Realisierung passend sein könnten. Wie auch in vielen anderen Bereichen, nicht nur des Software Engineerings, macht es Sinn vor der eigentlichen Modellentwicklung und Implementierung, Prototypen zu entwickeln. Im Folgenden werden zwei Prototypen

vorgestellt, die schlussendlich Erkenntnisse offenbaren, welche Techniken für diese zwei charakteristischen Bewegung am Geeignetesten sind.

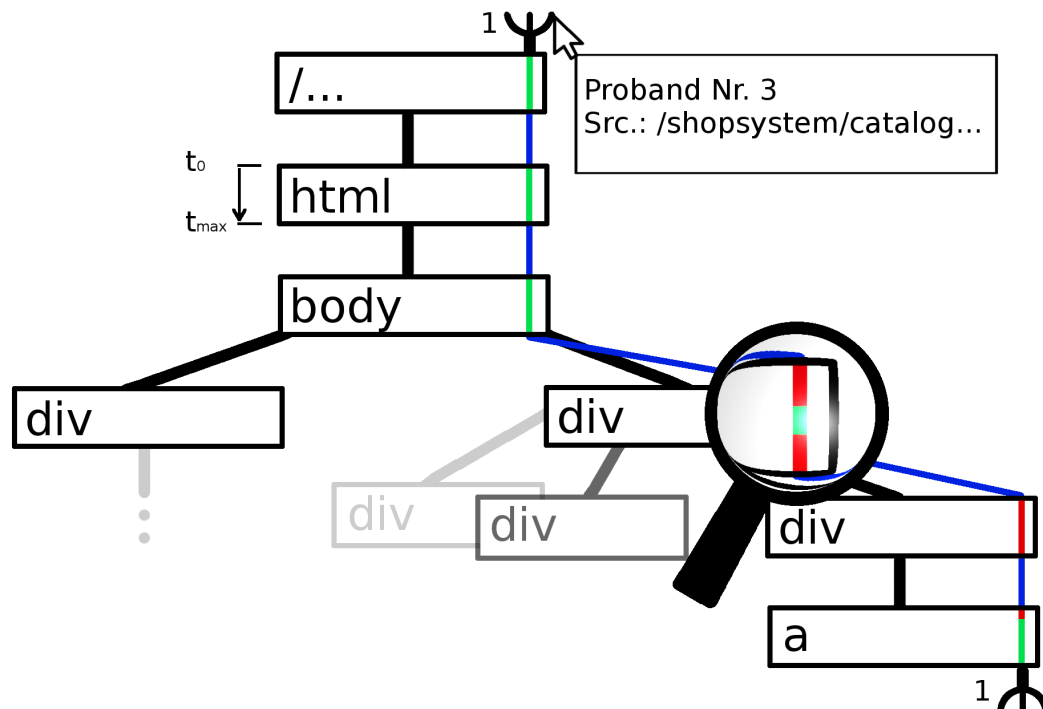
### 4.2.1 Prototyp 1

In der Benutzerstudie wurden für die jeweilige Bewegung Punkte erfasst. Mit den Koordinaten dieser Punkte ist es möglich auf die DOM Objekte der entsprechenden Webseiten zuzugreifen. Javascript bietet in diesem Zusammenhang Funktionalitäten an.

Bei diesem Prototyp sollte die Visualisierung auf der Darstellung dieser Objekte basieren. Die Grundidee dieser Darstellung war es, die DOM Objekte in einer Baumstruktur zu zeigen und die Bewegungen darauf zu projizieren. In Absatz 4.1.3 wurde die Datenspeicherung der Objekte in Tabellenform verdeutlicht. Dieser Prototyp sollte jedoch an der typischen grafischen Baumstruktur ansetzen. Aus den Datensätzen wurde daher ein Baum aller DOM Elemente erzeugt. Die Bewegungen und Aktionen von Maus und Auge sollten sich dann innerhalb dieser generierten Struktur bewegen.

Folgende Grafik 4.2 soll die Denkweise dieses Ansatzes verdeutlichen. Zu sehen ist die angesprochene Baumstruktur der DOM Objekte. Anzumerken ist hierbei, dass die Abbildung genau den Beispieldatensatz eines Baumes, der in Tabelle 4.1 ersichtlich ist, darstellt. An dieser Stelle sei als erstes auf die blaue Linie verwiesen. Diese stellt die Bewegung des Cursors eines Probanden dar. In diesem Fall kam der „Proband Nr.3“ von der Quelle „/shopsystem/catalog...“ und begann danach seine Aktionen auf der hier dargestellten Seite. Die kleinen „Krähenfuß-Symbole“, welche mit einer 1 versehen sind, sollen Verlinkungen zu anderen Seiten verdeutlichen. Bewegt man den Cursor auf diese Symbole, bekommt der Analyst Informationen über das Ziel beziehungsweise die Quelle des Seitenwechsels. Dazu bekommt er noch, die Information, welcher Proband diesen Wechsel getätigt hat.

Da es das Ziel ist, die Korrelation der beiden Bewegungen untersuchen zu können, musste ein Weg gefunden werden diese Informationen in Abhängigkeit der Zeit in die Visualisierung einzubauen. An dieser Stelle sei auf die Balken innerhalb der Elemente verwiesen. Zu erkennen ist, dass die Farbe dieser Balken nicht einheitlich ist. Dies hat den Sinn, dass der Abstand von Maus zu Auge im Farbton gespeichert ist. In dieser Abbildung beispielsweise, bedeutet ein grüner Balken, dass das Auge auf diesem Element in der näheren Umgebung war. Eine genau Skalierung welcher Farbton welchem Abstand entspricht sei erst einmal vernachlässigbar. Nun bleibt noch die Frage zu klären, wie der Zeitfaktor in diese Visualisierung eingeht. Der Eintrittspunkt einer Mausbewegung auf einem Element sei  $t_0 = \text{Timestamp}$ . Die Länge des Balkens innerhalb eines Elements beschreibt die Dauer, wie lange der Cursor auf diesem Element verweilte. Ein Balken kann demnach in dem Zeitintervall  $[t_0, \infty[$  beliebig die Farbe wechseln. Das bedeutet im Umkehrschluss, dass sich der Abstand in dieser Zeit auch verändert. Als Hilfe für den Analysten, könnte sich ein Infofenster öffnen, das die Zeit zum zugehörigen Abstandswert zeigt, falls ein solcher Balken mit dem Cursor berührt wird.



**Abbildung 4.2:** Visualisierungsansatz von Mausbewegungen als Pfad. Entfernungen der zeitlich zugehörigen Augenpositionen sind in bestimmten Farben dargestellt

#### 4.2.2 Prototyp 2

Der zweite Prototyp geht in eine anschaulichere Richtung. Er orientierte sich am Mantra von Shneidermann 2.1.3. Im Gegensatz zum ersten Prototyp bewegte sich dieser direkt auf der Oberfläche der Webseite. Dort sollte er einen Overview (vgl. Overview in 2.1) über die gesamten Daten bieten. Im Vordergrund stand bei diesem Ansatz, die zeitliche Entwicklung der Maus- und Augendaten. Visualisierungstechniken für diesen Ansatz, sollten Heatmaps und Scanpaths sein. Für beide Datentypen sollte die Möglichkeit bestehen, die Visualisierungen zu wechseln. Das heißt es konnte für die Mausbewegungen sowie für die Augenbewegungen jeweils eine der beiden Visualisierungen gewählt werden.

Die Abbildung 4.3 zeigt die Visualisierung mit einer Heatmap für die Augenbewegungen und Scanpaths für die Cursorbewegungen direkt auf der Oberfläche der jeweiligen Webseite. Der Overview zeigt die Seite, die gerade untersucht wird. Links und rechts am Rand befinden sich Szenenauswahlen (vgl. Zoom in 2.1). Die linke Auswahl besteht aus den Seiten von denen aus die aktuelle Seite angewählt wurde. Die linke Auswahl kann als eine Art Startpunkt für eine bestimmte Szene gesehen werden. In der rechten Szenenauswahl kann dann ein sich entfernender Weg gewählt werden. An dieser Stelle ist sehr wichtig zu erwähnen, dass diese Auswahl abhängig von den tatsächlich gegangenen Wegen der Probanden der aktuellen Seite sind. Wenn sich also, wie in Abbildung 4.3 zu sehen, auf der aktuell betrachteten Seite nur drei Probanden bewegt haben, sind in der rechten Szenenauswahl auch maximal drei

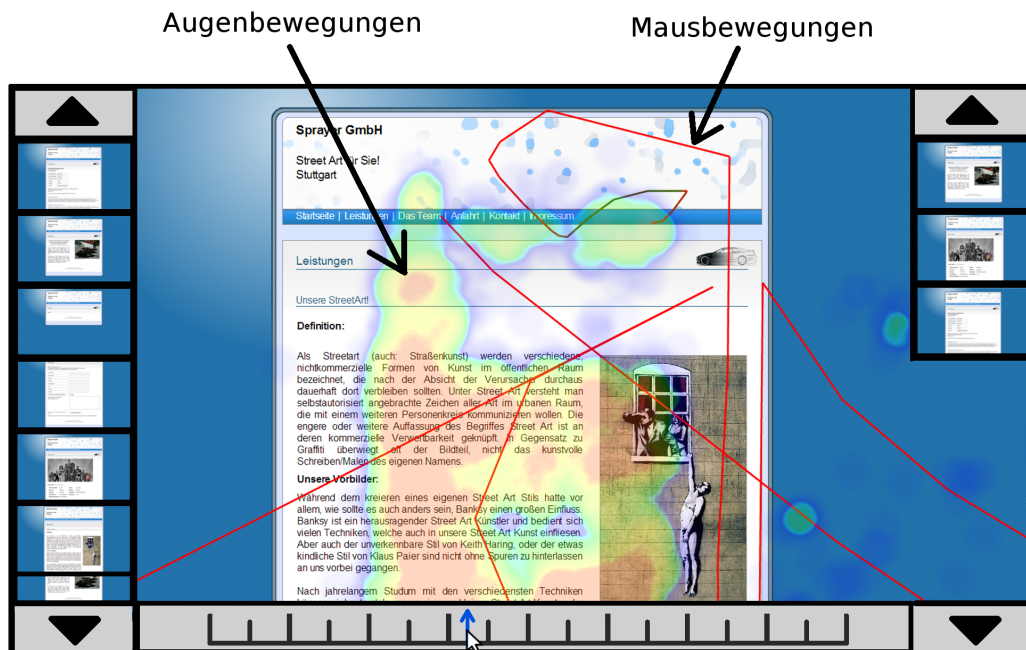


Abbildung 4.3: Übersicht der Weboberflächenbasierten Visualisierung.

mögliche Wege ersichtlich. Sind es beispielsweise 8 Probanden, dann gibt es maximal 8 mögliche Wege die Seite in dieser Szene verlassen zu haben. Am unteren Bildrand ist ein Zeitstrahl zu sehen. Dieser beginnt bei  $t_0$ . Wobei  $t_0$  die jeweilige Eintrittszeit der Probanden ist, die sich auf der Seite befanden. Anzumerken ist an dieser Stelle, dass sich jeder Proband für eine andere Dauer auf der Seite befand. Der Zeitstrahl besitzt daher die Länge des längsten Besuchs auf dieser Seite. Wenn jetzt ein Proband die Seite zu einem früheren Zeitpunkt verlässt, dann wird die jeweilige Szene in der rechten Auswahl hervorgehoben. An diese Szene wird zum Wechselzeitpunkt ein Label erstellt, in dem steht welcher Proband diesen Weg gegangen ist (vgl. Details-on-Demand in 2.1). Der Zeitschieber des Zeitstrahls ist mit der Maus einstellbar. Als weiteres Feature können einzelne Pfade mit „mouseover“ selektiert werden. Dabei werden uninteressante Pfade ausgeblendet (vgl. Filter in 2.1).

### 4.2.3 Erkenntnisse aus den Prototypen

Nach der Entwicklung von Prototypen, können Erkenntnisse für die zukünftige Arbeit gezogen werden. Neben Ansätzen, die für das weitere Entwicklungskonzept nicht in Frage kommen, gibt es meist Ansätze mit Potential. Diese können dann in das weitere Vorgehen aufgenommen und verfeinert werden.

Im Falle dieses Prototypings konnten einige Erkenntnisse gezogen werden. Bei Betrachtung beider Prototypen, wurde für die Bewegung der Maus, jeweils ein Pfad verwendet. Diese Scanpaths sind für die Augenbewegungen eher ungeeignet, da das Auge sich sehr viel mehr bewegt als die Hand. Dies würde ziemlich schnell zu Visual Clutter führen. Dennoch ist die

Visualisierungstechnik der Scanpaths ein sehr anschaulicher Weg, Bewegungsabläufe auf der Weboberfläche sichtbar zu machen. Zudem ist es sehr einfach möglich den einzelnen Wegen eine Zeitinformation zuzuteilen. Deshalb wurde diese Technik in weitere Überlegungen integriert. Hauptaugenmerk in weiteren Ansätzen war dabei die Scanpaths stets so übersichtlich wie möglich zu realisieren.

Die Baumdarstellung aus dem ersten Prototyp ist unübersichtlich. Handelt es sich um eine größere Anzahl an Probanden, dann wäre die Problem der Unübersichtlichkeit nicht zu vermeiden. Es entstünde ein großes dichtes Netz aus Pfaden, die vom Einen zum Anderen Element springen. Des Weiteren würde dem Analysten bei Betrachtung dieses Ansatzes die visuelle Verbindung zur eigentlichen Webseite fehlen. Die Idee die Information des Abstands von Auge zu Maus im Farbton des Pfades zu speichern, wurde jedoch auch als positive Erkenntnis bewertet.

Für die Visualisierung der Augenbewegungen ist die Heatmap am geeignetsten. Sie offenbart Ballungsgebiete aus der riesigen Masse an Blickpunkte und somit Interessensbereiche des Auges.

### 4.3 Datentransformation

Nach der Datenerfassung ist der Grundstein gelegt, die Grundidee der Korrelation in einer Visualisierung umzusetzen. Doch bevor die Arbeit mit den gewonnen Daten beginnen kann, muss sichergestellt werden, dass sich die Daten schon in einer Form befinden, sodass sie mit bestimmten Techniken visualisiert werden können. Der Idealfall wäre, die Rohdaten, die aus der Studie hervorgingen, ohne weiteren Aufwand direkt für den zu entwickelnden Visual Analytics Ansatz verwenden zu können. Dies war bei der Studie für diese Arbeit nicht sofort möglich.

In Abschnitten 4.1.2 und 4.1.3 sind jeweils Auflistungen zu sehen. Diese Zeigen die Datenausprägungen bezüglich Eye- und Mousetracking. Hauptaugenmerk liegt dabei, auf den Attributen Gaze Point X und Gaze Point Y. Bedenkt man, dass die Größe einer Webseite, fast nie der Größe des Bildschirms entspricht, offenbart sich ein Problem. Der Eyetracker ist in der Lage absolute Blickpunkte auf dem Bildschirm in x- und y-Richtung aufzuzeichnen. Demzufolge kann die Eingrenzungen der vom Eyetracker aufzeichnenbaren Blickpunkte wie folgt beschrieben werden.

$$(4.1) \left[ \left( \begin{array}{c} a_1 \\ a_2 \end{array} \right), \left( \begin{array}{c} b_1 \\ b_2 \end{array} \right) \right] = \{x \in \mathbb{N}^2 \mid a_1 \leq x_1 \leq b_1, a_2 \leq x_2 \leq b_2\}$$

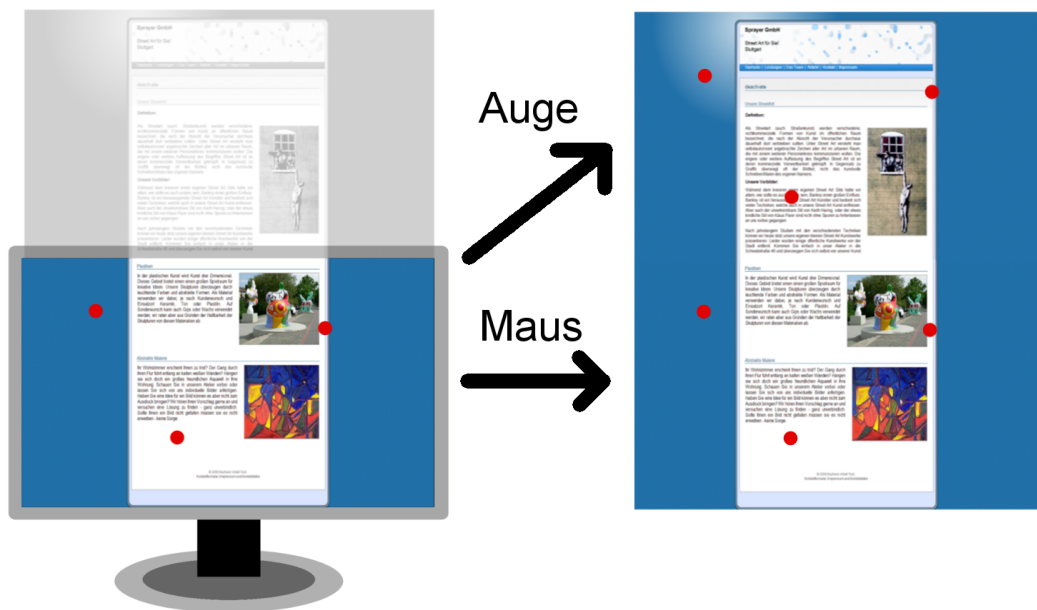
Wobei in diesem Fall  $\left( \begin{array}{c} a_1 \\ a_2 \end{array} \right)$  den Punkt  $\left( \begin{array}{c} 0 \\ 0 \end{array} \right)$ , also die linke obere Ecke des Bildschirms darstellt und  $\left( \begin{array}{c} b_1 \\ b_2 \end{array} \right)$  den Punkt  $\left( \begin{array}{c} \text{ScreenSizeX} \\ \text{ScreenSizeY} \end{array} \right)$ . Die Variable x stellt dabei den Punkt dar,

den der Eyetracker aufzeichnen kann.

Anders beim Mastracking. Dort variieren die Grenzen dynamisch mit der Größe der Webseite. Die Grenzdefinition des Mastrackings ist die selbe, wie in Formel (4.1), jedoch mit dem Unterschied das sich dort die Grenzen dynamisch mit der Größe der Webseite ändern.

Das heißt die obere Grenze  $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$  ist hier  $\begin{pmatrix} WidthWebseiteX \\ HeightWebseiteY \end{pmatrix}$ .

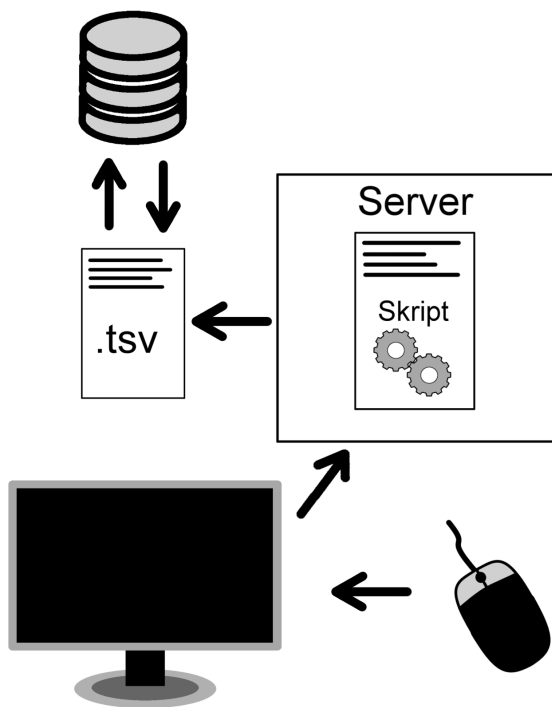
Verallgemeinernd kann man sagen, dass es kein Problem darstellt, die aus der Studie gewonnenen Daten webbasiert auf der entsprechenden Seite zu visualisieren. Auch die Blickpunkte könnte man darstellen. An dieser Stelle muss aber beachtet werden, dass diese Punkte nicht der tatsächlichen Position auf der Webseite entsprechen müssen. Man hat lediglich Blickpunkte, die sich immer in der oben beschriebenen (4.1) Grenzregion bewegen. Nicht berücksichtigt wird dabei, ob der Benutzer der Webseite eventuell zu einem Zeitpunkt gescrollt hat und alle Blickpunkte, die zeitlich nach dem Scrollen aufgenommen wurden, nicht mehr der tatsächlichen Position entsprechen.



**Abbildung 4.4:** Bildpunkte nach dem Scrollen entsprechen nicht der Realität

Erst wenn die Webseite wieder im Startzustand steht, haben die Punkte, die der Eyetracker aufgenommen hat, wieder die richtigen Koordinaten. Trotzdem ist zu beachten, dass man es den Augendaten nicht ansehen kann, wann diese wieder den tatsächlichen Punkten entsprechen. Dieses Szenario soll Abbildung 4.4 veranschaulichen. Zu einem bestimmten Zeitpunkt hat der Benutzer gescrollt. Die Augendaten werden jedoch immer vom Ausgangspunkt  $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$  aufgezeichnet und würden, wie in Abbildung 4.4 gezeigt, immer am Anfang der Seite visualisiert werden.

Um dem gerade beschriebenen Problem aus dem Weg zu gehen, beziehungsweise es zu beheben, macht man sich die Mausdaten zu nutze. Wie schon in Abschnitt 4.1 erwähnt, werden die Augen- und Mausdaten während der Studie synchron aufgezeichnet. Die Informationen, auf welcher Seite zu welchem Zeitpunkt wie viel Pixel nach unten gescrollt wurde, kann also ermittelt werden. Demzufolge war es möglich mit Hilfe eines Skripts, die gewonnenen Augendaten auf die dynamische Größe der jeweiligen Webseite abzubilden. Die Schwierigkeit lag dabei darin, die Augendaten automatisiert für jeden Benutzer einzeln zu transformieren und dabei die getätigten Seitenwechsel mit zu berücksichtigen.



**Abbildung 4.5:** Transformationsskript arbeitet mit den aufgezeichneten Mausdaten aus Datenbank

## 4.4 Finaler VA Ansatz

Das finale VA Ansatz ist der letzte Prototyp und schlussendlich auch Gegenstand der Implementierung. In Ihm vereinen sich alle Denkansätze der vorherigen Prototypen und Erkenntnisse aus der Datenerfassung. Auch an die Anforderungen aus dem Mantra von Shneidermann 2.1.3 sollte sich der finale Ansatz halten. Es wurde entschieden, das finale Konzept direkt auf der Oberfläche der Webseiten zu realisieren. Hier ist es vorgesehen, dass jede Seite explizit untersucht werden kann. Dies bedeutet es wird pro Seitenkontext(URL) eine Eigenständige Visualisierung erzeugt. Schließlich ist ja nicht nur eine Seite in der Benutzerstudie involviert, sondern mehrere. Demnach bekommt man für jede einzelne Seite Kontextbezogene Augen- und Mausdaten.

Die Grundlegenden Visualisierungstechniken waren hierbei Scanpath und Heatmap. Diese Techniken wurden für den hier vorliegenden Fall der Untersuchung von Korrelationen verfeinert, sodass eine vielfältige Filterung und zeitlichen Exploration möglich war. Da mit diesem VA Ansatz die These untersucht werden soll, ob von den Mausbewegungen eines Benutzers auf die Augenbewegungen geschlossen werden kann, steht bei diesem Lösungskonzept als Ausgangspunkt immer zuerst die Mausbewegung. Von der Mausbewegung aus will man

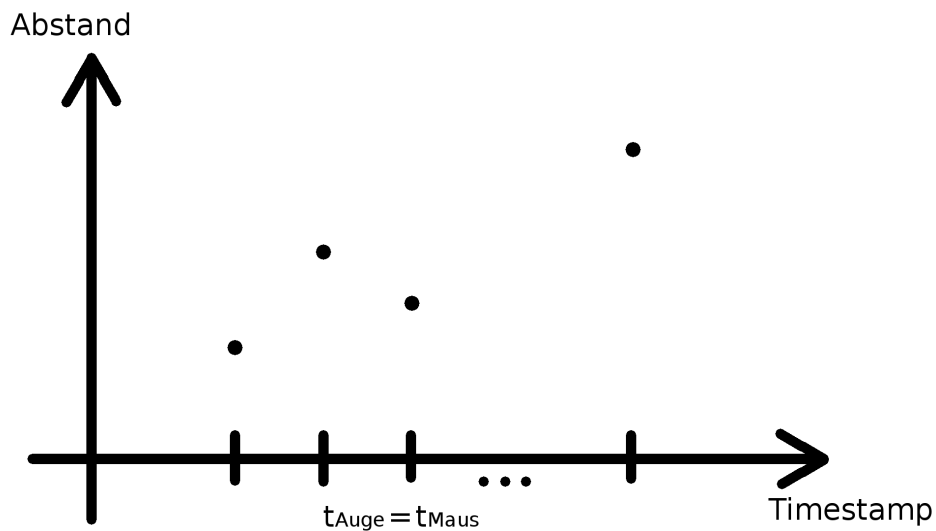


dann, Korrelationen zu den Augenbewegungen finden und dafür eventuell mögliche Gesetzmäßigkeiten ableiten. In der Arbeit [Kru14], werden solche Gesetzmäßigkeiten untersucht. Als Hilfsmittel dient dieses Konzept der VA.

### 4.4.1 Scanpath mit Abstandsinformation

Die Bewegungen der Maus werden in dieser Arbeit primär als Scanpath visualisiert. Die externe Javascript Bibliothek d3 2.3.2 hat an dieser Stelle eine einfache Methodik Pfade zu zeichnen. Nutzt man diese Funktionalität, ist es möglich Scanpaths zu generieren. Nun sollte in diesem Konzept kein gewöhnlicher Scanpath gezeichnet werden. Es sollte eine Möglichkeit gefunden werden, bei denen man die Abstandsinformation zu den Augendaten visuell in den Pfaden speichern kann. Hier wurde die Idee aus dem Prototyp aufgegriffen(4.2.2), bei der die Abstandsinformation im Farbton des Pfades gespeichert wird. Davor muss die Abstandsinformation aber erst einmal ermittelt werden. Grundansatz hierbei ist der Satz des Pythagoras. Betrachten wir alle Augen und Mausdaten eines Benutzers.

**Problem der Abstandsberechnung** Der Idealfall der Abstandsberechnung wäre es, wenn man die Maus und Augendaten synchron immer zum gleichen Zeitpunkt aufzeichnen würde. Dann würde es zu jedem Zeitpunkt an dem eine Mausbewegung aufgezeichnet wird, auch einen zugehörigen Augenpunkt geben. Es würde sich ein Diagramm ergeben, wie in Abbildung 4.6 zu sehen ist.



**Abbildung 4.6:** Der Idealfall. Zeitpunkt der Aufzeichnung von Maus- und Augenbewegung ist derselbe

Dann würde sich der Abstand, wie folgt ergeben :

$$A(\vec{m}, \vec{e}) = \sqrt{|m_x - e_x|^2 + |m_y - e_y|^2}$$

$$(4.2) \quad \vec{m}(t) = \begin{pmatrix} x \\ y \end{pmatrix}$$

mit dem Augenpunkt  $\vec{e}$  und Mauspunkt  $\vec{m}$

$$\vec{e}(t) = \begin{pmatrix} x \\ y \end{pmatrix}$$

Nun ist diese Idealsituation leider nicht gegeben. Die Mausdaten werden immer dann aufgezeichnet, wenn eine Aktion ausgeführt wird. Die Augendaten jedoch werden in einem festen Intervall kontinuierlich aufgezeichnet. Das hat zur Folge das nicht nur die Zeiten der Aufzeichnung nicht überstimmen, sondern auch noch die Anzahl der Augendaten viel größer ist.

Seien

$$T_m = \text{Menge der Timestamps der Maus}$$

$$T_e = \text{Menge der Timestamps der Augen}$$

Für diese gilt:

$$t_n \in T_m : t_n < t_{n+1}$$

$$z_n \in T_e : z_n < z_{n+1}$$

wobei  $n \in \mathbb{N}$

Dies bedeutet, dass die Timestamps aufsteigend geordnet sind. Wenn diese Timestamps der jeweilig charakteristischen Bewegung, die in der Vektordefinition 4.2 zusehen sind, eingesetzt werden, bekommen wir dazu einen Punkt mit x und y Koordinaten.

Wir wollen nun zu jedem Zeitpunkt einer Mausaktion, die dazugehörigen Augendaten finden und anschließend den Korrekten Abstand errechnen. Wir definieren nun eine Funktion wie folgt.

$$(4.3) \quad A(t_n, z_n) \begin{cases} a(t_n, z_n) & \exists z_n, z_{n+1} : z_n < t_n < z_{n+1} \\ \infty & \text{sonst} \end{cases}$$

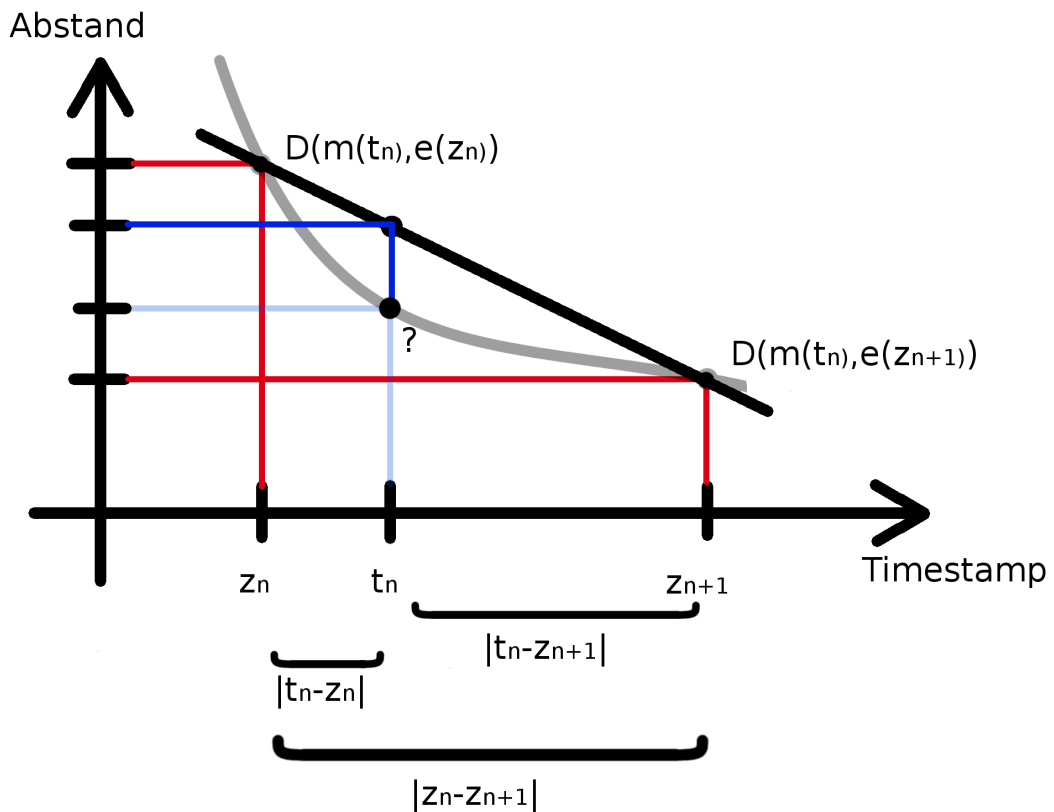
Die Funktion  $a(t_n, z_n)$  wird also mit der Zeit der Mausaktion und der Zeit einer Augenaufzeichnung. Der Augenzeitpunkt wird nur dann genommen, wenn er der zeitlich unmittelbar

vor der Mausektion erfolgte Augenpunkt ist. Bevor ein Blick auf die Funktion  $a(t_n, z_n)$  geworfen werden kann, muss noch eine andere Funktion definiert werden. An dieser Stelle kommt wieder der bekannte Satz des Pythagoras zu tragen und es können auch Parallelen zu 4.2 gezogen werden. Die unten stehende Funktion ermittelt aus den in 4.2 definierten Vektoren den Abstand.

$$(4.4) D(\vec{m}, \vec{e}) = \sqrt{|m_x - e_x|^2 + |m_y - e_y|^2}$$

Diese Gleichung geht aber immer noch davon aus, dass die zwei Bewegungen Maus und Auge zum gleichen Zeitpunkt aufgezeichnet wurden. An dieser Stelle hat der Augenpunkt aber einen anderen Zeitpunkt als die in 4.3 herausgefilterten Augendaten. Diese Augenpunkte sind unmittelbar davor bzw. danach. Nun kann es sein, dass der eine der beiden Punkte zeitlich sehr weit und der andere zeitlich sehr nah an dem Zeitpunkt der Mausektion liegt. Daher reicht es nicht aus, aus beiden Abständen von Maus zu Auge ein arithmetische Mittel zu bilden.

An dieser Stelle wäre es eine Möglichkeit durch die zwei Augenpunkte eine Gerade zu legen, um somit den Zwischenwert zu ermitteln. Die nachfolgende Grafik 4.7 soll dies veranschaulichen.



**Abbildung 4.7:** Veranschaulichung der linearen Interpolation, die zur Ermittlung des Augenabstands dient

An dieser Stelle ist es notwendig die Geradengleichung zu kennen. In unserem Fall würde diese so aussehen:

$$\frac{D(\vec{m}(t_n), \vec{e}(z_{n+1})) - D(\vec{m}(t_n), \vec{e}(z_n))}{z_{n+1} - z_n} = \frac{a(t_n, z_n) - D(\vec{m}(t_n), \vec{e}(z_n))}{(t_n - z_n)} \quad (m_0 = m_1)$$

$$(4.5) \quad a(t_n, z_n) = \frac{D(\vec{m}(t_n), \vec{e}(z_{n+1})) - D(\vec{m}(t_n), \vec{e}(z_n))}{z_{n+1} - z_n} \cdot (t_n - z_n) + D(\vec{m}(t_n), \vec{e}(z_n))$$

Damit bekommt man schlussendlich die Abstandsfunktion, die den Mausabstand beschreibt. Diese Formel stellt eine lineare Interpolation dar. Es wird eine Gerade durch die zwei Augenpunkte gelegt und der zeitlich dazugehörige auf der gerade liegende Mausabstand ermittelt. Diese Bestimmung des Abstandswertes könnte mit der Verwendung einer quadratische Interpolation noch verfeinert werden. Sie wird in Abbildung 4.7 angedeutet.

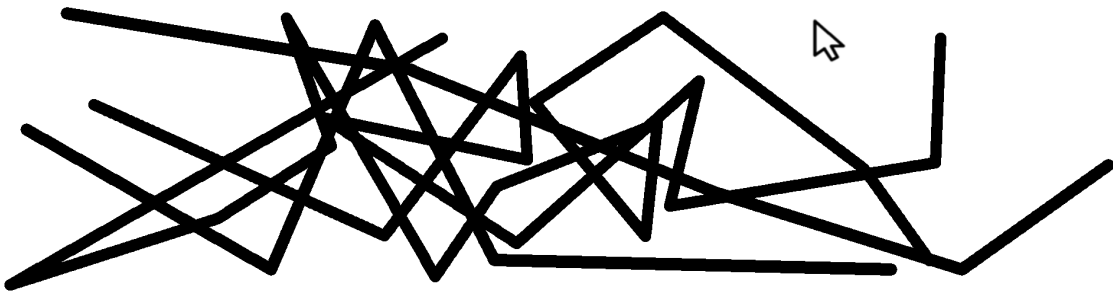
#### 4.4.2 Heatmap der Abstände

Bei dieser Heatmap sollen nur die Punkte angezeigt werden, bei denen der Abstand unter einem bestimmten Wert bleibt. Man bekommt so Bereiche auf der Homepage, die zeigen wo die Korrelation von Maus und Auge sehr stark ausgeprägt war. Für die Realisierung dieser Heatmap, müssen auch wieder Abstandswerte ermittelt werden. Die Abstandsermittlung ist im Grunde analog wie die in Abschnitt 4.4.1. Mit dem kleinen Unterschied, dass jetzt ausgehend von einem Augenpunkt zwei Mauspunkte gesucht werden. Hat man diese gefunden, wird wieder linear interpoliert.

#### 4.4.3 Filterung

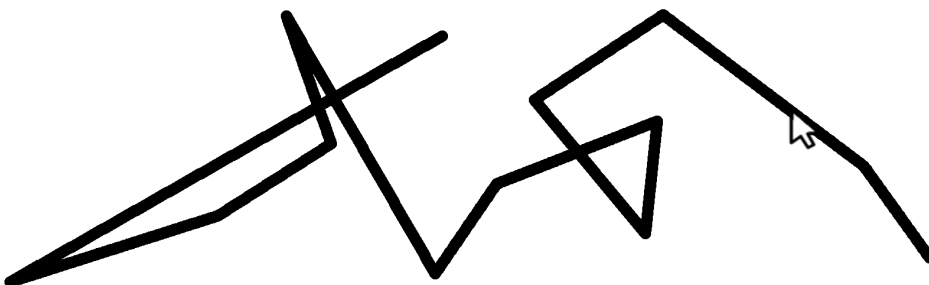
Bei dem hier zu entwickelnden Konzept ist es vorgesehen, dass die Mausbewegungen als Scanpath und die Augenbewegungen überwiegend als Heatmap dargestellt werden. Diese werden auf der Oberfläche der Webseite realisiert. Da nun in der Studie viele Seiten involviert waren, die von mehrere Benutzern besucht und bearbeitet wurden, kann dies sehr schnell zu Unübersichtlichkeit führen. Denn es werden sämtliche Daten visualisiert, die zum entsprechenden Seitenkontext passen. Das bedeutet es werden Scanpath und Heatmaps sämtlicher Daten generiert, welche beispielsweise auf der Startseite aufgezeichnet wurden. An dieser Stelle ist es für einen Analysten unabdinglich, aus diesen Daten interessante Bereiche herausfiltern zu können. Bei dieser Filterung kommt es hauptsächlich darauf an, dass der Analyst eine Art Ordnung schaffen kann, sodass es ihm möglich ist, die jeweiligen Hypothesen untersuchen zu können. Dazu wurden in diesem Konzept einige Möglichkeiten zur Filterung vorgesehen. Diese sollen im Folgenden vorgestellt werden.

**Selektion von Probanden** Mit dieser Filterungsfunktion soll interaktive Filterung zwischen Cursor und Visualisierung möglich sein. Es sollen auf der Oberfläche befindliche Pfade von Probanden mit der Maus per „mouseover“ selektiert werden können, die sich dann von den anderen abheben. Noch besser wäre es, wenn zu der Hervorhebung derjenigen Pfade, die per „mouseover“ mit dem Cursor selektiert werden, die anderen Pfade aus dem Gesamtbild ausgeblendet werden. Bei dieser Filterung liegt der Fokus auf der Betrachtung, des Weges eines einzelnen Benutzers. Die nachfolgenden Abbildung sollen das erläuterte Szenario verdeutlichen. Startpunkt ist dabei eine Menge an Scanpaths (Abbildung 4.8).



**Abbildung 4.8:** Visualisierungsbeispiel der Scanpath aller Probanden auf einer Seite. Die Maus hat noch keinen Pfad überfahren.

An dieser Stelle soll die Filterung zu dem Ergebnis aus Abbildung 4.9 führen.

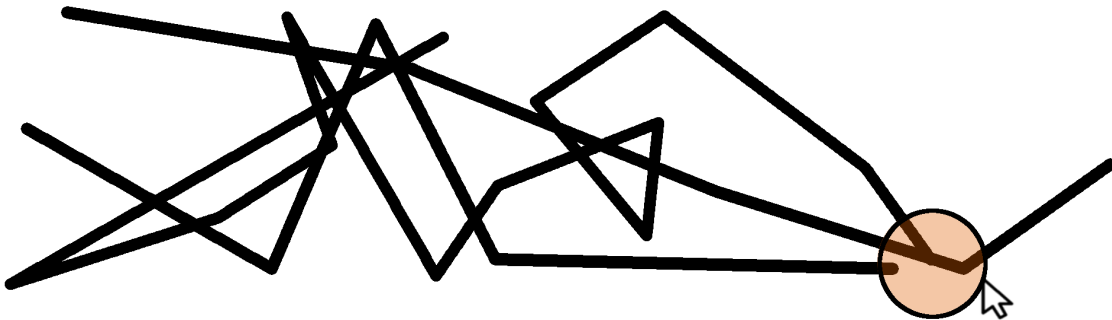


**Abbildung 4.9:** Visualisierungsbeispiel der Scanpath aller Probanden auf einer Seite. Die Maus hat einen Pfad berührt. Dieser Pfad wird selektiert und alle anderen werden ausgeblendet.

**Bereichsselektion** Bei dieser Filterung, soll der Ausgangsposition nicht ein bestimmter Pfad eines Benutzers sein, sondern ein bestimmter Bereich. Zu diesem Bereich möchte man nun wissen, welche Probanden diesen Bereich beim Besuch dieser Seite durchlaufen haben. Interesse an solch einer Einschränkungsmöglichkeit besteht, falls man eventuell gemeinsame Pfade von mehreren Benutzern untersuchen möchte. Mit einem Bereich lässt sich aber noch

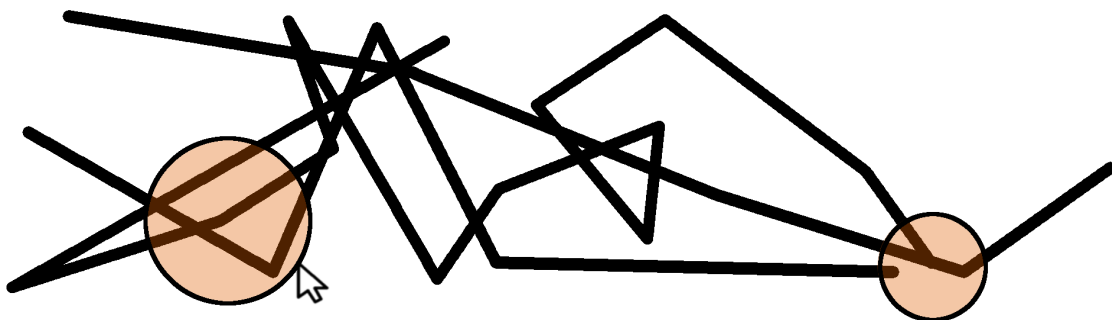
keine oder nur bedingt eine Aussage über gemeinsame Pfade mehrerer Benutzer treffen. Daher soll es an dieser Stelle möglich sein, mehrere Bereiche setzen zu können. Anschließend bekommt man zu dieser Menge an Bereichen eine dazugehörige Menge an Probanden angezeigt, die diese durchlaufen haben.

Die Nachfolgenden Abbildungen sollen die Funktion dieser Filtermöglichkeit veranschaulichen. Ausgangspunkt soll dabei die Abbildung 4.8 sein. Der Analyst zieht nun mit dem Cursor eine halbtransparente Kreisselektion auf.



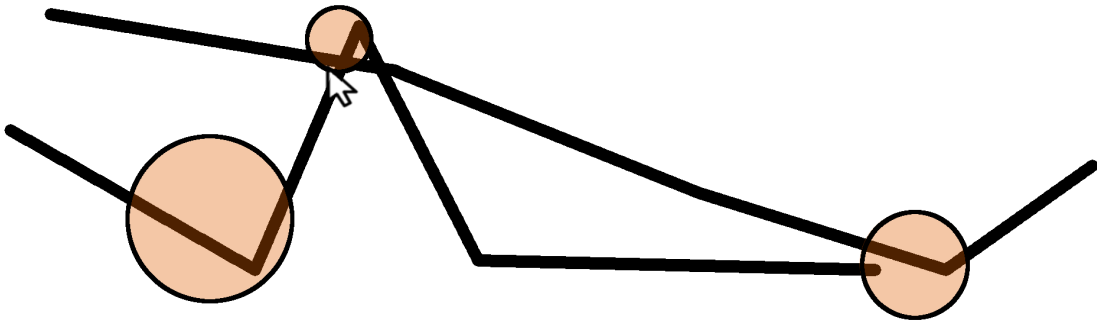
**Abbildung 4.10:** Der Analyst selektiert einen Bereich mit der Maus. Dieser selektierte Bereich beinhaltet nicht die Menge aller existierenden Pfade. Als Folge dessen wird ein Pfad aus der Selektion ausgeschlossen und ausgeblendet

In weiteren Schritten ist die Möglichkeit gegeben weitere Kreisselektionen zu setzen. Mit der besonderen Eigenschaft, dass diese nur noch auf die Menge der noch Selektierten Pfade einen Einfluss hat. In Abbildung 4.11 tätigt der Analyst eine weitere Selektion.



**Abbildung 4.11:** Visualisierungsbeispiel der Scanpath aller Probanden auf einer Seite. Die Maus Hat einen Pfad Überfahren. Dieser Pfad wird selektiert und alle anderen werden ausgeblendet.

In dieser Situation hat eine weitere Selektion keinen Einfluss auf die Menge an schon selektierten Pfaden. Es bleiben somit alle Pfade in der Selektierten Menge. Im nächsten Schritt, der in Abbildung 4.12 zu sehen ist, wird eine erneute Selektion gesetzt.



**Abbildung 4.12:** Visualisierungsbeispiel der Scanpath aller Probanden auf einer Seite. Die Maus hat einen Pfad überfahren. Dieser Pfad wird selektiert und alle anderen werden ausgeblendet.

Das Ergebnis dieser Selektion ist, dass diese Selektion die bereits existierende Menge an selektierten Pfaden verändert. Diese Abbildungen sollen verdeutlichen, wie es mit diesem Filter möglich ist, eine Art Route mehrerer Probanden sichtbar zu machen.

**Probanden Auswahl** Diese Filterungsmöglichkeit ist die einfachste. Wenn der Analyst eine gewisse Gruppe an Probanden untersuchen möchte, ist auf diesen Filter nicht zu verzichten. An dieser Stelle startet man von einem ganz anderen Standpunkt aus.

# 5 Implementierung

## 5.1 Architektur des Grundsystems

Das Grundsystem besteht aus drei Hauptkomponenten. Abbildung 5.1 zeigt diese. Der erste zu erwähnende Teil sei die Datenbank. In ihr sind die transformierten Daten gespeichert, in unserem Fall die Daten der durchgeführten Studie. Diese Daten stellen die Grundlage für die weitere Visualisierung dar. Die zweite Komponente, der Server, ist eine Schnittstelle zwischen den zu visualisierenden Daten und dem eigentlichen Visual Analytics Tool. Der Analyst soll sich auf den jeweiligen Webseiten mit einer speziellen Funktion in einen Visualisierungsbereich einwählen können. Diese Funktion soll über eine Tastenkombination auf der Seite erreichbar sein. Dies hat den Grund, dass die Webseiten die untersucht werden, teilweise wirklich im Web präsent sein können. Sonst würde die Webseite für die User, die die Webseite im normalen Gebrauch nutzen wollen, mit den störenden Visualisierungen dargestellt werden. So steht die Visualisierung nur für Klienten zur Verfügung, die sie wirklich anwählen. Der letzte Teil des Systems ist der schon angesprochene Klient. Auf dem Klient soll, dann die Visualisierung verwendbar sein. Ein Analyst wird über den Seitenkontext (den URL der Seite) sämtliche zu dieser Seite gehörenden Daten aus der Studie laden. Dazu muss er sich im Visualisierungsmodus befinden. Sind die Daten geladen können sie, auf dem Klienten visualisiert werden. In der Grafik 5.1 sind für die Funktion des Datenladens und des Visualisierens exemplarisch zwei abstrakte Klassen zu sehen. Diese sollen grob verdeutlichen, wie das Datenhandling in diesem System funktionieren soll.

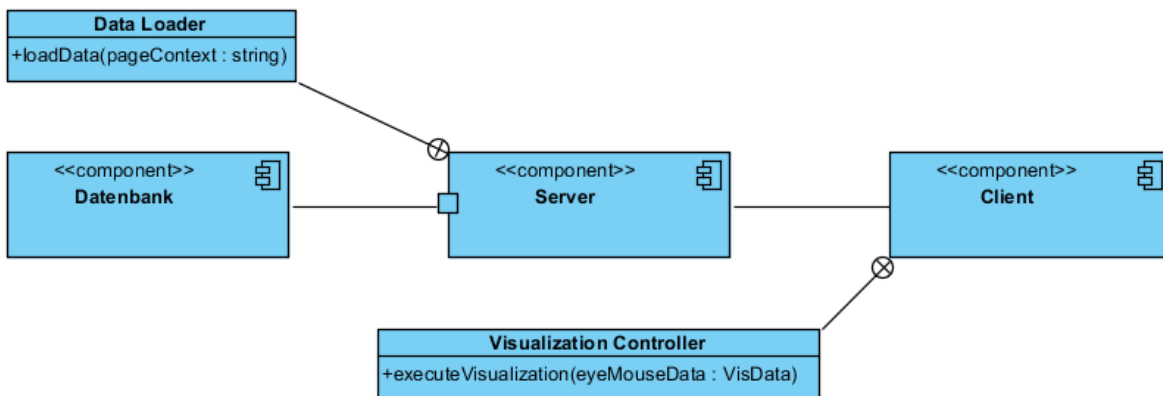


Abbildung 5.1: Gesamtsystem mit den Hauptkomponenten



**Serverseitige Prozesse** Die Aufgabe der serverseitigen Prozesse besteht darin, die Endwebseite so darzustellen, dass die visualisierten Daten betrachtet werden können. Diese Prozesskette beginnt bei der schon erwähnten Anmeldung in den Visualisierungsmodus. Ab diesem Zeitpunkt sieht der Klient in dieser Session die Webseite nur mit den visualisierten Daten auf der Oberfläche. Hat der Analyst sich erfolgreich auf der Seite angemeldet, ermittelt der Serverprozess den Seitenkontext (URL). Diese Information braucht der Prozess um die Parameter für die Datenbankabfrage setzen zu können und somit die für diese Seite passenden Daten zu laden. Im Anschluss an den Datenbeschaffungsprozess werden die Javascriptbibliotheken geladen. Diese erstellen dann die grafischen Elemente der Visualisierung auf der Oberfläche des Browsers anhand der Daten welche gerade geladen wurden.

**Klientseitige Prozesse** Nach den serverseitigen Aktionen, kommen die geladenen Javascript Bibliotheken zum Einsatz. An dieser Stelle wird schließlich die eigentliche Visualisierung generiert. Die verwendeten Visualisierungstechniken seien erst einmal unberücksichtigt. Gegenstand dieser Arbeit war es, Maus und Augendaten auf Korrelationen untersuchen zu können. Dabei musste es möglich sein, eine visuelle Analyse aufgezeichneter Eyetracking und Cursortracking Daten durchzuführen. Mit der Visualisierung musste mindestens das Filtern, Visualisieren, zeitliche Explorieren und eine statistische Auswertung der Daten möglich sein. Nach diesen Kriterien musste nun ein Modell entworfen werden das eine Realisierung eines solchen Tools beschreibt. Abbildung 5.2 soll das Modell und seine einzelnen Komponenten beschreiben.

**Modellbeschreibung** Als erstes soll auf die Klasse VisData eingegangen werden. Sie ist ein Singleton. Die Klassen MouseData und EyeData sind in folge dessen auch singleton, da sie von dieser Klasse erben. Der Grund dafür ist, dass die Daten bei jedem Aufruf einer neuen Seiten nur einmal aus der Datenbank geladen werden sollen. Von dieser Klasse soll daher auch nur eine Instanz erzeugt werden können. Diese Klassen stellen damit eine Art globalen Datensatz für Mausdaten und Augendaten dar. Da jeder Visualisierungscontroller mit diesen Daten arbeiten soll, ist zudem sichergestellt, dass jeder dieser Controller die gleiche Struktur an Daten vorfindet.

Der nächste Teil des Modells ist die Toolbar. Von hier aus sollen die einzelnen Visualisierungsfunktionalitäten angewählt werden können. Für die Zukunft sollte es die Möglichkeit geben, die Funktionalitäten unkompliziert erweitern zu können. In diesem Fall würde im Diagramm 5.2 eine neue Controller-Klasse dazukommen, die von der Toolbar instanziiert werden kann.

Die wichtigste Komponente des Modells ist der VisController. Er ist hier als ein Interface modelliert, da das Gesamtmodell einfach zu erweitern sein soll. Neue Controller sollen diesen implementieren. Die Klassen HeatmapController und SelectionController aus Abbildung 5.2 sollen Beispiele dafür darstellen. Jeder Controller arbeitet dann mit der gleichen fest definierten Datenstruktur an Maus- und Augendaten. Diese Datenstruktur entspricht dem Datentyp VisData. An diesem Punkt sollen die externen Bibliotheken zum Einsatz kommen. Sie werden in den technischen Grundlagen 2.3 dieser Arbeit erläutert und kurz erklärt. Über

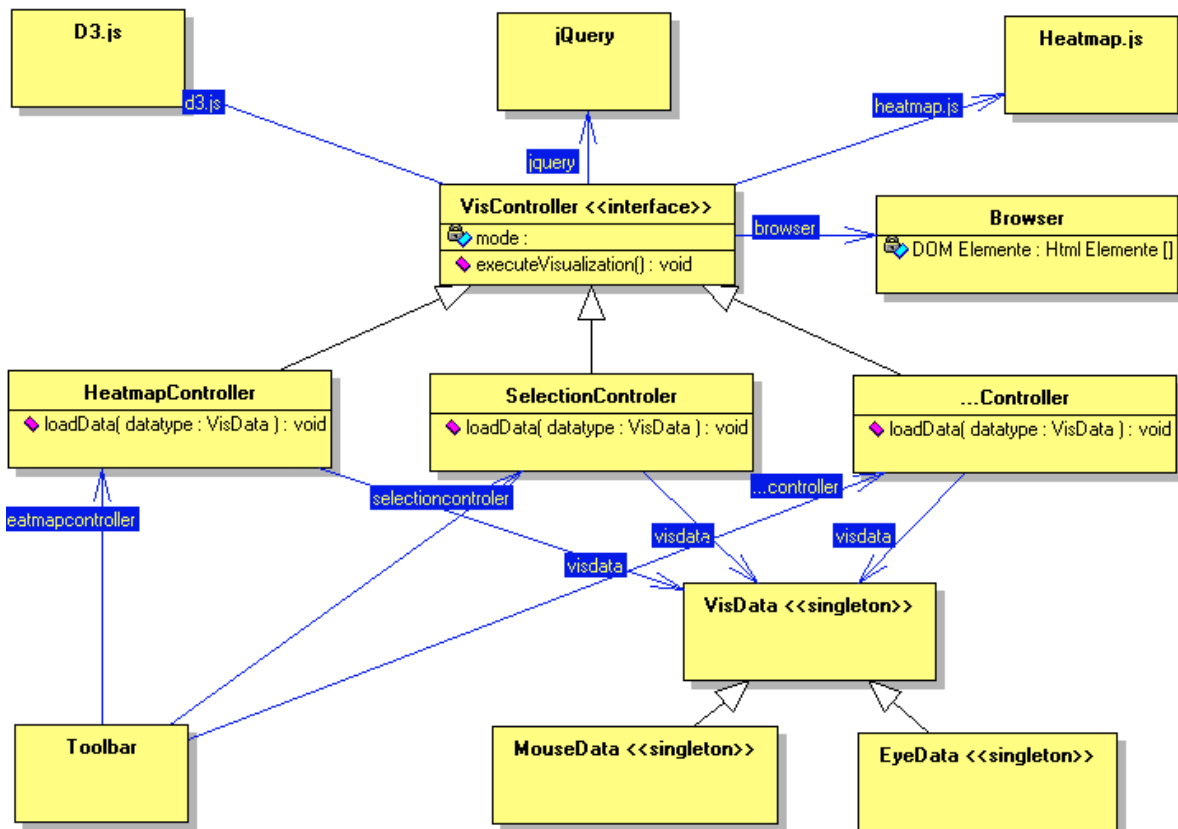


Abbildung 5.2: Modell des Visual Analytics Tools

diese Bibliotheken ist es möglich mit den Elementen der Webseite zu arbeiten, auf Basis von Daten strukturiert neue Elemente zu generieren oder eine Heatmap zu generieren.

## 5.2 Realisierung der VA

Im Folgenden werden die elementaren Realisierungen der entwickelten Ansätze aus dem Lösungskonzept vorgestellt. Es werden Ausschnitte aus dem Code des Programms präsentiert, um Funktionen und Arbeitsweisen zu verdeutlichen.

### 5.2.1 Scanpath Selektion

Bei der Realisierung der Scanpath war die externe Bibliothek D3 eine große Hilfe. Im Folgenden wird beschrieben, wie diese dabei angewendet wurde. D3 generiert hauptsächlich SVG Bereiche, um diese als Zeichenbrett für die Visualisierungen zu verwenden. Diese gezeichneten Grafiken werden auf Basis von Daten generiert. Diese Daten können in Form einer .tsv Datei extern zu Grunde liegen, oder beispielsweise zur Laufzeit des Scripts als

JSON (JavaScript Object Notation) Datentyp. Zweiteres ist bei dieser Implementierung der Fall. Das Listing 5.1 ist ein Auszug aus dem vom Server dynamisch generierten Javascript.

```
var data =
[
  { "P" : UID=2, "data" : [
    [996,608,185.48223313337,1386005173.687]
    [996,608,182.91706834146,1386005173.687]
    [806,378,144.64846656558,1386005210.08]
    ...
  ]
},
{ "P" : UID=3, "data" : [
  [987,604,1157.1451939948,1386070736.008]
  [987,604,1157.1451938904,1386070736.008]
  [978,590,1142.1838730118,1386070736.037]
  ...
}
...
];
```

**Listing 5.1:** JSON Datensatz der Mausdaten. Dieser Datensatz wird anschließend mit d3 visualisiert

Dieser Auszug beschreibt den JSON Datensatz der Mausbewegungen auf einer Seite. In dem Array Data sind vier Werte gespeichert. Die Werte data[0] und data[1] sind die x und y Koordinaten einer Mausaktion eines Probanden. Der Wert data[2] stellt den Abstandswert zu den dazugehörigen zwei zeitlich nächsten Augenbewegungen dar (Abstandermittlung 4.4.1). Der letzte Wert ist der Timestamp der Mausbewegung. In Folgenden soll darauf eingegangen werden, wie diese Daten zu den Scanpath Visualisierungen führen. Dabei ist die Information des Abstands mit inbegriffen.

Startpunkt einer jeden Visualisierung mit D3 ist es, einen Container für das Zeichenbrett, das SVG Element anzulegen oder ein bestehendes dafür zu nutzen. Das Anlegen eines solchen Containers wird hier mit jQuery implementiert. Anschließend wird mit d3 über das Attribute „id“ nach dem Element gesucht und diesem Element ein weiteres Div Element mit der „id“ „line“ hinzugefügt, auf dem die Scanpaths gezeichnet werden sollen. Das Div Element das zu Beginn angelegt wird, dient als Container für alle Visualisierungen dieses VA Tools. Anschließend daran wird die Grundlage für die einzelnen Scanpath Graphen gelegt.

```
//Anlegen eines Div Elements mit dem Attribut id und dem Wert mouseDivHover
var parentCrea = document.createElement("div");
parentCrea.setAttribute("id","mouseDivHover");
//Dieses Element soll immer die Groesse des kompletten Bildschirms haben
parentCrea.style.height = jQuery.getDocHeight();
//Hinzufuegen des Elements
document.body.appendChild(parentCrea);

//Div fuer die Scanpaths anlegen
d3.select('#mouseDivHover').append("div").attr("id", "line");
//SVG Element als Grundlage des Linienzeichnens
var graph = d3.select('#line').append("svg:svg").attr("class", "line-graph");
```

```
//SVG g Element fuer linesets
var linesGroup = graph.append("svg:g").attr("class", "lines");
```

### Listing 5.2: Da muss ich noch was machen

Nach den grundlegenden Aufgaben die für die Arbeit mit D3 bewältigt werden müssen, geht es dann darum mit Hilfe der Datensätze Linien zu zeichnen. Dazu bietet D3 eine recht simple Funktionalität an. Dieser Liniengenerator zeichnet, richtig angewendet, alle Pfade die den Daten entsprechen. Dazu muss vorher aber eine Funktion definiert werden, dass der Liniengenerator weiß wie er an die Daten kommt und er sie verwenden muss. Für die Information des Abstands, die sich in der Farbe widerspiegeln soll, wird eine Farbpalette an Farbtönen von grün nach rot definiert werden. Die Variable lineFunction definiert keine Variable, sondern eine Funktion, die jegliche Struktur an Datensätzen als Parameter entgegennehmen kann. Dieser Funktion muss nur eindeutig gesagt werden, welche Daten als x und welche als y Wert verwendet werden sollen.

```
//Liniengenerator
var lineFunction = d3.svg.line()
//anonyme Funktion arbeitet mit dem Datensatz (d) der im uebergeben wird
    .x(function(d) { return d[0]; })
    .y(function(d) { return d[1]; });
//Definition der Farbpalette
var mycolor = d3.scale.linear().domain([0,250]).range(["green","red"]);
//Zwischenspeicher fuer die Datensaeetze jedes Probanden
var linedata;
for (var i in data) {
// in linedata stehen nun die Daten von Proband i
    linedata = data[i];
    if(linedata.data != null)
//Zeichne alle Mausaktionen eines Probanden
    for(var z=0; linedata.data.length-1 > z; z++){
//Der Abstand zu zeitlich dazugehoerigen Augendaten
    var colorRange = linedata.data[z][2];
//Faerbe die Pfade mit Abstandswert <250 Pixel nah dran sind
    if(colorRange>250)
        colorRange=250;
//fuege einen Pfad hinzu mit den Daten...
    linesGroup
        .append("path")
        .attr("d", lineFunction(
//x und y des Probanden
            new Array(linedata.data[z],linedata.data[z+1])))
//speichern der ProbandenNr und des Timestamps in den Elementen
        .attr("class", "Proband_"+linedata.P)
        .attr("id", linedata.data[z][3])
// Farbe des Pfades setzen
        .attr("stroke", mycolor(colorRange))
        ...
    }
}
```

**Listing 5.3:** Datensätze der Mausbewegungen werden mit d3 als Pfade gezeichnet und je nach Abstandswert eingefärbt

Nun sind die Pfade aller Benutzer generiert. Für die Selektion dieser Pfade wurden an den Pfadelementen Event-Handler angemeldet. Diese Event-Handler behandeln die Events „mouseover“, „mousemove“ und „mousedown“. Diese Events sind für die Pfadselektion unabdinglich. Der Handler für die „mouseover“-Events leitet die Selektion der Pfade ein, die überfahren werden. Dazu ein kleiner Code Snippet 5.4, der die Funktionsweise hinreichend erklärt.

```
// Event Handler fuer mouse over events auf allen pfaeden
function pathMouseOver(){
[...]
```

```
var pathSelection = d3.selectAll("path");
var paths = pathSelection[0];
[...]
```

```
//Speicher fuer temporaer selektierte Pfade
tempSelPaths = [];
// hervorheben der Pfadteile die zu dem jeweiligen Probanden passen
// komplettes ausblenden der anderen userpfade
for(var i=0 ; i< paths.length ; i++){
    if(paths[i].className.baseVal ==
        d3.event.currentTarget.className.baseVal ){
        d3.select(paths[i]).attr("stroke-width", "5");
        tempSelPaths.push(paths[i]);
    }else{
        d3.select(paths[i]).style({'visibility': 'hidden'});
    }
}
tempSelPaths = ignorePointPaths(tempSelPaths);
[...]
```

**Listing 5.4:** Event-Handler regelt die Scanpathselektion bei Überfahren mit der Maus

Beim „mousemove“-Event auf einem Pfad erscheint ein kleines Infofenster über die Eigenschaften des Pfads. Eigenschaften wie der Timestamp und die Probandennummer. Das „mousedown“-Event, ist hauptsächlich dafür da, die Pfade in der Selektion zu behalten, sodass es möglich ist, mit dem Cursor an eine andere Stelle zu fahren, ohne dass die Selektion aufgehoben wird. Des Weiteren werden beim Klick auf einen Pfad an den einzelnen Mausaktionskoordinaten ein Punkt gezeichnet, um Zeitintervalle zu veranschaulichen. Der Startpunkt eines jeden Pfades, wird etwas dicker gezeichnet, dass man den Start des Scanpath erkennen kann.

Ein weiteres wichtiges Feature ist, dass beim Überfahren eines fest selektierten Pfades, die dazugehörigen Augenpunkte die zu dieser Zeit auftraten, gezeichnet werden. Dieses Feature ist wichtig, um die Daten zeitlich zu explorieren.



**Abbildung 5.3:** Abfahren des Pfades mit der Maus, zeichnet zeitlich dazu-gehörige Punkte der Augen-



**Abbildung 5.4:** Auf dem nächsten Pfad wurde vor den nächsten Mausaktion ein großer Teil des Textes gelesen

### 5.2.2 Bereichselektion

Bei der Bereichselektion muss man nun eine Interaktion des Cursors mit der Oberfläche behandeln. Wie auch in anderen Programmiersprachen, gibt es in Javascript dafür Event Handler. Auch an dieser Stelle bietet d3 etwas Eigenes an. Wieder benötigt man das am Anfang angelegt Div Element, das für die Visualisierungen erstellt wurde. In einem nächsten Schritt werden diesem Event-Handler zugeordnet. Vorerst einer, der Events behandelt, die davon ausgelöst werden, falls sich die Maus Taste vom gedrückten Status in den neutralen Status ändert. Ein weiterer, der Events behandelt die ausgelöst werden, falls der Maus Button gedrückt wird.

Für beide Event- Klassen sind Event-Handler definiert. Bei „mousedown“-Events werden die aktuellen x- und y-Koordinaten der Maus ermittelt und dort anschließend ein Kreis gezeichnet. Vorerst mit Radius 0. Direkt danach wird auf dem selben Div Element, auf dem sich alles bewegt, ein weiterer Event-Handler registriert. Dieser wird bei parallelem Maustastendruck und Mausbewegung ausgelöst. Die Definition der Funktion, die für die Behandlung der „mousemove“-Events zuständig ist, sieht die schon angesprochene dynamische Interaktion zwischen den Bewegungen des Cursors und die darauffolgende Reaktion des Kreiszeichnens vor. Der Radius wird dabei dynamisch mit jedem „mousemove“-Event neu ermittelt und der Kreis im Anschluss neu gezeichnet.

```
//Anmelden der Event-Handler auf dem Visualisierungsdiv
var vis = d3.select('#mouseDivHover')
    .on("mousedown", mousedown)
    .on("mouseup", mouseup);

//Event Handler fuer "mousedown"-Events auf div
```

## 5 Implementierung

---

```
//das ueber kompletten Body geht ( Kreisselektion )
function mousedown() {
//Aktuelle Mausposition
var m = d3.mouse(this);
startx = m[0]; starty = m[1];

//Zeichnen des Initialkreises mit Radius 0
line = circleSvg.append("circle")
    .attr("class", "drawRec")
    .attr("cx", m[0])
    .attr("cy", m[1])
    .attr("r", 0);

//Anschliessendes Anmelden des Event-Handler fuer "mousemove"-Events
vis.on("mousemove", mousemove);
}

//Event Handler fuer "mousemove"-Events auf div
//das ueber kompletten Body geht ( Kreisselektion )
function mousemove() {
[...]

//Ermittlung der aktuellen Mauskoordinaten
//nach eingetroffenen "mousemove"-Event
var m = d3.mouse(this);

//Berechnen des Radius mit der Kreisformel  $r^2 = \sqrt{x^2 + y^2}$ 
r = Math.sqrt(Math.pow(startx-m[0]) + (Math.pow(starty-m[1])));
line.attr("r", r);
[...]
}
```

**Listing 5.5:** Registrierung der Event-Handler um die Kreisselektion realisieren zu können. Implementelle Umsetzung der dynamischen Interaktion zwischen Oberfläche und Cursor

Nun fehlt noch die Behandlung der Events, die ausgelöst werden, falls der Tastendruck wieder aufgelöst wird. Mit in die Behandlung dieser Events kommt das anschließende Prüfen, ob sich in dieser Kreisregion Mauseaufzeichnungen befanden.

Zum Verständnis der folgenden Codestelle 5.6, muss die Funktionsweise der Funktionen `resetSelProbs()`, `notifyProbs()` und `showSelectedPaths(manuelSelProb,true)` erläutert werden. Es existiert ein globales String Array, in das die Probanden gespeichert werden, deren Pfade durch diese Kreisselektion laufen. Die Funktion `notifyProbs()` ist dabei dafür zuständig, die für die Selektion in Frage kommenden Probanden im Array anzumelden. Die Funktion `resetSelProbs()` ist das Gegenstück zu dieser Funktion. Sie löscht die in Frage kommenden Probanden der vorherigen Selektion. Für die Mouse Punkte aller Benutzer der Seite, gibt es ebenfalls ein Array, in das eine fest definierte Datenstruktur für Punkte entworfen wurde. Diese Datenstruktur besitzt unter anderem die Attribute „visibility“ und „path\_class“. Beide dieser Attribute werden für die Filterung der anzuzeigenden Punkte benötigt. Bei jeder vorherigen Kreisselektion werden die Datensätze der Mauspunkte aktualisiert, sodass diese

bei erneuten Selektionen durch die Abfrage der Sichtbarkeit fallen. Dies geschieht beim Ausführen der Funktion `showSelectedPaths(manuelSelProb,true)`. Des Weiteren, wie der Name der Funktion verrät, werden hier die selektierten Pfade der gefilterten Probanden angezeigt.

```
//Event Handler fuer mouse up Events auf div
//das ueber kompletten Body geht ( Kreisselektion )
function mouseup() {
[...]
//zuruecksetzen der selektierten Probanden
resetManSelProbs();
vis.on("mousemove", null);
for(var i=0 ; i< siteMousePts.length ; i++){
//Ermitteln der x und y Koordinaten der Mauspositionen
    xAbsolute = parseFloat(siteMousePts[i].data[0]);
    yAbsolute = parseFloat(siteMousePts[i].data[1]);

// y^2 < r^2 - x^2 && x^2 < r^2 - y^2
//startx und starty sind die Koordinaten des Kreismittelpunkts
    if((Math.pow(Math.abs(starty -yAbsolute))
        < Math.pow(r) - Math.pow(Math.abs(startx -xAbsolute)))
        &&(Math.pow(Math.abs(startx -xAbsolute))
        < Math.pow(r) - Math.pow(Math.abs(starty -yAbsolute))))){
        if(siteMousePts[i].visibility == "visible"){
            notifyProbs(siteMousePts[i].path_class);
        }
    }
}
showSelectedPaths(manuelSelProb,true);
[...]
```

**Listing 5.6:** Implementelle Umsetzung der Pfade die durch den gezeichneten Kreis definierten Bereich an Koordinaten

### 5.2.3 Heatmap des Auge-Maus-Abstands

Das zeichnen einer Heatmap, übernimmt eine externe Javascriptbibliothek 2.3.2. Diese macht es ziemlich simpel eine Heatmap zu zeichnen. In dem Listing 2.1 ist ein Beispiel aufgeführt, wie die Externe Bibliothek verwendet wird. Es wird auch ein Datensatz gezeigt, den die Heatmap erwartet. Auch in der Implementierung dieses Konzepts, wurde ein Datensatz aus den Augendaten erzeugt, mit dem die Heatmap arbeiten konnte. Eine zweite Möglichkeit eine Heatmap mit dieser Bibliothek zu zeichnen, ist es, die Punkte einzeln zeichnen zu lassen und somit nach und nach die Heatmap als Gesamtbild zu erzeugen. Dies war auch bei dem Erzeugen, der Abstandsheatmap der Fall. Bevor überhaupt mit der Heatmapgenerierung begonnen werden konnte, mussten die Augendaten so gefiltert werden, dass nur die gezeichnet werden, welche in einem bestimmten Abstand zu den entsprechenden



Mausdaten stehen. Diese Abstandsermittlung wurde konzeptionell in Abschnitt 4.4.2 erläutert. Die Funktion `calcHeatAbstand(eye, mouse)` berechnet den Abstand zwischen den Augenpunkten eines Probanden und den dazugehörigen Mauspunkten. Die Funktion `filterDistanceArray(newData)` gibt ein Array an Augendaten zurück, die nur einem gewissen Abstand entsprechen.

```
[...]
//Speicher fuer die Augenpunkte mit zusaetzlich gespeichertem Abstandswert
    var newData = new Array();
// (data = globalen Mausdaten)
    for (var i in data) {
        probArray = new Array();
//Zuordnen aller Augenpunkte eines Probanden und speichern in probArray
        for(var a = 0; a < globalEyeJSONData.length;a++){
            if(globalEyeJSONData[a].user == data[i].P){
                probArray.push(globalEyeJSONData[a]);
            }
        }
//Augenpunkte des Probanden an Stelle i mit berechnetem Distanzwert
        newData.push(calcHeatAbstand(probArray, data[i].data));
    }
//nur Punkte zulassen welche einen gewissen Abstand einhalten
    var points = filterDistanceArray(newData);
    for(var i=0; i < points.length ; i++){
//Zeichnen jedes Punktes
        distHeatMapSet.store.addDataPoint(points[i][0],points[i][1]);
    }
[...]
```

**Listing 5.7:** Implementelle Erzeugung der Abstandsheatmap

### 5.2.4 Probandselektion

Der einzige Aufwand bei der Implementierung der Probandenselektion, war es die Toolbar um die Buttons zu erweitern um die jeweiligen Probanden auswählen zu können. Die Aktionen des Filterns nach Pfaden für spezielle Probanden konnte von den Implementierungen der anderen Selektionen weitestgehend übernommen werden.

## 6 Fallstudie

In diesem Kapitel wird beschrieben, wie man mit dem entwickelten Konzept Eye- und Mastrackingdaten auf Korrelation untersuchen kann. Zu Beginn wird eine Menge an Hypothesen aufgelistet, welche dann Bestandteil der Fallstudie der Funktionsweise des VA Konzepts sein sollen. Bezugssystem ist hier aus sich der Funktionen des Tools, was bedeutet, dass von den Funktionen aus beschrieben wird, was mit dieser untersucht werden kann und wie es untersucht wird. Dazu soll anschließend ein Ablauf dazu geschildert werden.

### 6.1 Hypothesenuntersuchung

Die untenstehende Tabelle 6.1 zeigt die untersuchten Hypothesen. Diese wurden der Arbeit [Kru14] entnommen.

- Hypothese 1: Legt der Benutzer die Maus beim Lesen beiseite, ist die Verweildauer der Maus ähnlich der Lesedauer des Textes
- Hypothese 2: Bewegt der Benutzer während dem Lesen die Maus, sind Blickpunkt und Maus nahe beieinander
- Hypothese 3: Durch das Scroll-Verhalten lässt sich auf die Lese-Intensität des Besuchers schließen
- Hypothese 4: Beim Verwenden von Eingabefeldern, Buttons, Menüelementen und Hyperlinks sind Blickpunkt und Maus nahe beieinander
- Hypothese 5: Wird während dem Lesen eine Grafik betrachtet, wird auch die Maus auf die Grafik gezogen

**Tabelle 6.1:** Auflistung der Hypothesen, die mit diesem VA Tool untersucht wurden. Die vollständige Beschreibung der Hypothesen und der Ergebnisse der Untersuchung sind der Arbeit [Kru14] zu entnehmen

#### 6.1.1 Scanpath Selektion

Diese Funktionalität war bei der Analyse der Hypothese 1, Hypothese 2 und Hypothese 5 hilfreich um die Korrelation zu untersuchen.

**Zu untersuchende Bereiche** Die Hauptsächlich zu untersuchenden Elemente waren hierbei, Menüs, Hyperlinks, Buttons, Formularelemente, Grafiken und Texte. Dabei wurden speziell die Wege der Benutzer auf den bestimmten Elementen untersucht. An dieser Stelle konnten auch die zeitlichen Abläufe sichtbar gemacht werden.

**Ablauf der Analyse** Der Analyst tätigt mit der Maus, verschiedene Selektionen an Pfaden. Beim Überfahren dieser kann er den Zeitpunkt der Aktion einsehen. Fokussiert er diesen Pfad hat er zudem auch die Möglichkeit, die zeitlich dazugehörigen Augendaten in Form von Punkten sichtbar zu machen.



Abbildung 6.1: Zu einer Mausbewegung zugehörige Augenbewegungen

### 6.1.2 Filterung nach Probanden

Die Filterung nach Probanden war in Hypothese 2 nützlich. Es wurden Pfade untersucht, die sich für eine langen Zeit über dem Text befanden. Nach Hypothese 2 soll untersucht werden, ob die Augen dieser Probanden auch auf dem Text waren.

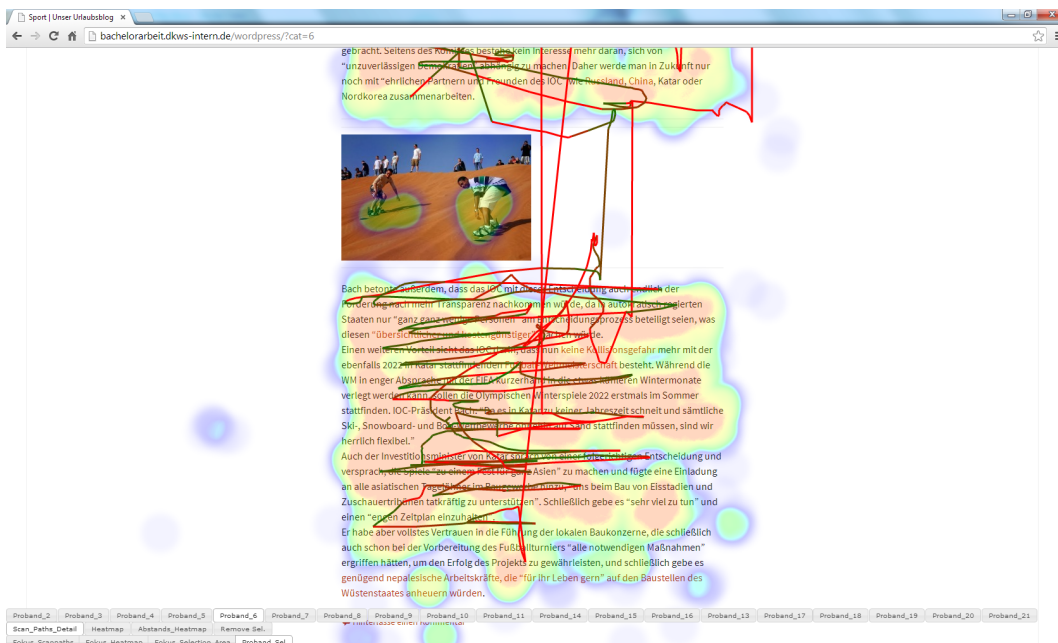
**Zu untersuchende Bereiche** Diese Filtermöglichkeit ist hilfreich, wenn eine bestimmte Personengruppe im Detail untersucht werden soll. Man kann sich bestimmte Personengruppen zusammenstellen und das Verhalten dieser analysieren. Auch wenn ein Proband in besonderem Interesse steht, kann diese Funktion verwendet werden. Es wird nach der Auswahl der Personen eine Gesamtvisualisierung dieser Probanden in Form von Heatmap und Scanpath generiert.

**Ablauf der Analyse** Hier soll das Leseverhalten von Probanden untersucht werden. Dabei muss der Analyst vorher Probanden finden, bei denen die Mausbewegungen sich überwiegend auf dem Text befanden. Das heißt, der Pfad sich im Textfluss befindet.

Aktuer	Aktion
Analyst	Benutzt die Scanpath Selektion, um Probanden mit auffälligen Pfaden zu finden
Visualisierung	Zeigt Infofenster der im Fokus stehenden Pfade
Analyst	Wechselt zur Filterung nach Probanden
Visualisierung	Bietet Checkboxen für jeden Probanden an, für den auf der Seite Daten vorhanden sind an
Analyst	Wählt den eben ermittelten Probanden aus
Visualisierung	Zeigt nur den Probanden der ausgewählt wurde. Angezeigt werden Augenbewegung als Heatmap und Mausbewegung als Scanpath.

**Tabelle 6.2:** Use Case zum Beispielszenario für die Filterung nach Probanden. Die Vorbedingung ist, dass der Analyst die Visualisierung einsatzbereit vorliegen hat.

Man beginnt bei der Selektion von einzelnen Pfaden, die sich über dem Text befinden. Daraus bekommt man die Information, welcher Proband dies war. Nun wechselt man zur Probanden Selektion und markiert alle Probanden, die in dem Interesse der Hypothese stehen. Die Selektion ist über Buttons in der Toolbar möglich.



**Abbildung 6.2:** Selektion eines Probanden

### 6.1.3 Bereichselektion

Diese Selektion war in Bezug auf die Arbeit [Kru14] nicht relevant für die Untersuchung der Korrelation von Maus und Auge. Für diese Selektion war die Menge an zu analysierenden User zu klein. Dieser Filter ist in der Lage, Hauptstöße an Mauspfaden von vielen Benutzern sichtbar zu machen. Für die charakteristischen Daten, die aus der Benutzerstudie zu dieser Arbeit durchgeführt wurde, offenbarten keine dieser sich gleichenden Massenpfade.

**Ablauf der Analyse** Das folgende Beispielszenario soll zeigen, wie die Bereichsselektion eingesetzt werden könnte. Dieses Szenario war teil der Benutzerstudie, die in der Arbeit von F.Krüger [Kru14] näher erläutert wird.

Lesen Sie die Überschrift und beantworten Sie die darin enthaltene Frage.

Untersucht werden soll, welche Probanden beim Lesen der Frage die Maus mitbenutzen. Anschließend soll untersucht werden, ob die in Frage kommenden Probanden, mit der Maus das Ergebnis speziell markieren oder darauf zeigen.

Aktuer	Aktion
Analyst	Tätigt auf dem Text, in dem die Fragestellung steht, eine Bereichsselektion.
Visualisierung	Selektiert die Menge an Pfaden, die durch den Selektionsbereich verlaufen
Visualisierung	Blendet alle Pfade, die nicht durch den Selektionsbereich verlaufen, aus
Analyst	Tätigt eine Selektion auf dem Ergebnissymbol
Visualisierung	Selektiert die Menge an Pfaden, die durch den Selektionsbereich verlaufen
Visualisierung	Blendet alle Pfade, die nicht durch den Selektionsbereich verlaufen, aus
Analyst	Erkennt an der Färbung des Pfades, ob die Augen in unmittelbarer Nähe befand

**Tabelle 6.3:** Use Case zum Beispielszenario für die Bereichsselektion. Die Vorbedingung ist, dass der Analyst die Visualisierung einsatzbereit vorliegen hat.

### 6.1.4 Abstands Heatmap

Die Abstandsheatmap ist für die Analyse der Korrelation sehr geeignet. Es macht Bereiche sichtbar, bei den Interessengebiete von Auge und Cursor sich zu einer bestimmten Zeit annähernd gleichen. Diese Funktionalität war bei der Analyse der Hypothese 4 und Hypothese 5 hilfreich um die Korrelation zu untersuchen.

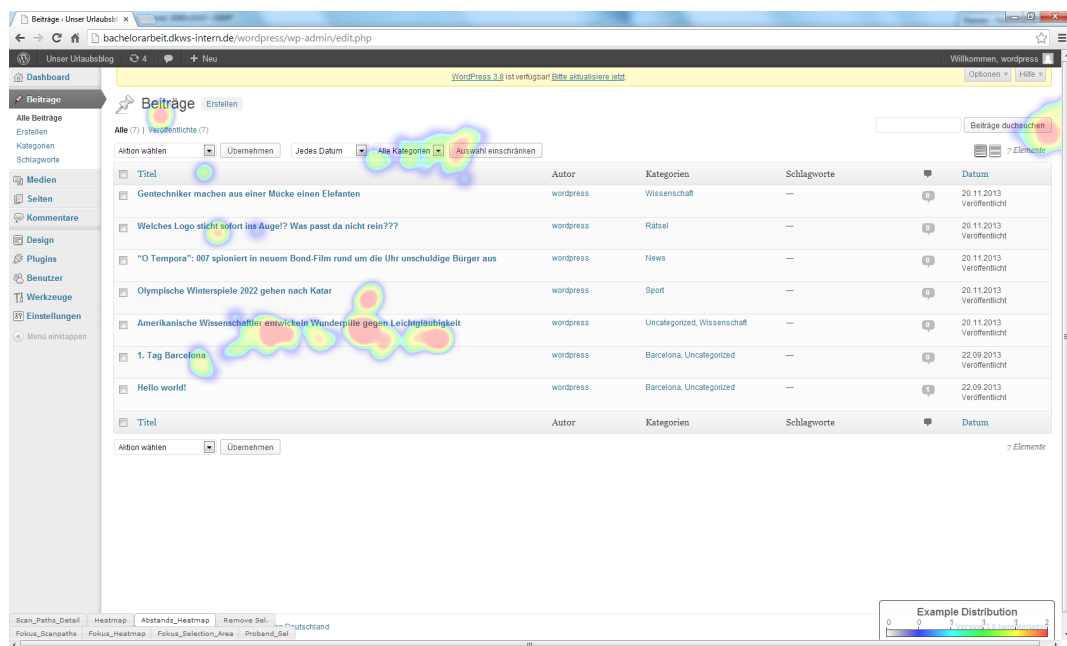
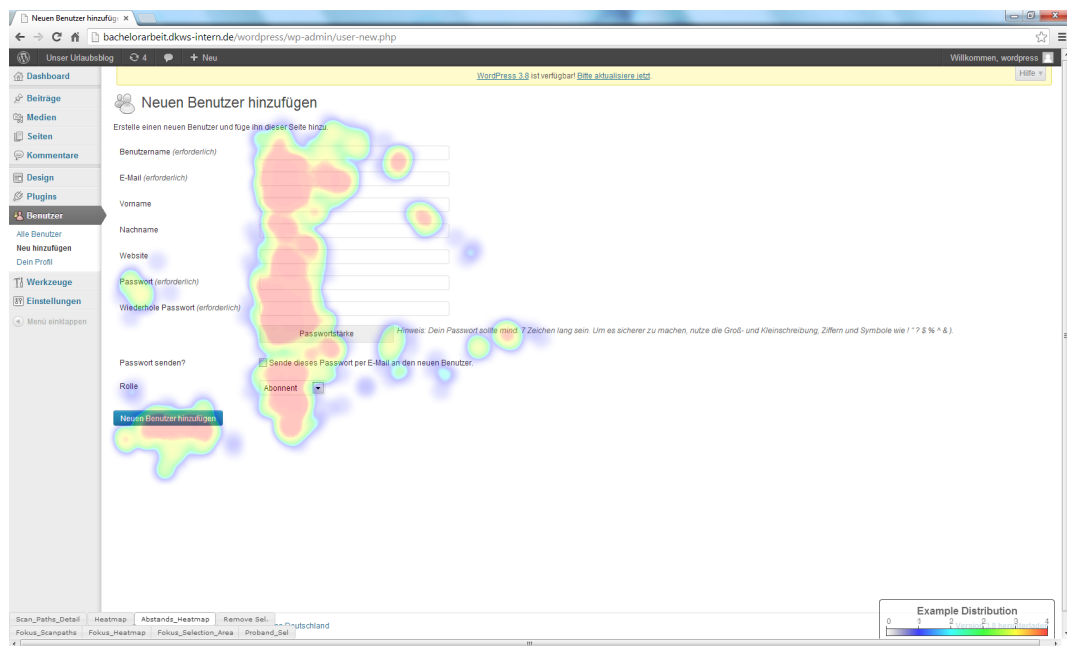


**Abbildung 6.3:** Setzen zweier Kreisselektionen auf der Oberfläche, einer in der Studie untersuchten Webseiten. Hier wird zum Beispiel untersucht, welche Probanden beim Lesen der Frage mit der Maus über dem Text waren und anschließend die Lösung auch mit Hilfe der Hand gesucht haben.

**Zu untersuchende Bereiche** Bei dieser Heatmap gibt es keine speziellen Bereiche, bei denen man sagen könnte, dass sie besonders gut mit ihr zu untersuchen sind. Sie ist immer auf die komplette Seite anwendbar. Dabei offenbaren sich auf der ganzen Oberfläche Interessenfelder. Jedoch war bei der Studie, die im Rahmen dieser Arbeit durchgeführt wurde, auffällig, dass gerade bei Menüs und Formularen ein auffälliges Muster direkt auf den Elementen entstand.

**Ablauf der Analyse** An dieser Stelle sei eine Ablaufbeschreibung vernachlässigt, da man die Heatmap mit einem Buttondruck generieren kann. Im Folgenden wird aber eine Grafik aufgeführt die das Erscheinungsbild dieser Heatmap zeigen soll.

## 6 Fallstudie



**Abbildung 6.4:** Auf dieser Seite waren die Probanden an der Stelle der Studie, an der sie einen Beitrag suchen mussten. An Hand der Abbildung kann man ablesen, welcher Beitrag gesucht war. Des weiteren sind in Betracht gezogene Alternativmöglichkeiten zu erkennen, falls der Beitrag nicht sofort ersichtlich war. Das Suchfeld oben rechts war dabei, die Alternative

## 7 Zusammenfassung und Ausblick

In diesem Kapitel wird der gesamte Ablauf dieser Ausarbeitung zusammengefasst. Dabei werden Schwerpunkte der einzelnen Kapitel benannt, die auf dem Weg zum endgültigen Visual Analytics Konzept führten. Anschließend wird ein Ausblick gegeben, wie der entwickelte Ansatz einsetzbar ist. Dazu wird noch auf Themen eingegangen, die zukunftsrelevant sind.

### Zusammenfassung

Zu Beginn dieser Arbeit wird auf Grundbegriffe und Abläufe eingegangen, die im weiteren Verlauf der Arbeit zum Verständnis beitragen sollen. Zu diesen Grundlagen gehört der Prozess der Visual Analytics. Dieser wird in einzelnen Teilschritten aufgeführt und detailliert erklärt. Anschließend werden gängige Eyetracking Visualisierungstechniken vorgestellt, die auch Bestandteil dieser Arbeit sind. Des Weiteren werden noch technische Grundlagen angegeben, die für die Realisierung des Endkonzepts verwendet wurden.

Ein weiterer Schritt war die ausführliche Recherche von schon vergangenen Forschungen in Richtung Eyetracking und Mousetracking Visualisierung, um ein Verständnis zu bekommen, wie diese Art von Daten in der Vergangenheit visualisiert wurden. Auch das Vorgehen bei der Entwicklung von schon realisierten Visualisierungskonzepten auf diesem Gebiet, war hilfreich. Zudem wurden die Ergebnisse von denen dieser Arbeit abgegrenzt.

Die Entwicklung des Endkonzepts geht als eine Hauptaufgabe in diese Ausarbeitung ein. Diese teilt sich in mehrere Teilaufgaben. Teilaufgaben hierbei waren die Datenerfassung, Datentransformation, Entwicklung von Prototypen und schlussendlich das Erstellen des Endgültigen Ansatzes. Sind diese ausgearbeitet, kann das Konzept umgesetzt werden. Dazu sind Schwerpunkte aus der Implementierung angegeben und beschrieben.

Eine Fallstudie soll zeigen, wie der Ansatz bei der Untersuchung von Hypothesen, verwendet werden kann. Es wird hier auf die einzelnen Filtermöglichkeiten und Funktionen eingegangen. Ein Beispielablauf einer Analyse mit der jeweiligen Funktion soll verdeutlichen, wie diese einsetzbar sind.

### Ausblick

Der endgültige Visual Analytics Ansatz macht es möglich, die erfassten Daten aus der Studie auf Korrelation zu untersuchen. Basierend auf den Visualisierungstechniken der Heatmap



und des Scanpath, werden die Bewegungen dargestellt.

Zukünftige Schritte könnten nun sein, diesen Visualisierungsansatz um noch weitere Techniken zu erweitern. Ob diese nun speziell für die Anforderungen an die Visualisierung entwickelt werden oder bestehende Techniken an diesem Problem ansetzen, bleibt zu klären. Auch die schon bestehenden Implementierungen können noch verfeinert werden. Die Filterung nach Probanden beispielsweise wäre noch um zahlreiche Filtermöglichkeiten zu erweitern. Man könnte sich vorstellen, die Daten in entsprechenden Äquivalenzklassen von Probanden einteilen zu können, um diese dann gesondert zu betrachten. Filterungen der Probanden nach dem Alter, technischem Stand oder Aufenthaltsdauer im Internet, wären dabei interessante Ansätze. Dabei wäre aber eine ausführlichere Studie vonnöten, um auch in den einzelnen Äquivalenzklassen genügend Daten vorzufinden.

Die errechneten Abstandswerte von Augpunkt zu Mauspunkt könnten auch noch verbessert werden. Anstelle der linearen Funktion, könnte vielmehr eine Polynomielle oder Gaußfunktion gewählt werden.

Ein weiterer Ansatz wäre die Möglichkeit für bestimmte Bereiche untersuchen zu können, von welchen Seiten aus dieser Bereich angesteuert wurde. So könnte man untersuchen ob Benutzer auf Webseiten möglicherweise Umwege nehmen, um an einen bestimmten Punkt zu kommen.

# Literaturverzeichnis

- [Ale13] M. Alexander. Seminararbeit zu Datenvisualisierung Thema: Heatmaps, 2012/2013. (Zitiert auf Seite 20)
- [Exa13] d3.org Examples. Nutrient Contents - Parallel Coordinates, 2013. URL <http://exposedata.com/parallel/>. [Online; accessed 18-January-2014]. (Zitiert auf den Seiten 6 und 13)
- [HW09] D. Holten, J. J. van Wijk. Force-Directed Edge Bundling for Graph Visualization. *Eurographics/ IEEE-VGTC Symposium on Visualization 2009*, 8, 2009. (Zitiert auf den Seiten 6, 18 und 19)
- [JGP<sup>+</sup>08] T. Joachims, L. Granka, B. Pan, H. Hembrooke, G. Gay. Accurately Interpreting Clickthrough Data as Implicit Feedback. *SIGIR '05 Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 8:154–161, 2008. (Zitiert auf Seite 25)
- [KKEM10] D. Keim, J. Kohlhammer, G. Ellis, F. Mansmann. Eurographics Association, 38621 Goslar, Germany, 2010. (Zitiert auf den Seiten 6, 11, 12 und 15)
- [KPSN04] D. A. Keim, C. Panse, M. Sips, S. C. North. Visual Data Mining in Large Geospatial Point Sets. *IEEE Computer Graphics and Applications*, 9:36–44, 2004. (Zitiert auf Seite 11)
- [Kru14] F. Krueger. *Untersuchung der Korrelation zwischen Maus- und Augenbewegungen bei Webseiten*. Bachelorarbeit, Institute of Visualisation and interactive Systems, University of Stuttgart, 2014. (Zitiert auf den Seiten 8, 30, 40, 57 und 60)
- [LN12] N. E. learning Laboratory (NELL). Reflecting on Eye-tracking, 2012. URL [http://nellatnci.files.wordpress.com/2012/12/peerfinder\\_scanpath.png](http://nellatnci.files.wordpress.com/2012/12/peerfinder_scanpath.png). [Online; accessed 18-January-2014]. (Zitiert auf den Seiten 6 und 17)
- [NC12] V. Navalpakkam, E. F. Churchill. Mouse Tracking: Measuring and Predicting Users' Experience of Web-based Content. *CHI '12 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 10:2963–2973, 2012. (Zitiert auf den Seiten 6, 23, 24 und 25)
- [RFAS08] K. Rodden, X. Fu, A. Aula, I. Spiro. Eye-Mouse Coordination Patterns on Web Search Results Pages. *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, 6:2997–3002, 2008. (Zitiert auf den Seiten 6, 25 und 26)

- [Shn96] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *Proc. 1996 IEEE Conference on Visual Languages*, 7:336–343, 1996. (Zitiert auf den Seiten 11 und 16)
- [Wik13a] Wikipedia. Document Object Model, 2013. URL [http://de.wikipedia.org/wiki/Document\\_Object\\_Model](http://de.wikipedia.org/wiki/Document_Object_Model). [Online; accessed 18-January-2014]. (Zitiert auf Seite 20)
- [Wik13b] Wikipedia. Normalisierung Datenbank, 2013. URL [http://de.wikipedia.org/wiki/Normalisierung\\_%28Datenbank%29](http://de.wikipedia.org/wiki/Normalisierung_%28Datenbank%29). [Online; accessed 18-January-2014]. (Zitiert auf Seite 14)

Alle URLs wurden zuletzt am 17.01.2014 geprüft.

## **Erklärung**

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

---

Ort, Datum, Unterschrift